Department of Electrical and Computer Engineering

Telecommunications and Information Technology Division

Visualization and Virtual Reality Group

Dissertation

Biomechanical Simulation of Virtual Physiological Humans: Modeling of Musculoskeletal Kinematic and Dynamic Redundancy Using Coordinate Projection Methods

Submitted by

Dimitar Stanev

Electrical and Computer Engineer

In Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy

University of Patras

PhD Dissertation №: 367

Patras, December, 20, 2018

Πανεπιστήμιο Πατρών
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Dimitar Stanev

Πιστοποίηση

Η παρούσα Διδακτορική Διατριβή με θέμα:

"Biomechanical Simulation of Virtual Physiological Humans: Modeling of Musculoskeletal Kinematic and Dynamic Redundancy Using Coordinate Projection Methods"

του **Dimitar Stanev** του **Yuriy**, Διπλωματούχου Ηλεκτρολόγου Μηχανικού & Τεχνολογίας Υπολογιστών παρουσιάστηκε δημοσίως στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών στις 20^{η} Δεκεμβρίου 2018, και εξετάσθηκε και εγκρίθηκε από την ακόλουθη Εξεταστική Επιτροπή:

- **Κωνσταντίνος Μουστάκας**, Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών
- **Ευάγγελος Δερματάς**, Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών
- **Κωνσταντίνος Κουτσογιάννης**, Αναπληρωτής Καθηγητής, Τμήμα Φυσικοθεραπείας, Τ.Ε.Ι. Πατρών
- **Βασίλειος Μπαλτζόπουλος**, Καθηγητής, Liverpool John Moores University, Ηνωμένο Βασίλειο
- **Αντώνιος Τζές** Καθηγητής, New York University, Abu Dhabi, UAE
- **Δημήτριος Τζοβάρας** Ερευνητής Α΄ Βαθμίδας και Διευθυντής του Ινστιτούτου Τεχνολογιών Πληροφορικής και Τηλεπικοινωνιών (Ι.Π.ΤΗΛ) του Εθνικού Κέντρου Έρευνας και Τεχνολογικής Ανάπτυξης (Ε.Κ.Ε.Τ.Α.)
- **Δέσποινα Δεληγιάννη**, Καθηγήτρια, Τμήμα Μηχανολόγων & Αεροναυπηγών Μηχανικών, Πανεπιστήμιο Πατρών

Πάτρα, 20, 12, 2018

Ο Επιβλέπων

Αναπληρωτής Καθηγητής
Κωνσταντίνος Μουστάκας

Ο Πρόεδρος του Τμήματος

Καθηγητής
Σταύρος Κουμπιάς

To my parents Yuriy and Daniela, my wife Anastasia and our son Odyssey. This dissertation is equally yours!
–D.S.

Abstract

The principles of musculoskeletal model and simulation have received much attention over the last decades, enabling the prediction of surgical treatments related to different motion limiting disorders. However, their application in clinical practice is still limited partly because the experimental equipment used for measuring the kinematics and kinetics required for the analysis is expensive and obtrusive. The assessment of the internal state (e.g., muscle forces and joint reaction loads) from those measurements does not lead to a unique solution, due to the inherit redundancy of the musculoskeletal system. More specifically, there are more degrees of freedom than those required to perform certain tasks and each degree of freedom is actuated by multiple muscles, leading to infinite combinations of muscle forces that satisfy the movement. Unfortunately, this raises the following questions: which is the true solution employed by the central nervous system and whether the choice of a particular solution can lead to misinterpretation of results?

Coordinate projection methods and their extension for musculoskeletal modeling and simulation are the topic of this thesis. These methods transform the equations of motion into a space of low- or high-dimensionality according to the projection operator. Five different subspaces are studied, namely task, joint, muscle, constraint and null space as well as their relationship. Task space projection simplifies the motion planning problem and the process of synthesizing virtual simulations. This is of great importance, since simulations can be arranged effortlessly and intuitively. Joint space representation is the de facto standard for formulating the underlying equations of motion and dynamics simulation methods. Muscle space projection provides a convenient representation for interfacing musculoskeletal and segmental level models and forms a basis for practical control applications. Constraint projection serves to incorporate the kinematic constraints into the inverse dynamics model. Null space projection can be used to model the redundancy effects of the musculoskeletal system consistently and identify the feasible solution that satisfy the movement and physiological muscle constraints. The redundant nature of the musculoskeletal system introduces variability/uncertainty in simulated quantities leading to misinterpretation of the results if ignored. Therefore, this groundwork provides the appropriate formalization to successfully address these issues, facilitating the application of broader types of studies in the realm of motor coordination.

Περίληψη

Οι αρχές μοντελοποίησης και προσομοίωσης μυοσκελετικών συστημάτων έχουν προσελκύσει έντονο ενδιαφέρον τις τελευταίες δεκαετίες, επιτρέποντας την πρόβλεψη επιπτώσεων χειρουργικών επιλογών στη βελτίωση ή μη παθήσεων που προσβάλουν την κινητική λειτουργία του ανθρώπινου σώματος. Παρ΄ όλα αυτά, η εφαρμογή αυτών των μεθόδων στην κλινική πράξη είναι αρκετά περιορισμένη, διότι απαιτείται σύνθετος και ακριβός εξοπλισμός καταμέτρησης της κινητικής και δυναμικής κατάστασης του ασθενή ώστε να γίνει η κατάλληλη ανάλυση και μελέτη. Η εκτίμηση της εσωτερικής κατάστασης (π.χ. δυνάμεων μυών και αντίδρασης αρθρώσεων) δεν οδηγεί σε μοναδική λύση που να ικανοποιεί την καταγεγραμμένη κίνηση, γιατί το μυοσκελετικό σύστημα διέπεται από πλεονασμό βαθμών ελευθερίας. Πιο συγκεκριμένα, υπάρχουν παραπάνω κινηματικοί βαθμοί ελευθερίας από όσους απαιτούνται για την εκτέλεση μιας κίνησης και κάθε βαθμός ελευθερίας επηρεάζεται από πολλούς μυς με αποτέλεσμα να υπάρχουν άπειροι συνδυασμοί μυϊκών δυνάμεων που να ικανοποιούν την ίδια κίνηση. Τελικά αυτό μας δημιουργεί τους εξής προβληματισμούς: ποια είναι η πραγματική λύση μυϊκών δυνάμεων και μήπως η επιλογή μιας αυθαίρετης λύσεις μπορεί να οδηγήσει σε εσφαλμένα συμπεράσματα;

Για τον σκοπό αυτό έγινε διεύρυνση των μεθόδων προβολής συντεταγμένων, οι οποίες μετασχηματίζουν τις εξισώσεις κίνησης σε κάποιο χώρο επιλογής, για τη μελέτη των μυοσκελετικών συστημάτων. Πιο αναλυτικά, μελετήθηκαν πέντε κατηγορίες προβολής συντεταγμένων: της προβολή στον χώρο της εργασίας, των αρθρώσεων, των μυών, των κινηματικών περιορισμών και του μηδενόχωρου. Η προβολή στον χώρο εργασίας απλοποιεί το πρόβλημα της σύνθεσης εικονικών σεναρίων προσομοίωσης, χωρίς απαραίτητα να χρειαστεί να γίνει η καταγραφή της κίνησης αλλά και της ηλεκτρομυϊκής διέγερσης. Η προβολή στον χώρο των αρθρώσεων είναι η βασική αναπαράσταση για τη μοντελοποίηση και ανάλυση των μυοσκελετικών συστημάτων. Η προβολή στον χώρο των μυών επιτρέπει τη δημιουργία διεπαφών του μυϊκού συστήματος και του ιδιοδεκτικού συστήματος αντανακλαστικών. Αντίστοιχα, η προβολή στον χώρο των κινηματικών περιορισμών επιτρέπει τον συνυπολογισμό των δυνάμεων που απαιτούνται για την ικανοποίηση των περιορισμών κατά την αντίστροφη δυναμική ανάλυση. Η μοντελοποίηση του πλεονασμού λύσεων των μυοσκελετικών συστημάτων μπορεί να επιτευχθεί βάσει του μηδενόχωρου. Η σύνθεση σεναρίων προσομοίωσης αποκλειστικά στον υπολογιστή καθιστά δυνατή τη διερεύνηση των επιπτώσεων χειρουργικών επιλογών στην αλλαγή της κινηματικής και δυναμικής του μυοσκελετικού συστήματος. Παράλληλα, η εκτίμηση των εφικτών λύσεων του συστήματος βάσει του μηδενόχωρου επιτρέπει τη ποσοτικοποίηση της αβεβαιότητας των προβλέψεων. Η παράλειψη των λύσεων μηδενόχωρου μπορεί να οδηγήσει σε εσφαλμένα συμπεράσματα και ως εκ τούτου η παρούσα διατριβή θέτει τα θεμέλια για τη σωστή αντιμετώπιση αυτών των προβλημάτων.

Summary

This dissertation focuses on the branch of biomechanics with emphasis on the development of methods for handling redundancy in modeling, simulation and analysis of musculoskeletal systems. It is an undeniable fact that musculoskeletal systems are inherently redundant. There are more degrees of freedom than those required to perform certain tasks and each degree of freedom is actuated by multiple muscles. This over-availability poses numerous challenges in the process of modeling and simulation that can negatively affect the validity of the models and the obtained results, rendering their application frequently inappropriate for clinical practice.

Despite that mathematical tools for studying redundant systems do exist, they are still not widely adopted in the field of musculoskeletal modeling and simulation. To this end, we examine whether coordinate projection techniques can be used to model and study musculoskeletal redundancy. These methods transform the equations of motion into a space of low- or high-dimensionality according to the projection operator. In this work, we established five useful representations, namely constraint, task, joint, muscle and null space as well as their relationships. The projection from a low- to a high-dimensional space is not uniquely defined (e.g., determine the muscle forces given a set of joint space generalized forces), however, this indeterminacy can be captured mathematically using the notion of null space. Null space solutions, although typically ignored in musculoskeletal system simulations, offer deep insights and provide a much broader framework for modeling a wide range of phenomena arising in clinical practice, such as rigidity and slack muscle disorder.

We present a motivating case study that aims to create a decision support system for the reconstruction of the anterior cruciate ligament after injury, based on personalized musculoskeletal models and simulations. Surgical and rehabilitation treatment planning relies on subjective clinical assessment rather than an objective computational models that are based on principles of physics and physiology. It is thus important to evaluate whether this kind of methodology can potentially assist clinicians to optimally plan the surgical procedure. A full body musculoskeletal model with a six degrees of freedom knee joint, ten ligaments and a tibiofemoral contact model is constructed. The parameters of the model are calibrated based on available measurements. Forward dynamics is utilized to predict the model's response to a given set of externally applied forces, posture configuration and physio-

logical parameters corresponding to different surgical decisions. Inverse dynamics is employed on a gait and jump movements to evaluate the ligament strain profiles. We show that computational musculoskeletal models are a valuable tool for predicting the outcome of surgical decision, demonstrating that such analyses may lead to improved clinical decision-making and ultimately a better health outcome or lower cost of treatment. On the other hand, there are multiple aspects that need to be thoroughly examined in order to enable proper in silico medicine. Predictive simulation is possible only in a forward dynamics setting, yet the muscle excitations that generate a coordinated movement are not known a priori. Additionally, there are infinitely many combinations of muscle forces that can result in the same movement, due to muscle redundancy, hence predicted quantities that depend on these estimates (e.g., joint reaction loads) cannot be calculated reliably.

To address the first problem, a mixed dynamics scheme, where virtual simulations can be planned without necessarily relying on experimental measured kinematics and muscle activity, was implemented. This scheme uses an inverse dynamics model-based controller that drives the model in a closed-loop forward dynamics manner. The controller utilizes task space projection and accepts a set of desired task accelerations in order to produce the muscle excitations required to satisfy the movement. The resulting excitations are fed to the forward dynamics module in order to predict the response of the model to the given input. Task space representation is preferred, because the planning is encoded more naturally with fewer variables required to specify a movement as compared to joint space. Complex movement behaviors can be described through a combination of well-defined set of interrelating task goals. The equations of motion were further projected into the constraint manifold before deriving the task space controller in order to account for the kinematic constraints of the model. This does not only enable applications featuring constrained topologies (e.g., complex anatomical joints and closed kinematic chains), but most importantly the usage of constraint modeling for reducing the required controller command; proper utilization of constraints reduces the generalized forces required by the controller during a specific motion. We present two methods that operate in task space, namely a dynamically-based method for solving the inverse kinematics problem and a method for performing muscle driven dynamics simulations that closely reproduces experimental measurements of kinematics and ground reaction forces.

We approached the musculoskeletal redundancy problem by extending the kinematic and dynamic relationships between task, joint and muscle space using the notion of null space. More specifically, the kinematically redundant degrees of freedom in the context of a task were identified by considering the task and joint space relationships. Likewise, the relations between joint and muscle or task and muscle spaces were used to infer the levels of muscle co-contraction, which result in the same movement behavior. The factors affecting kinematic and dynamic redundancy were identified and included in a closed form solution in order to calculate the feasible muscle forces. This not only demonstrated an effective approach for finding the family of feasible solutions, but also exposes the structure of the

null space, enabling useful post-analyses (e.g., feasibility studies, joint reaction analysis, task/joint stiffness evaluation, etc.). The main advantage of the proposed approach is that the feasible muscle forces are action-specific, accounting for the dynamic evolution of the motion, while also satisfying the physiological constraints of the muscles, outlining the factors that affect the solution space. The redundant nature of the musculoskeletal system introduces variability/uncertainty in simulated quantities leading to misinterpretation of the results if ignored. Therefore, this framework provides the appropriate formalization to successfully address these issues, facilitating the application of broader types of studies in the realm of motor coordination.

Due to the over-availability of actuators, there exist infinitely many solutions for muscle forces giving rise to the same movement, it is thus important to consider not only the limbs' motion, but also the degrees of muscle co-contraction. To characterize the solution space we examine different primitives that are also regulated by the central nervous system. It turns out that the task and joint stiffness are important quantities that regulate the overall stability, impedance and admittance of the musculoskeletal system. Undoubtedly, this is crucial for understanding the muscle coordination mechanisms and the various strategies available to the central nervous system, aiding in the development of effective evaluation and treatment of disorders such as Parkinson's disease. Practical and experimental limitations severely hinder the in vivo measurement of stiffness, while model-based estimation suffers from the musculoskeletal redundancy. To overcome the latter limitation, we chose to model the influence of musculoskeletal redundancy using the notion of null space and identify the possible solutions that satisfy the task and muscle constraints. We present a method to determine the feasible task and joint stiffness of the musculoskeletal system for any movement. Results show that the musculoskeletal system is capable of achieving a highly variable stiffness using muscle co-contraction, highlighting the importance of performing feasibility studies.

Acknowledgments

I would like to express my sincere appreciation to my supervisor Prof. Konstantinos Moustakas for the support and mentoring which I have received during these years, granting me the freedom to pursuit my research ideas and always believing in my skills. His professionalism, hard-working, wisdom and patience are some of the many traits that I admire and which resulted in an excellent collaboration between us. Thanks are also due to Prof. Constantinos Koutsojannis for the splendid collaboration especially in the beginning of my PhD, Prof. Evaggelos Dermatas for his aspiring and practical teachings during my studies and the scientific committee for our cooperation both during my studies and at a more professional level.

I am gratefully to my colleague and friend Aristotelis Spathis-Papadiotis for his essential help, valuable remarks and countless discussions on philosophical, technical, scientific and other topics of interest. Many thanks to my friends, students and colleagues for the great time that we shared in the Visualization and Virtual Reality Group.

Other acknowledgments are owed to the Stanford SimTK: OpenSim team for putting together a great community which was a source of inspiration. I am grateful for the opportunities that you provided, helping me learn things, among which to always share my scientific results and help the community move forward.

All my thanks to my family for always believing, supporting and respecting my decisions. Last but not least, my gratitude to my wife for supporting me in every difficult moment and celebrating each success, and to my baby son for bringing happiness and motivation in our life.

This work was supported in part by the EC Horizon 2020 project OActive: Advanced personalized, multi-scale computer models preventing osteoarthritis, Grant Agreement No. 777159 and by the EC FP7 project NoTremor: Virtual, physiological and computational neuromuscular models for the predictive treatment of Parkinson's disease, Grant Agreement No. 610391. The funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Contents

Abstract **vii**

Summary **ix**

Acknowledgments **xiii**

List of Figures **xxiv**

List of Tables **xxv**

Acronyms **xxvii**

1 Introduction **1**
1.1 Outline 4

2 Fundamentals **7**
2.1 Mathematical Background 7
2.1.1 Subspace Projection 7
2.2 Musculoskeletal Modeling 11
2.2.1 Skeletal Modeling 12
2.2.2 Constraint Modeling 13
2.2.3 Muscle Dynamics 15
2.2.4 Muscle Routing 17
2.3 Musculoskeletal Simulation and Analysis 20
2.3.1 Scaling 20
2.3.2 Inverse Kinematics 21
2.3.3 Inverse Dynamics 22
2.3.4 Forward Dynamics 23
2.3.5 Muscle Optimization 24
2.4 An Open-Source Educational Example 26
2.4.1 Eye Modeling 26
2.4.2 Eye Simulation 29

3 Anterior Cruciate Ligament Reconstruction Through Personalized Simulation 33
3.1 Introduction . . . 34
3.1.1 Related Work . . . 35
3.2 Methods . . . 35
3.2.1 Ligaments Modeling . . . 35
3.2.2 Musculoskeletal Modeling and Simulation . . . 36
3.2.3 Lachman Test . . . 38
3.2.4 Clinical Data . . . 38
3.3 Results . . . 39
3.3.1 Ligament Parameter Calibration . . . 39
3.3.2 Clinical Decision Support System . . . 40
3.4 Discussion-Conclusions . . . 44

4 Simulation of Constrained Musculoskeletal Systems in Task Space 49
4.1 Introduction . . . 50
4.1.1 Related Work . . . 51
4.2 Methods . . . 53
4.2.1 Equations of Motion of the Plant . . . 53
4.2.2 Inverse Dynamics Controller . . . 54
4.2.3 Inverse Task Space Controller for a Single Task . . . 56
4.2.4 Inverse Task Space Controller for Multiple Tasks . . . 57
4.2.5 Task Space Inverse Kinematics . . . 59
4.2.6 Task Space Computed Muscle Control . . . 62
4.2.7 Muscle Effort Assessment . . . 63
4.3 Results . . . 65
4.3.1 Comparison of the Constrained Models . . . 66
4.3.2 Inverse Kinematics on Gait . . . 68
4.3.3 Inverse Muscle Driven Simulation on Gait . . . 70
4.3.4 Absolute (Cartesian) Coordinates . . . 71
4.3.5 Simulation of Closed Kinematic Chains . . . 71
4.4 Discussion-Conclusions . . . 72

5 Modeling Kinematic and Dynamic Redundancy Using Null Space Projection 79
5.1 Introduction . . . 80
5.1.1 Related work . . . 82
5.2 Methods . . . 83
5.2.1 Relation Between Task and Joint Space Quantities . . . 83
5.2.2 Task Space Equations of Motion . . . 85
5.2.3 Relation Between Muscle and Joint Space Quantities . . . 87
5.2.4 Muscle Space Equations of Motion . . . 89
5.2.5 Exploitation of Kinematic and Dynamic Redundancy . . . 90
5.3 Results . . . 94

5.3.1 Muscle Space Projection and Reflexes 95
5.3.2 Characterization of the Feasible Muscle Force Space . . . 96
5.3.3 The Effect of Null Space on the Joint Reaction Loads . . . 98
5.4 Discussion-Conclusions 99

6 Stiffness Modulation of Redundant Musculoskeletal Systems 105
6.1 Introduction 106
6.2 Methods 107
6.2.1 Muscle, Joint and Task Space Stiffness 107
6.2.2 The Influence of Kinematic and Dynamic Redundancy . . 109
6.3 Results 111
6.3.1 Variability in the Arm Endpoint Stiffness 111
6.3.2 Calculation of the Feasible Joint Stiffness During Walking 113
6.4 Discussion-Conclusions 114

7 Concluding Remarks 117

A Appendix 121
A.1 Implementation Details 121
A.2 Fundamental Theorems 121
A.3 Tensor-Vector Product 124
A.4 Simple Arm Model 125
A.5 List of Publications 130

Bibliography 144

List of Figures

2.1 Various relations between the different subspaces associated with matrix $\boldsymbol{A}$. Note that the null space solutions have zero contribution when projected by $\boldsymbol{A}$. 9

2.2 In the presence of constraints, all permissible configurations lie on the constraint manifold. To satisfy these restrictions, the constraint forces must be applied. These forces do no work under any virtual displacement that satisfy the constraints. 14

2.3 A three element Hill-type model. The contractile element is in parallel with a passive element, and they are in series with the tendon (image courtesy (Millard et al., 2013)). 16

2.4 Comparison of the characteristic curves (F-L, F-V, F-PE and F-T) between Millard and Thelen muscle models (image courtesy (Millard et al., 2013)). 17

2.5 The active F-L and F-PE curve definition of the Millard muscle model as implemented in `OpenSim`, (image courtesy `OpenSim` user guide). 18

2.6 Model for mapping muscle forces into joint space generalized forces (image courtesy (Valero-Cuevas, 2009b)). 18

2.7 An arm model with two DoFs and six muscles. There are muscles that are mono or bi-articular. The muscle path is modeled as a collection of points. Furthermore, the muscles can wrap around the bones as can be seen for the biceps. 19

2.8 IK errors (total RMS error, marker error and max coordinate error) improved after applying scaling to adjust the general model to subject's size. 22

2.9 Schematic of the CMC method (Thelen and Anderson, 2006). . . 25

2.10 An eye model containing six EMs and three DoFs. 26

2.11 Simulated saccade response for different model parameters. Left subplot represents the simulated generalized coordinates, middle the generalized velocities and right the estimated EMs excitation levels. 31

3.1 A portion of the developed model, with the muscles visualized with red and the ligaments with green. 37
3.2 Rolimeter configuration during Lachman test. 39
3.3 Comparison between experimentally measured and model response to the Lachman test after calibration of the ligament parameters. The diagram shows the anterior-posterior displacement with respect to the force applied to the posterior aspect of the proximal tibia. 42
3.4 The diagrams show the anterior-posterior displacement with respect to the force applied to the posterior aspect of the proximal tibia during the simulated Lachman test for different surgical choices. On the left diagram the response of the model is investigated based on the insertion of single or double bundles for the ACL ligament. On the right diagram an investigation is made by adjusting the resting length of the two bundles, by increasing or decreasing the initial resting length. 43
3.5 Simulation of the strain force for LCL (left) and aDMCL (right) during the simulated Lachman test which are the main restrictors in the absence of ACL bundles. 44
3.6 Comparison between simulated strains for all ligaments with respect to the reference strain of a healthy knee flexion motion (reference model). Investigation of 8 different attachment positions of the aACL on the tibia and their effect on the simulated strain curves. 46
3.7 Simulation of gait (left) and jump action (right) for different insertion positions of the aACL on the tibia. For the gait simulation the annotations are TO: toe off and HC: heel contact. For the jump action the annotations are ST: stand, SE: seated and JU: jump postures. 47

4.1 Architecture diagram of the TSIK method. The input to the system is a set of marker trajectories or other types of measurements (e.g. IMU). Markers are grouped based on their attachment location on the body and a set of task goals is extracted from the recorded motion. A tracking scheme is used to track the task goals (e.g. PD controller) and the desired goals are transformed into generalized forces by the ITSC. Finally, these forces actuate the model in a FD manner. The product of this algorithm is both a set of task goals and the simulated motion that reproduces the observations. 60
4.2 This figure presents two frames of recorded gait. An ellipse is placed to denote the grouping of the markers, where the color defines the type of the task assigned to each group as an example. The transition, transformation ($\boldsymbol{T}(n, n+1)$) between the two frames for the foot segment has been annotated. 61

4.3 Architecture diagram of the TSCMC. The input can be the task goals originating from the TSIK or a set of goals from a different objective controller. The goals are tracked by an internal tracking scheme (e.g. PD controller), while the desired goals are used by ITSC to derive the required generalized forces. These forces are then mapped to muscle excitation patterns through an optimization procedure and the model is numerically integrated for the given muscle activation patterns. 63

4.4 This figure depicts the time instances of a customly planned trajectory. The upper extremity model (Saul et al., 2014) consists of seven DoFs and forty-seven muscles. The model defines a set of constraint equations that regulate the movement of the shoulder complex. 65

4.5 The simulated task trajectory of the upper limb model for the finger point task. The $y - z$ plane defines the resulting pattern. The trajectories of Model-A (Equation 4.9) are drawn with red dotted lines, while the blue dashed lines denote the trajectories of Model-B (Equation 4.12). Both models result in identical motion. 67

4.6 The computed forces that drive the model to perform the required movement. These forces are decoupled into motion related forces (task and null space forces) and constraint forces. A comparison of the magnitude values for the two models is presented. The trajectories of Model-A (Equation 4.9) are drawn with red dotted lines, while the blue dashed lines denote the trajectories of Model-B (Equation 4.12). Model-A results in larger motion forces command compared to Model-B, while Model-B results in larger constraint forces than Model-A. 68

4.7 The simulated generalized coordinates ($\boldsymbol{q}$) of the upper limb model as a result for the defined custom trajectory. The trajectories of Model-A (Equation 4.9) are drawn with red dotted lines, while the blue dashed lines denote the trajectories of Model-B (Equation 4.12). Both models result in identical motion (residuals are zero). 69

4.8 Comparison of the total RMS, min and max error between the TSIK and the optimization-based IK algorithms for a single gait cycle. The shaded area encloses the min and max marker error as its upper and lower bound respectively. 70

4.9 This figure presents the muscles excitations for a single gait cycle in comparison between CMC (red solid line) method and the proposed TSCMC (blue dashed line). There are some minor differences, as the two methods have their implementation specifics, but in general they agree. 74

4.10 This figure presents the residual forces/torques for a single gait cycle in comparison between CMC (red solid line) method and the proposed TSCMC (blue dashed line). 75

4.11 Left: an unconstrained model (generalized coordinates) with two DoFs. Right: constrained model that uses absolute coordinates twelve DoFs and ten constraint algebraic equations. 76

4.12 Body kinematics analysis of the two DoF (generalized coordinates) (blue) and the model that uses absolute coordinate (red). XYZ quantities correspond to the translational components, while $Oxyz$ quantities represent the orientation of the bodies. The task goal is a sinusoidal movement of the tip of the second body in the $\boldsymbol{y}$ direction. The two models produce identical movements (residuals are zero). 76

4.13 A model of a closed kinematic chain topology. The first body (left) is permitted to rotate around the $\boldsymbol{z}$-axis with respect to the ground frame. The second body (right) is connected to the ground by an offset of the length of the third body (top) and it is also permitted to rotate around the $\boldsymbol{z}$-axis. The third body is connected with the first and second bodies by two pin joints. 77

4.14 Body kinematics analysis of the two models Model-A (blue), Model-B (red) on a closed kinematic chain model. XYZ quantities correspond to the translational components, while $Oxyz$ quantities represent the orientation of the bodies. The goal is to rotate body 1 along the $\boldsymbol{z}$-axis. Since body 2 is connected to body 1 by body 3 (top) their orientation is identical. Furthermore, because of the structure we expect that body 3 do not rotate at all for this particular experiment. The two models produce identical movements (residuals are zero). 77

5.1 The inequality Equation 5.30 is sampled using Algorithm 5.1 for different levels of iteration depth (d). 94

5.2 Posture analysis of the system in muscle space, including a "descending" command from the CNS. The first row of figures presents the simulated joint space coordinates and speeds. The second row depicts the evolution of the muscle space quantities of the controller. The model is perturbed and the response of the system is observed. The system absorbs the impact and returns to its initial configuration. The controller gains are $k_p = 10$, $k_d = 10$ and the loop delay $\tau_{so} = 20$ ms. 97

5.3 Posture analysis of the system in muscle space without "descending" command from the CNS. The first row of figures presents the simulated joint space coordinates and speeds. The second row depicts the evolution of the muscle space quantities of the controller. After the perturbation the system settles to a new, stable equilibrium point. The controller gains are $k_p = 0$, $k_d = 10$ and the loop delay $\tau_{so} = 20$ ms. 98

5.4 Collective simulation results for the simplified hand movement. The evolution of the general coordinates and speeds as well as the simulated movement are presented in the figures of the first row. The second row shows the generalized forces, the simulated and desired task goals and the task space forces computed by the controller. The parameters used in this experiment are $k_p = 50$, $k_d = 5$, $a = 0.3$, $b = 4$, $t_0 = 1$ and $\gamma = \pi$. 100

5.5 The evolution of the feasible force set. Two instances of the simulated movement along the $-\boldsymbol{x}$ direction are shown to demonstrate the evolution of the feasible muscle space. The box plot depicts the force variability of each muscle as contributed to the redundant nature of the system. The correlation matrix shows the positive or negative correlation between the different muscle pairs. 101

5.6 The effect of the null space muscle forces on the joint reaction loads. The normalized (with respect to body weight) reaction loads on the hip joint during walking are reported along with the heel strike and toe-off events. The shaded area is the reaction force range as attributed to the null space solutions of muscle forces. The red dotted line are the results obtained from `OpenSim`. The large range in the reaction loads confirms that misinterpretation of the results is possible if the null space solutions are ignored. . . . 102

5.7 Overview of various relationships between task, joint and muscle space. 104

6.1 Collected results of the simulated arm movement (along the $-\boldsymbol{x}$ direction, $\gamma = \pi$) at three time instants. Left diagram shows the feasible endpoint stiffness using scaled ellipses (scaling = 0.0006), where the red and green lines denote the major and minor axes, respectively. The middle plot illustrates the ellipse parameter distribution (area and orientation ϕ). The box plot on the right depicts the feasible joint stiffness distribution. 112

6.2 Comparison between the sampled and symbolically obtained moment arm of the vastus intermedius muscle at the knee joint as a function of the knee flexion angle (left). Likewise, the moment arm of the hamstring muscle at the knee joint as a function of the hip and knee flexion angles (right). 113

6.3 The feasible joint stiffness of the hip, knee and ankle joints during walking with the heel strike (HS) and toe-off (TO) events annotated accordingly. 114

A.1 Diagram of the simplified arm model, which has three DoFs and nine muscles, some of them being bi-articular. The muscle origins are labeled as a_i and the muscle insertions as b_i. l_i stands for muscle length, Lc_i for center of mass and J_i for joint center. 125

List of Tables

2.1 Muscle path points for the six EMs defined in the local frame of the eye globe (dimensions are given in meters). 28
2.2 Millard muscle parameters for the EMs. 28
2.3 Parameters of the active F-L characteristic curve. 28
2.4 Parameters of the F-PE characteristic curve. 28

3.1 Average, min and max values for the five ligaments determined from MRI measurements from 30 young patients. 40
3.2 Initial values for the ligament parameters; l_r reference length, ϵ_r reference strain and k stiffness (a: anterior, p: posterior, i: inferior and D: deep). 41
3.3 Potential variables that can be modified in order to investigate their effect on the biomechanical behavior of the knee. 41
3.4 The error between simulated and reference strains for every ligament and for every insertion position of the aACL on the tibia (the minimum errors are marked with bold). The error is defined as the accumulated absolute value between reference and simulated curves after alignment of the time series with respect to knee flexion-extension angle. Minimum values show that for the concrete insertion position the ligament strain curve is the closest to the reference strain curve of an intact knee during flexion. 43

A.1 A summary of publicly developed materials related to the dissertation. 121
A.2 Body parameters: mass, inertia, length and center of mass (CoM). 129
A.3 Muscle parameters: muscle origin, insertion and maximum force. . 130

Acronyms

ACL Anterior Cruciate Ligament. 4, 33–35, 38, 40–44, 46, 47, 117, 121

CMC Computed Muscle Control. 24, 25, 49, 51, 63, 66, 70, 73–75

CNS Central Nervous System. 6, 11, 26, 82, 88, 93, 95, 97, 98, 105, 106, 114, 115, 118, 119

DAE Differential Algebraic Equations. 13, 15, 52–54

DMC Direct Marker Control. 49, 59

DO Dynamic Optimization. 24, 25

DoF Degree of Freedom. 63, 71, 76, 80, 118

DoFs Degrees of Freedom. 2, 3, 5, 12, 13, 17, 19, 26, 27, 33, 34, 36, 49, 53, 56, 65–67, 71, 76, 80–83, 85, 94, 99–101, 107, 108, 111, 113, 118, 125

DSS Decision Support System. 33, 44, 121

EMG Electromyography. 45, 52, 106

EMs Extraocular Muscles. 20, 26–31

EoMs Equations of Motion. 3, 5, 7, 11–13, 15, 18, 20, 23, 36, 49–51, 53–56, 72, 81, 85, 89, 90, 94, 95, 100, 125, 126

F-L Force-Length. 15, 17, 18, 28, 29

F-PE Passive-Force-Length. 15, 17, 18, 28

F-T Tendon-Force-Length. 16, 17

F-V Force-Velocity. 15, 17, 28, 29

FD Forward Dynamics. 3, 5, 12, 15, 20, 23–25, 29, 33, 37, 40, 44, 45, 49, 51, 52, 54, 60, 63, 73, 90, 94, 95, 100, 111

FE Finite Element. 35

FEM Finite Element Method. 1

GRF Ground Reaction Forces. 5, 13, 53, 118

HD Hybrid Dynamics. 24

ID Inverse Dynamics. 2, 3, 5, 12, 20–24, 37, 43–45, 49, 52, 54, 55, 58, 66, 68, 89, 90, 94–96, 99, 100, 110, 111, 113, 118

IK Inverse Kinematics. 5, 21, 22, 49, 51, 59, 66, 68, 70, 72, 73, 99, 110, 113, 118

IMU Inertial Measurement Unit. 59, 60, 73

IMUs Inertial Measurement Units. 21

IO Inferior Oblique. 27, 28

IR Inferior Rectus. 27, 28

ITSC Inverse Task Space Controller. 54, 59, 60, 62–64, 71, 72

KE Kinetic Energy. 125, 126

LCL Lateral Collateral Ligament. 35, 40–42, 44

LR Lateral Rectus. 27, 28

MCL Medial Collateral Ligament. 35, 40–42, 44

MD Mixed Dynamics. 5, 12, 24, 49, 94, 100, 118

MoCap Motion Capture. 21, 52, 59

MPP Moore-Penrose Pseudoinverse. 9, 10, 84–88, 91, 123, 124

MR Medial Rectus. 27, 28

MRI Magnetic Resonance Imaging. 38–41, 44

ODE Ordinary Differential Equations. 15, 52, 54

PCL Posterior Cruciate Ligament. 35, 40, 41

PD Proportional Derivative. 25, 29, 30, 60, 63, 96, 111

PE Potential Energy. 126

RMS Root Mean Square. 22, 59, 68, 70

RRA Residual Reduction Algorithm. 70

SO Static Optimization. 24, 25, 64, 99, 110

SR Superior Rectus. 27, 28

SVD Singular Value Decomposition. 10, 62, 85

TKA Total Knee Arthroplasty. 34

TSCMC Task Space Computed Muscle Control. 51, 62, 63, 66, 70, 73–75

TSIK Task Space Inverse Kinematics. 51, 60, 62, 63, 68, 70

UMH Uncontrolled Manifold Hypothesis. 97

VPH Virtual Physiological Humans. 34, 44

Chapter 1

Introduction

Computational modeling and simulation is a powerful technique for analyzing the outcomes of a mathematical model associated with a particular system. Once a model is created and validated, it can be used to explore hypotheses and theories by employing simulations and comparing the predicted quantities with available experimental measurements. It helps us extract value from data, ask questions about behaviors and then use the answers to understand, design, manage and predict the behaviors of complex systems and processes. Perhaps, one of the most prominent representatives of these technologies is the Finite Element Method (FEM), used to create virtual models of products, even before the products are manufactured, enabling much of the technological progress seen in recent decades. Therefore, it is important to consider how these practices can be translated and applied to the study of biological systems and in particular biomechanics.

Over the last decades, the branch of biomechanics has made a significant leap forward. This is partly contributed to the rise of the digital age and the countless technological advances in engineering, medicine and biology. Nowadays, biological computer models of various complexity are created in order to explore hypothesis and answer scientific questions. Experimental measurements of high sophistication, including cell, tissue and organ, are possible, thus the developed models tend to approximate the biological system more accurately. Even so, we are still far away from understanding how the brain interacts with the body to produce coordinated movement. The benefit of such a knowledge is critical to understanding neural computation, vertebrate evolution and function, clinical dysfunction and rehabilitation (Valero-Cuevas, 2009b).

Computational models of movement provide a framework for integrating facts about the biology of movement, that can be used to explore motor coordination strategies (Latash et al., 2010; Steele et al., 2015), analyze athletic performance and simulate treatments for motion limiting disorders (Delp and Loan, 2000). Biological systems are hard to model and analyze mainly due to their inhered complexity and inability to measure different parts of the system with low uncertainty (Valero-Cuevas et al., 2003; Hicks et al., 2014; Myers et al., 2015). One of the biggest chal-

lenges is to show that the model predictions are robust, accurate and convene important information about the physiology itself. Nonetheless, many studies (Arnold et al., 2000; Besier et al., 2005; Reinbolt et al., 2009; Hamner et al., 2010; Hicks et al., 2011; Fregly et al., 2011; Ferber et al., 2016; Uchida et al., 2016) have shown that musculoskeletal computer simulations can complement experimental studies. Therefore, important elements of movement, including neural signals and muscle forces, are extremely difficult to measure experimentally in vivo, thus having significant clinical implications in several orthopedic and neurological contexts.

The problem with complex living organisms is that they are dramatically entangled, so the functioning of each of their parts can hardly be assumed independent from the others. Movement is the result of a complex interaction between the muscular, skeletal and nervous systems. The description and analyses of these systems tend to simplify the phrasing of the control problem, mostly because the mathematics of robotics is best developed for torque-driven limbs (Inouye et al., 2012). In order to create anatomically realistic and physiological plausible musculoskeletal models that could be used in clinical practice, we must understand that biological actuators (Zajac, 1989; Thelen, 2003; Millard et al., 2013) and anatomical joints (Seth et al., 2010; Saul et al., 2014) are significantly different in contrast to the idealized actuators and joints commonly used in robotics.

The dramatic progress in medical imaging and biomedical instrumentation has enabled accurate 3D reconstruction and parameterization of musculoskeletal models. Subject-specific musculoskeletal modeling can be applied to study motion limiting disorders, allowing inclusion of personalized anatomy and properties (Valente et al., 2014). Personalized computational models of the musculoskeletal system could facilitate objective prediction of subject-specific functional outcome for different treatment designs being considered by experts.

Simulation and analysis of complex systems is now possible due to the advances in calculus, mathematics, computational science and engineering, and perhaps equally important, the increase in raw computational power offered by modern hardware and software. This development is essential, because we can solve the very large system of equations that describe the physiological problem under study. To this end, different methods for analyzing the kinematics and dynamics of the musculoskeletal system has been proposed and developed (Erdemir et al., 2007), complementing clinical analyses by providing useful insights that are otherwise inaccessible (e.g., estimation of muscle forces).

What makes musculoskeletal modeling and simulation challenging is not only the process of creating accurate mathematical representations of the underlying system, but also the multiplicity of solutions obtained when they are analyzed considering that these systems are inherently redundant, both kinematically (more Degrees of Freedom (DoFs)) and dynamically (more muscles) (Valero-Cuevas, 2009b). This causes numerous challenges when the system is analyzed in an Inverse Dynamics (ID) manner, where the muscle forces are estimated from the measured kinematics and externally applied forces. The essence of the problem is that there are more unknown quantities than equations, resulting in infinitely many pos-

sible solutions. To overcome these limitations, various constraining assumptions are introduced (Anderson and Pandy, 2001; Thelen et al., 2003; Thelen and Anderson, 2006), which can severely hinder the validity of the obtained results (Pedotti et al., 1978; Prilutsky and Zatsiorsky, 2002; Wesseling et al., 2015; Koeppen et al., 2017). Since the ID methods are a valuable tool for analyzing musculoskeletal models, it is important to develop new methods that can cope with indeterminacy.

Although ID methods are valuable in clinical settings (De Groote et al., 2009), they cannot be used to predict situations that have not yet happened (Ong et al., 2016). The process of recording a movement is not only highly time-consuming and expensive, but also frequently impractical in patients with movement disorders. Often we are interested in testing a hypothesis, new algorithms or reproduce experiments, without having the luxury to record the exact movement. The question is whether we can design virtual experiments and perform analyses without any experimental measurements. This requires mechanisms to plan virtual simulation that are anatomically realistic and predict the interaction of the model with its environment. As the body has many DoFs, it is easier to record a movement experimentally rather than design anatomically realistic behaviors virtually through motion planning. It turns out that, even if the motion of the body is high-dimensional, we can still create realistic movements by working in a low-dimension subspace. This is achieved by expressing the Equations of Motion (EoMs) into the space of a task using the theory of projection operators (Khatib, 1995).

Recent advances in Forward Dynamics (FD) methods make them a popular choice for simulation (Buchanan et al., 2006; van den Bogert et al., 2011). Even though they are relatively more complex and computationally more demanding as compared to the ID methods, they are a key component for performing predictive simulations (Ackermann and van den Bogert, 2010; De Groote et al., 2016; Lee and Umberger, 2016). What makes the application of FD methods challenging is the process of finding the appropriate muscle excitation patterns that will result in a coordinated movement. To overcome this challenge, we examine whether ID and FD methods can complement each other, since the former can provide an estimate of the model inputs and the latter can predict the outcome of the given command.

Coordinate projection techniques are used extensively in this dissertation (Raibert and Craig, 1981; Eich, 1993; Aghili, 2005; De Sapio et al., 2006; Sentis, 2007). These methods transform the EoMs into a space of low- or high-dimensionality according to the projection operator. We established five useful representations, namely constraint, task, joint, muscle and null space as well as their relationships. Constraint projection serves to incorporate constraints into the EoMs. Task space projection leads to an intuitive encoding of the motion planning problem. Muscle space projection provides a convenient representation for interfacing musculoskeletal and segmental level models and forms a basis for practical control applications. Furthermore, the null solution space can be identified by examining the relationship between the different projections, enabling the modeling and evaluation of hypotheses related to healthy behavior and pathological conditions presented in the coordination process. The latter permits the development of new methods for

modeling, simulation and analysis of redundant musculoskeletal systems.

Reproducibility and reuse of scientific results, especially in the field of musculoskeletal modeling and simulation has been challenged recently. The developed models are highly complex and sensitive to parameter changes, while thorough validation and verification requires a lot of effort. However, implementing model sharing and reproducibility analysis introduces scientific, technological and cultural challenges to the academic enterprise, which has traditionally relied on exchange of knowledge through scholarly publishing (Erdemir et al., 2016). In this dissertation, we we use extensively the `OpenSim` platform (Delp et al., 2007), by building a library of simulations and models that can be exchanged, tested, analyzed and improved through a multi-institutional collaboration. Therefore, the source code along with any developed material are made publicly available (section A.1), providing simple examples so that the readers can reproduce, understand and reuse the presented methods.

In silico technologies are transforming the fundamental biomedical and clinical research. The improvement of subject-specific modeling enables the creation of tailored mathematical representations of individual's physiology. Computer aided virtual simulations of "what if" studies may then be performed in order to obtain accurate predictions for different treatment options being considered by the clinicians. New simulation methods for studying redundant musculoskeletal systems, as developed in this dissertation, could then be used to properly interpret the obtained results as well as provide robust and accurate predictions. If in silico technologies succeed in augmenting traditional clinical practices, then such analyses may lead to improved clinical decision-making and ultimately a better health outcome or lower cost of treatment.

1.1 Outline

Each section is self-contained with references to other chapters when considered necessary. This dissertation is organized as follows:

- In chapter 2, we introduce the fundamental concepts of musculoskeletal modeling and simulation by reviewing the current state-of-the art.

- In chapter 3, we present a motivating case study that **aims to create a decision support system for the reconstruction of the Anterior Cruciate Ligament (ACL) after injury, based on personalized musculoskeletal models and simulations**. We evaluate whether this kind of methodology can potentially assist clinicians to optimally plan the surgical procedure. If computational musculoskeletal models can predict the outcome of a surgical decision reliably, then such analyses may lead to improved clinical decision-making. We demonstrate that this is possible in the context of ACL injuries, however, there are multiple aspects that need to be thoroughly examined in

order to enable proper in silico medicine. This cultivated the development of new methods that address these limitations in the following chapters.

- In chapter 4, we **develop methods for planning virtual simulations of constrained musculoskeletal systems without necessarily relying on experimentally measured kinematics and muscle activity**. ID methods are useful for analyzing settings that have happened based on the experimentally measured kinematics and kinetics, however, it would be even more useful if musculoskeletal models are utilized in a predictive manner. This is only possible in a FD setting, yet the muscle excitations that produce a coordinated movement are not known a priori. ID and FD methods are combined together (Mixed Dynamics (MD)) in order to perform predictive simulation without relying on muscle excitations as an input. The motion planning is solved using task space projection methods in order to reduce the DoFs required to synthesize a specific motion. The operational task space formalization (Khatib, 1995) provides the mathematical tools to map the EoMs of the underlying system to the space of the task and control the movement in this domain. The proposed algorithms and controllers are designed so as to handle constrained topologies. This is of significant importance because biological models (e.g., anatomical joints (Seth et al., 2010)) and interaction with the environment (e.g., crank-rotation tasks (Davoudabadi Farahani et al., 2016)) are modeled using constraints. We present two methods that operate in task space, namely a dynamically-based method for solving the Inverse Kinematics (IK) problem and a method for performing muscle driven dynamics simulations that closely reproduce experimental measurements of kinematics and Ground Reaction Forces (GRF).

- In chapter 5, **methods for handling redundancy in the process of modeling, simulation and analysis of musculoskeletal systems are develop**. Musculoskeletal models will always be dynamically redundant, as muscles can only pull or resist stretching, thus each joint requires at least a pair of an agonist-antagonist leading to an underdetermined system of equations. How one can be sure whether the quantities (e.g., joint reaction loads) that rely on these estimates can be computed reliably, if the system can have infinite solutions of muscle forces satisfying the motion? In our approach, we reformulated the equations that describe the musculoskeletal systems in order to handle redundancy. Then we identified the feasible solution space without introducing assumptions (e.g., minimum effort solutions) that can limit the scope and extend of the analysis. From a theoretical perspective, the main contribution of this part is the modeling of musculoskeletal redundancy using the notion of null space and the calculation of the feasible solution space, enabling useful post-analyses (e.g., feasibility studies, joint reaction analysis, task/joint stiffness evaluation, etc.). This can potentially enable the modeling, simulation and evaluation of hypotheses related to normal and patho-

logical conditions presented in the coordination process (e.g., slack muscle disorder, increase in joint stiffness and rigidity), which can manifest in the null solution space.

- In chapter 6, **a model for computing the task and joint space stiffness using inverse methods that account for the musculoskeletal redundancy effects is presented**. The ultimate goal is to create models which can accurately predict the limbs' stiffness from experimentally measured kinematics and kinetics. This is particularly useful since the Central Nervous System (CNS) does not coordinate the motion of the limbs alone, but also the degrees of muscle co-contraction, thus the stiffness is an important quantity that can provide valuable information during the analysis phase. In order to obtain model-based estimates for the limbs' stiffness one must solve the muscle redundancy problem, by typically assuming some objective criterion (e.g., activation squared). Undeniably, single solution methods are of great importance, however, the assumptions that are introduced can severely hinder the validity of the obtained results. This is especially evident in the case of rigidity in Parkinson's disease, which is characterized by the inability of the muscles to relax. Clearly, a particular solution will not only bias the results, but will also affect the calculation of other quantities that depend on the muscle forces, such as joint reaction loads or stiffness quantities. On the other hand, identification of the feasible solution space can help to properly interpret results obtained from the redundant musculoskeletal systems. Considering these facts, we show that global stiffness evaluation is possible, hence this metric can be used to characterize various actions and complement traditional analyses. The calculation of the feasible stiffness can aid in finding patterns of low stiffness capacity in order to improve the design of products and interventions that target these aspects of movement (e.g., exoskeleton design, interaction with the environment, ergonomics, etc.)

- chapter 7 summarizes and discusses the contributions of this dissertation and future research directions.

- Appendix A presents important topics that are referenced by the sections.

Chapter 2

Fundamentals

In this chapter, we introduce the fundamental concepts of musculoskeletal modeling and simulation. We start our discussion with fundamental theorems and propositions from linear algebra and present the notion of subspace projection that is extensively used in this dissertation (e.g., Chapters 4, 5 and 5). We then move to the topics of computational musculoskeletal modeling, which are subdivided into three categories, namely skeletal, constraint and muscle modeling. Skeletal modeling refers to the physiological parameterization of body segments and anatomical joints towards deriving the rigid-body EoMs describing the system. Constraint modeling is also introduced since biological models and interaction with the environment commonly impose restrictions on the motion of bodies. Muscle models are incorporated into the EoMs, so as to study how neuromuscular systems produce mechanical function. We continue with the subject of musculoskeletal analysis and simulation, covering topics related to generic model parameterization, (inverse) kinematics and inverse/forward/mixed/hybrid dynamics as well as static/dynamic muscle optimization. A physiologically realistic human eye model, that can be used for kinematics and dynamics analysis, is presented as an open-source educational example.

2.1 Mathematical Background

The mathematical notation follows the ISO 80000–2:2009 guidelines. Variables are set in italic (e.g., $a \in \Re$); constants and function are set in roman; vector and matrices are set in bold italic. Vectors are denoted by lower-case letters (e.g., $\boldsymbol{c} \in \Re^m$) and matrices by upper-case (e.g., $\boldsymbol{C} \in \Re^{n \times n}$).

2.1.1 Subspace Projection

In this dissertation, we use extensively coordinate projection methods (chapter 4) to project the EoMs into different subspaces or to model musculoskeletal redundancy using null space projection (chapter 5), deeming the introduction of the relevant

notation necessary. For a given linear transformation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in \Re^n,\ \boldsymbol{b} \in \Re^m$$
$$\boldsymbol{A} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} = \begin{bmatrix} | & & | \\ \boldsymbol{c}_1 & \cdots & \boldsymbol{c}_n \\ | & & | \end{bmatrix} = \begin{bmatrix} - & \boldsymbol{r}_1^T & - \\ & \vdots & \\ - & \boldsymbol{r}_m^T & - \end{bmatrix} \tag{2.1}$$

where the $\boldsymbol{A}$ matrix defines a mapping $\boldsymbol{A} : \Re^n \rightarrow \Re^m$. The column space (a.k.a. image or range) of $\boldsymbol{A}$ is a space spanned by its n m-dimensional column vectors

$$\mathbb{C}(\boldsymbol{A}) = span(\boldsymbol{c}_1,\ \ldots,\ \boldsymbol{c}_n) \subseteq \Re^m \tag{2.2}$$

which is an r-dimensional ($r \leq n$ independent columns) subspace of $\Re^m$ composed of all possible linear combinations of its n column vectors. The row space of $\boldsymbol{A}$ is a space spanned by its m n-dimensional row vectors

$$\mathbb{R}(\boldsymbol{A}) = span(\boldsymbol{r}_1,\ \ldots,\ \boldsymbol{r}_m) \subseteq \Re^n \tag{2.3}$$

which is an r-dimensional subspace of $\Re^n$ composed of all possible linear combinations of its m row vectors. The left null space of $\boldsymbol{A}$ ($\mathbb{N}(\boldsymbol{A})$), is the set of all $\boldsymbol{x}$ that satisfy the homogeneous equation

$$\mathbb{N}(\boldsymbol{A}) = \{\boldsymbol{x} \in \Re^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\} \subseteq \Re^n \tag{2.4}$$

and similarly the right null space of $\boldsymbol{A}$ ($\mathbb{N}(\boldsymbol{A}^T)$) is the set of all $\boldsymbol{b}$ that satisfy

$$\mathbb{N}(\boldsymbol{A}^T) = \{\boldsymbol{b} \in \Re^m : \boldsymbol{A}^T\boldsymbol{b} = \boldsymbol{0}\} \subseteq \Re^m. \tag{2.5}$$

The following properties between the different subspaces hold for an operator $\boldsymbol{A}$ ($\perp$ stand for orthogonal complement and the symbol $\oplus$ denotes the direct sum)

$$\begin{gathered} \mathbb{R}(\boldsymbol{A}) = \mathbb{C}(\boldsymbol{A}^T),\ \mathbb{C}(\boldsymbol{A}) = \mathbb{R}(\boldsymbol{A}^T) \\ \mathbb{R}(\boldsymbol{A}) \cap \mathbb{N}(\boldsymbol{A}) = \emptyset,\ \mathbb{R}(\boldsymbol{A}) \perp \mathbb{N}(\boldsymbol{A}),\ \mathbb{R}(\boldsymbol{A}) \oplus \mathbb{N}(\boldsymbol{A}) = \Re^n \\ \mathbb{C}(\boldsymbol{A}) \cap \mathbb{N}(\boldsymbol{A}^T) = \emptyset,\ \mathbb{C}(\boldsymbol{A}) \perp \mathbb{N}(\boldsymbol{A}^T),\ \mathbb{C}(\boldsymbol{A}) \oplus \mathbb{N}(\boldsymbol{A}^T) = \Re^m. \end{gathered} \tag{2.6}$$

The general solution of Equation 2.1 in the underdetermined case ($n > m$), since this is how redundancy is presented, is of the form

$$\boldsymbol{x} = \boldsymbol{x}_{\parallel} + \boldsymbol{x}_{\perp} \tag{2.7}$$

where $\boldsymbol{x}_{\parallel} \in \mathbb{R}(\boldsymbol{A})$ is a particular solution and $\boldsymbol{x}_{\perp} \in \mathbb{N}(\boldsymbol{A})$ is an arbitrary vector that belongs to null space of $\boldsymbol{A}$ ($\boldsymbol{A}\boldsymbol{x}_{\perp} = \boldsymbol{0}$). The relations between the different subspaces can be summarized in Figure 2.1.

A projection operator $\boldsymbol{P}$ is a linear transformation from a vector space to itself, for which the idempotence property holds ($\boldsymbol{P}^2 = \boldsymbol{P}$). The operator is termed orthogonal if $\boldsymbol{P} = \boldsymbol{P}^T$ (Fisher and Mujtaba, 1991), which is equivalent to the range and the null space being mutually orthogonal. We continue by defining four

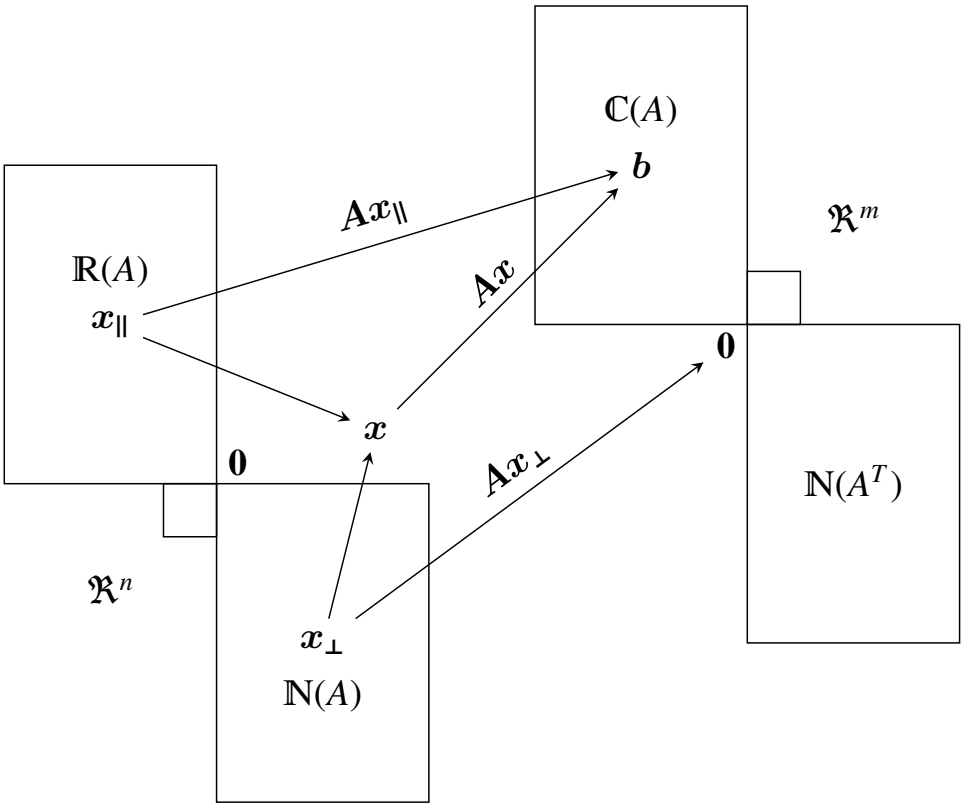


Figure 2.1: Various relations between the different subspaces associated with matrix $\boldsymbol{A}$. Note that the null space solutions have zero contribution when projected by $\boldsymbol{A}$.

projection operators, their properties and relations among the different subspaces for a general system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

Definition 2.1. *[Projection operators]*
Let $\boldsymbol{T_A} = \boldsymbol{A}\boldsymbol{A}^+$ ($\boldsymbol{T}$ is used if $\boldsymbol{A}$ is a tall matrix) and $\boldsymbol{F_A} = \boldsymbol{A}^+\boldsymbol{A}$ ($\boldsymbol{F}$ is used if $\boldsymbol{A}$ is a fat matrix) be two orthogonal projection operators derived from matrix $\boldsymbol{A}$, where $\boldsymbol{A}^+$ is the Moore-Penrose Pseudoinverse (MPP) matrix. The following assertions hold true:

(a) $\boldsymbol{T}_{\boldsymbol{A}}^T = \boldsymbol{T_A}$ *and* $\boldsymbol{T}_{\boldsymbol{A}}^2 = \boldsymbol{T_A}$.

(b) $\boldsymbol{F}_{\boldsymbol{A}}^T = \boldsymbol{F_A}$ *and* $\boldsymbol{F}_{\boldsymbol{A}}^2 = \boldsymbol{F_A}$.

(c) $\boldsymbol{T_A}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{F_A} = \boldsymbol{A}$ *and* $\boldsymbol{A}^+\boldsymbol{T_A} = \boldsymbol{F_A}\boldsymbol{A}^+ = \boldsymbol{A}^+$.

(d) $\boldsymbol{T_A}$ *is the projector onto the* $\mathbb{C}(\boldsymbol{A})$.

(e) $\boldsymbol{F_A}$ *is the projector onto the* $\mathbb{R}(\boldsymbol{A})$.

(f) $\boldsymbol{N_{T_A}} = (\boldsymbol{I} - \boldsymbol{T_A})$ *is the projector onto the* $\mathbb{N}(\boldsymbol{A}^T)$.

(g) $\boldsymbol{N_{F_A}} = (\boldsymbol{I} - \boldsymbol{F_A})$ *is the projector onto the* $\mathbb{N}(\boldsymbol{A})$.

(h) $\boldsymbol{N_{T_A}}\boldsymbol{A} = \boldsymbol{0}$ *and* $\boldsymbol{A}\boldsymbol{N_{F_A}} = \boldsymbol{0}$ *are the left and right null spaces, respectively.*

Proof. The following proofs rely on the Singular Value Decomposition (SVD) theorem ($\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, $\boldsymbol{A}^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^T$) and the properties of the MPP (Definition A.2).

(a) $\boldsymbol{T}_{\boldsymbol{A}}^T = \boldsymbol{T_A}$ and $\boldsymbol{T}_{\boldsymbol{A}}^2 = \boldsymbol{T_A}$.

$$(\boldsymbol{A}\boldsymbol{A}^+)^T = \boldsymbol{A}\boldsymbol{A}^+$$
$$(\boldsymbol{A}\boldsymbol{A}^+)(\boldsymbol{A}\boldsymbol{A}^+) = \boldsymbol{A}(\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+) = \boldsymbol{A}\boldsymbol{A}^+$$

where $(\boldsymbol{A}\boldsymbol{A}^+)^T = \boldsymbol{A}\boldsymbol{A}^+$ and $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{A}^+$ are properties of the MPP (Definition A.2).

(b) $\boldsymbol{F}_{\boldsymbol{A}}^T = \boldsymbol{F_A}$ and $\boldsymbol{F}_{\boldsymbol{A}}^2 = \boldsymbol{F_A}$. Similar to (a) with $(\boldsymbol{A}^+\boldsymbol{A})^T = \boldsymbol{A}^+\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$ (Definition A.2).

(c) $\boldsymbol{T_A}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{F_A} = \boldsymbol{A}$ and $\boldsymbol{A}^+\boldsymbol{T_A} = \boldsymbol{F_A}\boldsymbol{A}^+ = \boldsymbol{A}^+$ can be proven using $\boldsymbol{A}\boldsymbol{A}^+\boldsymbol{A} = \boldsymbol{A}$ and $\boldsymbol{A}^+\boldsymbol{A}\boldsymbol{A}^+ = \boldsymbol{A}^+$.

(d) $\boldsymbol{T_A}$ is the projector onto the $\mathbb{C}(\boldsymbol{A})$.

$$\begin{aligned}\boldsymbol{T_A} = \boldsymbol{A}\boldsymbol{A}^+ &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^T = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^+\boldsymbol{U}^T \\ &= \begin{bmatrix}\boldsymbol{U}_r & \boldsymbol{U}_n\end{bmatrix}\begin{bmatrix}\boldsymbol{1}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0}_n\end{bmatrix}\begin{bmatrix}\boldsymbol{U}_r^T \\ \boldsymbol{U}_n^T\end{bmatrix} = \boldsymbol{U}_r\boldsymbol{U}_r^T\end{aligned}$$

where $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and r is the rank and n the nullity of $\boldsymbol{A}$.

(e) $\boldsymbol{F_A}$ is the projector onto the $\mathbb{R}(\boldsymbol{A})$.

$$\begin{aligned}\boldsymbol{F_A} = \boldsymbol{A}^+\boldsymbol{A} &= \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{\Sigma}\boldsymbol{V}^T \\ &= \begin{bmatrix}\boldsymbol{V}_r & \boldsymbol{V}_n\end{bmatrix}\begin{bmatrix}\boldsymbol{1}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0}_n\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_r^T \\ \boldsymbol{V}_n^T\end{bmatrix} = \boldsymbol{V}_r\boldsymbol{V}_r^T\end{aligned}$$

where $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$.

(f) $\boldsymbol{N_{T_A}} = (\boldsymbol{I} - \boldsymbol{T_A})$ projects onto the $\mathbb{N}(\boldsymbol{A}^T)$, because $\mathbb{C}(\boldsymbol{A})$ and $\mathbb{N}(\boldsymbol{A}^T)$ are mutually orthogonal complement (Equation 2.6).

(g) $\boldsymbol{N_{F_A}} = (\boldsymbol{I} - \boldsymbol{F_A})$ projects onto the $\mathbb{N}(\boldsymbol{A})$, because $\mathbb{R}(\boldsymbol{A})$ and $\mathbb{N}(\boldsymbol{A})$ are mutually orthogonal complement (Equation 2.6).

(h) $\boldsymbol{N_{T_A}A} = \mathbf{0}$ and $\boldsymbol{AN_{F_A}} = \mathbf{0}$.

$$\boldsymbol{N_{T_A}A} = \boldsymbol{A} - \boldsymbol{AA}^{+}\boldsymbol{A} = 0$$

$$\boldsymbol{AN_{F_A}} = \boldsymbol{A} - \boldsymbol{AA}^{+}\boldsymbol{A} = 0$$

□

2.2 Musculoskeletal Modeling

Mechanics is that area of science concerned with the behavior of physical bodies when subjected to forces or displacements, and the subsequent effects of the bodies on their environment. Classical mechanics is the study of statics and dynamics, where dynamics is the study of kinematics and kinetics. In other words, dynamics is the study of the motions of bodies and statics is the study of forces in the absence of changes in motion. On the other hand, kinematics is the study of motion without considering its possible causes (forces) and kinetics is the study of the possible causes of motion. Furthermore, this subject can be subdivided into three disciplines, namely rigid-body, deformable-body and fluid mechanics, whereas the former is the topic of this section under the scope of human biomechanics.

> "Biomechanics is the study of the structure and function of biological systems by means of the methods of mechanics", *–Herbert Hatze*

Musculoskeletal systems provide humans the ability to move by means of interaction between the muscular and skeletal structures. To quantify and understand these interactions it is necessary to define comprehensive models of the different subsystems and combine them using the principle of physics and physiology. The modeling of the skeletal system relies on the same principles used in mechanical engineering and multibody analysis. Unlike their man-made counterparts, however, human joints rarely exhibit the simple, uncoupled motion that is engineered into revolute and prismatic joints (Seth et al., 2011). Body segment parameters, such as length, area, volume, mass, density, center of mass, moment of inertia and center of volume are fundamental to the application of mechanics to the understanding of human movement. On the other hand, the muscular system defines a relationship between the muscle excitation originating from the CNS and the nonlinear force generation produced by the muscles, transmitted on vertebrate limbs via tendons (Valero-Cuevas, 2009b). This perspective has important consequences for the understanding of healthy function, disability and rehabilitation.

The dynamics of musculoskeletal systems is described by the EoMs that define the relationship between the forces acting on the system and the accelerations they produce. A dynamics algorithm is a procedure for calculating the numeric values of quantities that are relevant to the dynamics (Featherstone, 2007). We

will be concerned mainly with algorithms for two particular calculations, namely FD and ID. The former involves the calculation of the acceleration response of a given musculoskeletal system to a given applied force and is used for simulation. The latter concerns on the calculation of the force that must be applied to a given musculoskeletal system in order to produce a given acceleration response and is used mainly for analysis and control. In chapter 4, we will consider a third kind of calculation, called MD, where the FD method is used for simulating the model response, with the input provided by an ID model-based controller representing the underlying system.

In the following chapters, we will be using complex full body musculoskeletal models, such as `gait2354` (Yamaguchi and Zajac, 1989; Delp et al., 1990; Anderson and Pandy, 1999) and more recent (Rajagopal et al., 2016), that are a product of decades of research. They have been extensively validated and used in a variety of movement analyses and are an excellent choice for demonstrating new algorithms. The models are implemented in `OpenSim` (Delp et al., 2007), which is an open-source framework for modeling and simulation of musculoskeletal systems. Occasionally, we will present the concepts of the proposed algorithms using simplified models to enhance the interpretation of the results. For example, section A.4 presents a planar arm model that has three DoFs and nine muscles, some of them being bi-articular. This model has been used in chapter 5 and chapter 6 in conjunction with a more detailed gait model. In order to introduce the basic concept of musculoskeletal modeling and simulation, we will build a realistic human eye model in section 2.4, that can be used for kinematics and dynamics analysis (Filip et al., 2018).

2.2.1 Skeletal Modeling

The purpose of skeletal modeling is to encode the physiological and anatomical properties of the system under study into the EoMs. The process starts with the definition of the body segments their dimension, mass, moment of inertia and center of mass. The next step is to constrain the motion of the bodies by establishing the kinematic restrictions according to the anatomical definition of the joints and the permissible DoFs. The joints are responsible for the kinematic constraints in the system, so the term "joint" imposes a kinematic relationship between a pair of rigid bodies. This description will result in an open or a closed kinematic tree when the different components are assembled.

When the topology of the model is defined, the EoMs for the whole system can be assembled. Two of the most popular approaches are the symbolic and iterative methods (Featherstone, 2007). The former is commonly obtained by applying the principles of Lagrangian and Hamiltonian mechanics (Sussman and Wisdom, 2001) or using Kane's method (Purushotham and Anjeneyulu, 2013). The latter does not form the analytical expression of the EoMs, but solves them iteratively either in an ID, using $O(n)$ Newton-Euler algorithm, or in a FD manner, using the $O(n)$ articulated-body algorithm. Clearly, the iterative approach seems more attrac-

tive, especially when the system has many DoFs, and it is universally applicable to any model. Regardless of the choice the formal representation of the unconstrained EoMs has the following form

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{2.8}$$

where $\boldsymbol{M} \in \Re^{n \times n}$ denotes the symmetric, positive definite joint space inertia mass matrix, n the number of model DoFs and $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \Re^n$ the joint space generalized coordinates and their derivatives with respect to time. The term $\boldsymbol{f} \in \Re^n$ models all internal and external applied forces (e.g., gravity, Coriolis, GRF, etc.), whereas $\boldsymbol{\tau} \in \Re^n$ the vector of applied generalized forces that actuate the model. Most of the quantities in the equations are a function of the generalized coordinates and derivatives, thus this dependency will be omitted for simplicity. The forward and inverse dynamics calculations can be encapsulated by a pair of functions, fd and id in order to depict the input and output relationship

$$\ddot{\boldsymbol{q}} = fd(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}; model) \tag{2.9}$$

and

$$\boldsymbol{\tau} = id(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}; model). \tag{2.10}$$

2.2.2 Constraint Modeling

Motion constraints are an important part of musculoskeletal systems, since biological models, such as anatomical joints (Seth et al., 2010), closed kinematic chains and interaction with the environment (e.g., crank-rotation tasks (Davoudabadi Farahani et al., 2016)) are modeled through constraints. The fundamental equations of differential variational principles for a constrained mechanical system lead to the following Differential Algebraic Equations (DAE) of index-3 ((Eich, 1993))

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{f} + \boldsymbol{\tau}_c = \boldsymbol{\tau} \tag{2.11}$$

$$\phi(\boldsymbol{q}) = \boldsymbol{0} \tag{2.12}$$

where $\boldsymbol{\tau}_c \in \Re^n$ represents the generalized forces induced by constraints. Equation 2.12 corresponds to a set of c constraint algebraic equations which can be differentiated twice with respect to time

$$\boldsymbol{\Phi}\dot{\boldsymbol{q}} = \boldsymbol{0}, \ \boldsymbol{\Phi} = \frac{\partial \phi}{\partial \boldsymbol{q}} \tag{2.13}$$

and

$$\boldsymbol{\Phi}\ddot{\boldsymbol{q}} = -\dot{\boldsymbol{\Phi}}\dot{\boldsymbol{q}} = \boldsymbol{b} \tag{2.14}$$

where $\boldsymbol{\Phi} \in \Re^{c \times n}$ is the constraint Jacobian.

At this point it is worth considering the effect of Equations 2.12, 2.13 and 2.14 on the permissible configuration solutions (Equation 2.12). The algebraic constraint equations imply that any admissible configuration $\boldsymbol{q} \in \Re^n$ should lie on the constraint manifold (Aghili, 2005; De Sapio and Park, 2010) (Figure 2.2). The first derivative of the constraints (Equation 2.13) suggests that $\dot{\boldsymbol{q}} \in \mathbb{N}(\boldsymbol{\Phi})$, following the definition of null space (Equation 2.4). As a consequence of the D'Alembert's principle, constraint forces do no work under any virtual displacement that satisfy them ($\boldsymbol{\tau}_c \delta \boldsymbol{q} = \boldsymbol{0}, \forall \delta \boldsymbol{q} \in \mathbb{N}(\boldsymbol{\Phi})$ which is equivalent to $\boldsymbol{\tau}_c \perp \delta \boldsymbol{q}, \forall \delta \boldsymbol{q} \in \mathbb{N}(\boldsymbol{\Phi})$). In turn, this implies that $\boldsymbol{\tau}_c \in \mathbb{N}(\boldsymbol{\Phi})^{\perp} = \mathbb{C}(\boldsymbol{\Phi}^T)$. Thus, the generalized constraint forces can be represented as a linear combination of columns of $\boldsymbol{\Phi}^T$

$$\boldsymbol{\tau}_c = \boldsymbol{\Phi}^T \boldsymbol{\lambda} \tag{2.15}$$

where $\boldsymbol{\lambda} \in \Re^c$ stands for the vector of Lagrange multipliers. While $\dot{\boldsymbol{q}}_{\|} = \boldsymbol{0}$, where $\dot{\boldsymbol{q}}_{\|} = \{\dot{\boldsymbol{q}} : \dot{\boldsymbol{q}} \in \mathbb{N}(\boldsymbol{\Phi})^{\perp}\}$, the same does not always hold for the acceleration ($\ddot{\boldsymbol{q}}_{\|} \neq \boldsymbol{0}$) by inspecting the second derivative of the constraints (Equation 2.14).

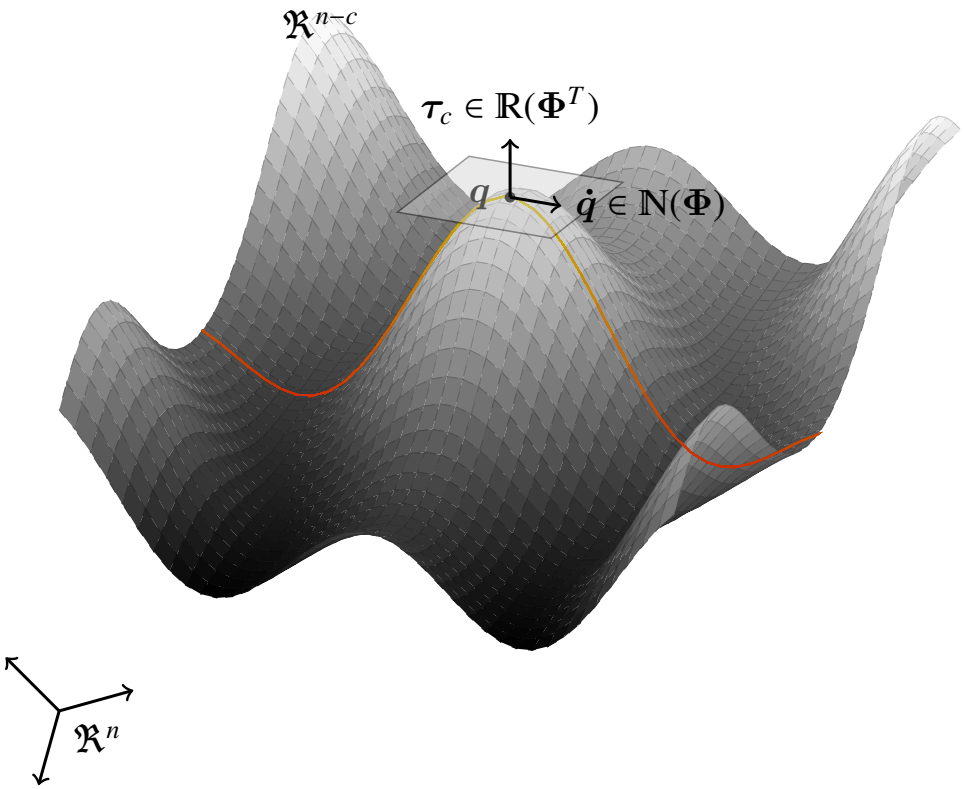


Figure 2.2: In the presence of constraints, all permissible configurations lie on the constraint manifold. To satisfy these restrictions, the constraint forces must be applied. These forces do no work under any virtual displacement that satisfy the constraints.

Equation 2.11 can be solved in a FD manner for the unknowns $\ddot{\boldsymbol{q}}$, $\boldsymbol{\lambda}$, through the complement of Equations 2.12, 2.13 and 2.14. In chapter 4, we will present a method for projecting the EoMs into the permissible configuration space using the notion of null space. The constraint derivatives can reduce the DAE to index-1, making the system solvable by an Ordinary Differential Equations (ODE) solver. However, numerical integration inevitably leads to drifts that eventually result in constraint violation ($\boldsymbol{\phi}(\boldsymbol{q}(t)) \neq \mathbf{0}$). Coordinate projection (Eich, 1993) or Baumgarte's stabilization (Baumgarte, 1972) can be introduced to ensure exponential elimination of the constraint error, given the appropriate initial conditions.

2.2.3 Muscle Dynamics

Integration of muscles into the EoMs is a challenging task, because the modeler has to consider not only the nonlinear dynamics of the muscles, but also their complex geometric interaction with the skeletal system. There are two major classes of muscle models that have been used in biomechanics and motor control, namely Hill-type and Huxley-type. The dynamics of the Hill-type model are phenomenological, meaning that mathematical functions are used to reproduce experimental observations of the activation dynamics, Force-Length (F-L), Force-Velocity (F-V) characteristics (Zajac, 1993; Thelen, 2003; Millard et al., 2013) as well as other nonlinear effects, such as muscle fatigue (Ackermann and van den Bogert, 2010). As for the Huxley-type model, the dynamics capture the cross-bridge process more closely (Huxley, 1974), producing physiologically plausible results. However, this model is computationally demanding and model parameters are not easily obtained or scaled across different muscles as is the case for the Hill-type model.

A typical three element Hill-type muscle model is presented in Figure 2.3, highlighting some of the most important physiological factors that influence its force production. The maximum active force a muscle can develop varies nonlinearly with its length, represented by the active F-L curve $f^L(\tilde{l}^M)$, peaking at a force of f_o^M at the optimal fiver length l_o^M. The tilde notation is used to denote forces, velocities, muscle lengths, and tendon lengths that are normalized by the parameters of the model f_o^M, $v_{\max}^M$, f_o^M and f_s^T, enabling scaling across different muscles (Zajac, 1993). During non-isometric contractions, the force developed by muscle varies nonlinearly with its rate of lengthening, which is represented by the F-V curve $f^V(\tilde{v}^M)$. Force is also developed when the muscle is stretched beyond a threshold length, regardless of whether the muscle is activated, which is represented by the Passive-Force-Length (F-PE) $f^{PE}(\tilde{l}^M)$. Formally, the muscle force f^M is computed using the characteristic curves as follows

$$f^M(a, \tilde{l}^M, \tilde{v}^M) = f_o^M(a f^L(\tilde{l}^M) f^V(\tilde{v}^M) + f^{PE}(\tilde{l}^M)) \tag{2.16}$$

where a is the muscle activation ranging from 0 to 1. A muscle can neither generate force nor relax instantaneously, thus activation dynamics are modeled using first-order ODE

$$\frac{da}{dt} = \frac{u-a}{\tau(a,u)},\ \tau(a,u) = \begin{cases} \tau_{\text{act}}(0.5+1.5a), & u > a \\ \tau_{\text{deact}}/(0.5+1.5a), & u \le a \end{cases} \tag{2.17}$$

where u represents the muscle excitation, $\tau_{\text{act}} = 10$ms, $\tau_{\text{deact}} = 40$ms the activation and deactivation time constants, respectively.

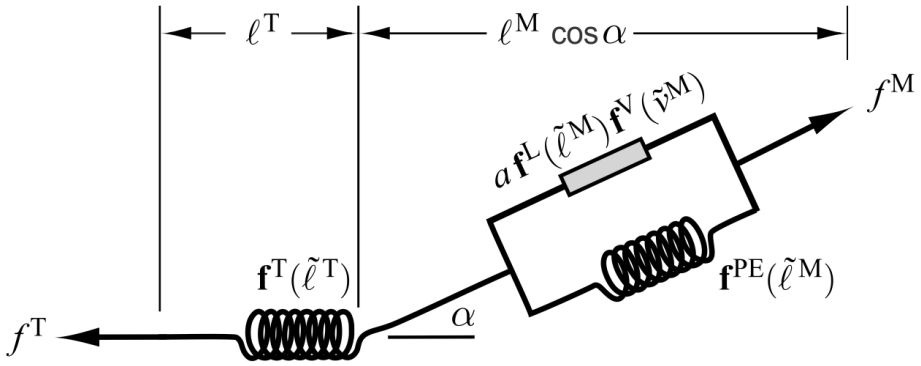


Figure 2.3: A three element Hill-type model. The contractile element is in parallel with a passive element, and they are in series with the tendon (image courtesy (Millard et al., 2013)).

Muscles attach bone through tendons. Since a long tendon may stretch appreciably beyond its slack length (f_s^T) when under tension, the tendon is modeled as a nonlinear elastic element developing force, according to the Tendon-Force-Length (F-T) curve $f^T(\tilde{l}^T)$. Muscle fibers attach to tendon at a pennation angle (α), scaling the force they transmit to the tendon. If the tendon is assumed to be elastic and the mass of the muscle is assumed to be negligible, then the muscle force equilibrium equation takes the following form

$$f^M(a, \tilde{l}^M, \tilde{v}^M)\cos(\alpha) = f_o^M f^T(\tilde{l}^T). \tag{2.18}$$

In order to solve these equations numerically, they can be assembled into a set of first-order differential equations (Pandy, 2001)

$$\begin{aligned} \dot{\boldsymbol{a}}_m &= \boldsymbol{g}(\boldsymbol{u}_m, \boldsymbol{a}_m; \boldsymbol{\theta}_1) \\ \dot{\boldsymbol{l}}_m &= \boldsymbol{h}(\boldsymbol{l}_m; \boldsymbol{\theta}_2) \end{aligned} \tag{2.19}$$

where $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are the parameters of the muscles. In turn, the muscle forces are a function the above state variables and muscle model parameters $\boldsymbol{\theta}_3$

$$\boldsymbol{f}_m = \boldsymbol{d}(\boldsymbol{a}_m, \boldsymbol{l}_m, \dot{\boldsymbol{l}}_m; \boldsymbol{\theta}_3). \tag{2.20}$$

Although most Hill-type muscle models share the same mathematical formulation, the characteristic curves differ between models, leading to differences in

force production Figure 2.4. In the case of the Millard muscle model (Millard et al., 2013), each curve has its own fitting routine that accepts parameters that are physiologically meaningful Figure 2.5, allowing the user to change the shape of the curve without worrying about possible numerical singularities.

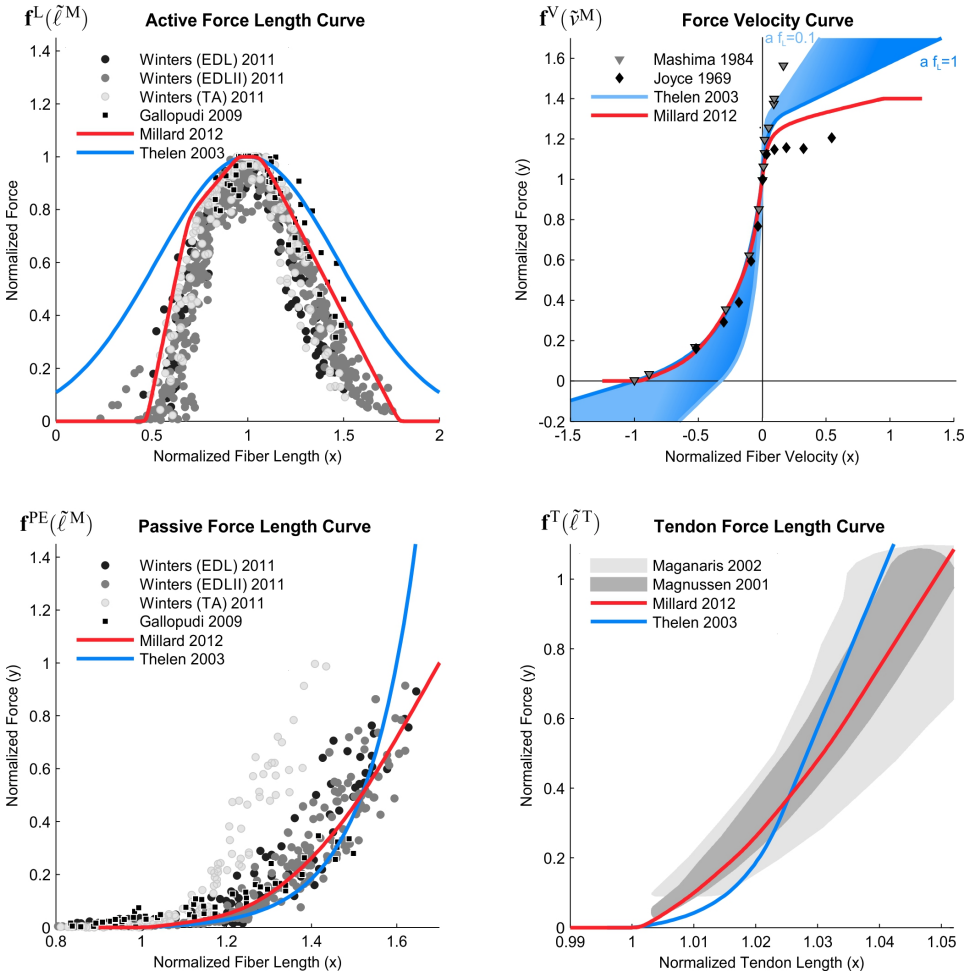
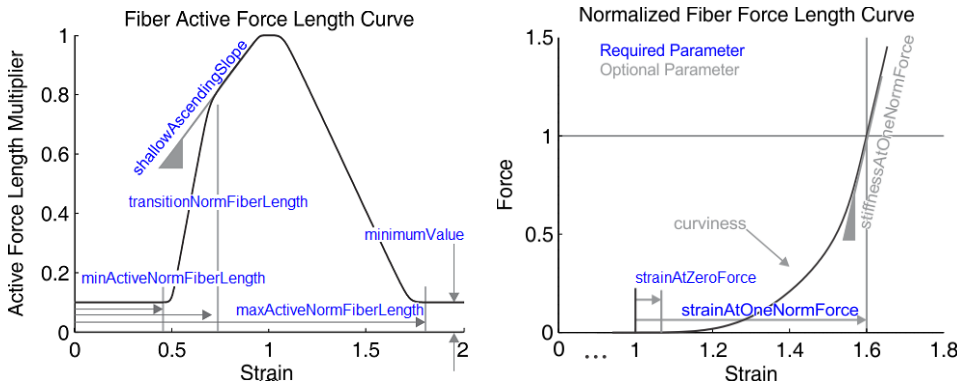


Figure 2.4: Comparison of the characteristic curves (F-L, F-V, F-PE and F-T) between Millard and Thelen muscle models (image courtesy (Millard et al., 2013)).

2.2.4 Muscle Routing

Man-made robotic limbs are driven by motors or pistons that act on the kinematic DoFs either via linkages, cables, or gears. These actuators can exert forces and torques in both clockwise and counterclockwise directions. Unlike torque-driven actuators, tendon-driven actuators can only pull or resist stretching (Valero-Cuevas, 2009b). This asymmetry makes muscles distinctly different from the ideal actuators, which, however complicates the modeling part. While muscles may have



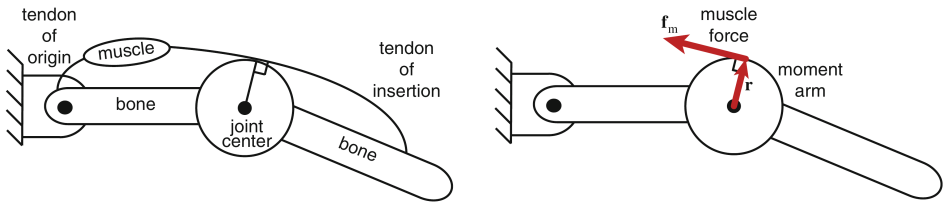
(a) active F-L curve **(b)** F-PE curve

Figure 2.5: The active F-L and F-PE curve definition of the Millard muscle model as implemented in `OpenSim`, (image courtesy `OpenSim` user guide).

some mechanical disadvantage at first sight, it is in fact the opposite, because muscles are lightweight, allowing for remote actuation and flexibility of tendon routing permitting savings in metabolic consumption (Wilson et al., 2001). The scope of this section is to provide a link between the muscle routing properties and the EoMs of the skeletal system.

Let us define a model for mapping muscle forces into joint space generalized forces. Figure 2.6a depicts a muscle that is attached on two bodies and spans a single joint. In Figure 2.6b, the muscle force $\boldsymbol{f}_m$ transmitted by the muscle produces a torque τ, because the muscle crosses the joint at a distance $\boldsymbol{r}$ (moment arm)

$$\tau = \boldsymbol{r} \times \boldsymbol{f}_m. \tag{2.21}$$



(a) **(b)**

Figure 2.6: Model for mapping muscle forces into joint space generalized forces (image courtesy (Valero-Cuevas, 2009b)).

The muscle's path is defined from anatomical landmarks, such as the muscle origin and insertion points. The path does not necessarily follow a straight line, thus intermediate constraints should also be introduced. Muscles can wrap around bones or other muscles as the range of motion changes (Sherman et al., 2013;

Carbone et al., 2015). In more realistic models (Figure 2.7), muscles can span multiple joints and as a result, they can induce acceleration directly or indirectly at multiple levels in the kinematic tree.

Suppose that we have a way to calculate the muscle length $\boldsymbol{l}_m$ as a function of the generalized coordinates ($\boldsymbol{q}$) (pose) accounting for the various routing constraints that must be satisfied

$$\boldsymbol{l}_m = \boldsymbol{f}(\boldsymbol{q}),\ \boldsymbol{l}_m \in \Re^m,\ \boldsymbol{q} \in \Re^n,\ n < m \tag{2.22}$$

then the generalized coordinates fully define the musculotendon lengths, assuming that the muscles are pretensioned. Following this definition, the derivative of Equation 2.22 with respect to time is given by

$$\dot{\boldsymbol{l}}_m = \boldsymbol{R}(\boldsymbol{q})\dot{\boldsymbol{q}},\ \boldsymbol{R}(\boldsymbol{q}) = \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial q_1} & \dots & \frac{\partial \boldsymbol{f}_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{f}_m}{\partial q_1} & \dots & \frac{\partial \boldsymbol{f}_m}{\partial q_n} \end{bmatrix} \tag{2.23}$$

where $\boldsymbol{R}$ is the muscle moment arm ($\boldsymbol{R} : \Re^n \rightarrow \Re^m$), which is always a tall matrix (Proposition A.5), since there are more muscles than joints. Equation 2.23 typically has a sparse structure, because the length of each muscle only depends on the coordinates of the spanning joints.

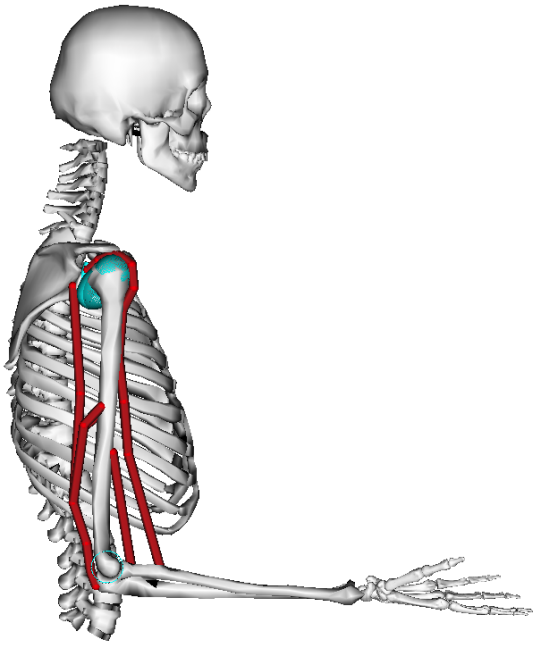


Figure 2.7: An arm model with two DoFs and six muscles. There are muscles that are mono or bi-articular. The muscle path is modeled as a collection of points. Furthermore, the muscles can wrap around the bones as can be seen for the biceps.

From the virtual work principle, we can establish the relationship between joint and muscle space forces

$$\begin{aligned} \boldsymbol{\tau}^T \delta \boldsymbol{q} &= -\boldsymbol{f}_m^T \delta \boldsymbol{l}_m \\ \boldsymbol{\tau}^T \delta \boldsymbol{q} &= -\boldsymbol{f}_m^T \boldsymbol{R} \delta \boldsymbol{q} \\ \boldsymbol{\tau} &= -\boldsymbol{R}^T \boldsymbol{f}_m \end{aligned} \tag{2.24}$$

where the mapping ($\boldsymbol{R}^T : \Re^m \rightarrow \Re^n$) is a fat matrix (Proposition A.4). The negative sign convention is introduced, admitting that a muscle induces positive work during shortening ($\dot{l}_m < 0$). As a result, we can combine Equations 2.11 and 2.24 together, in order to link the muscle forces into the EoMs

$$\begin{aligned} \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{f} + \boldsymbol{\Phi}^T \boldsymbol{\lambda} &= -\boldsymbol{R}^T \boldsymbol{f}_m \\ \boldsymbol{\Phi}\ddot{\boldsymbol{q}} &= \boldsymbol{b}. \end{aligned} \tag{2.25}$$

2.3 Musculoskeletal Simulation and Analysis

Once a model is defined, it can be used to answer scientific questions, test hypotheses or even construct "what if" studies based on the principles of physics and physiology. The usefulness of musculoskeletal models lies in the fact that they can predict quantities, such as muscle forces or joint reaction loads, that are hard or unable to measure in vivo. While it is evident that this approach complements clinical analyses, providing useful information on the kinematic and dynamic evolution of quantities, the accuracy of the obtained results depends heavily on the assumptions that are made. In this section, we will introduce state-of-the-art algorithms used for analyzing the kinematics and dynamics of musculoskeletal systems, with emphasis on the assumptions that are made and their influence on the predicted quantities.

The dynamic analysis of musculoskeletal systems relies on the principle of Newton's second law $f = ma$. This simple relation, states that there are two possible interpretations, namely either the motion (the acceleration) is known and this equation can be solved for the forces that satisfy it (ID) or the forces are known and the resulting motion can be predicted (FD). In the case of musculoskeletal systems the first approach seems more appropriate since the motion of the segments can be measured, while the muscle forces are hard to obtain non-invasively. In the following sections, we will first discuss the ID approach, then we will continue with the FD methods. Finally, in subsection 2.4.2, we will present a simple method to predict the Extraocular Muscles (EMs) coordination patterns necessary to track a desired saccade based on the human eye model.

2.3.1 Scaling

Before initiating any kind of kinematics and dynamics analysis, proper parameterization of the musculoskeletal model must be performed. This is a necessary step since the model outputs are sensitive to modeling parameters (Reinbolt et al., 2005;

Laz and Browne, 2010; Myers et al., 2015). Frequently, one cannot measure individual anthropometrics, thus they are estimated through indirect methods (Winter, 1990). To overcome these challenges generic musculoskeletal models are created and scaled accordingly. The process of scaling is not only about modifying the overall geometric properties of the generic model, but also readjusting the different muscle parameters. Different scaling strategies can be employed in accordance with the available anthropometrics measurements. One of the most common approaches is to measure individual segments, either directly or indirectly from the marker positions, and change the dimensions of the segments accordingly. However, this may be impractical for applications in which the measurements are not available. Another approach would be to scale segment parameters according to the body height and total mass of individual subjects (Dempster, 1955; Drillis et al., 1964; Winter, 1990).

When scaling muscle parameters, such as muscle strength which depends on the volume of a segment, it is important to consider the fat percentage of the body. Between two persons of similar segment masses, the one with a higher fat percentage will have less muscle strength, because the volume otherwise occupied by the muscles is replaced by inactive fat. The resultant value of the fat percentage can be used to compute an estimate of the strength of each muscle in the model (Rasmussen et al., 2005). Generic scaling, as implemented in `OpenSim`, scales only the musculotendon lengths, while fixing other muscle parameters to their original values.

We will introduce IK in the following section, however, it is important to highlight the influence of scaling on the obtained results (Stanev and Moustakas, 2014). Figure 2.8 depicts the different IK errors as reported by `OpenSim` for a gait analysis, comparing whether scaling was performed prior to IK. The three types of errors are computed for the whole gait, providing the mean, max and std values. The marker error is defined as the distance between an experimental marker and the corresponding marker on the model. The coordinate error is the difference between an experimental coordinate value and the coordinate value computed by IK. Notably, the max and mean errors are reduced after scaling and less importantly the std values are higher. A possible explanation for the latter is that in the case of IK without scaling the error is more systematic due to dimensionality mismatch, even so, the benefit of scaling is clearly demonstrated.

2.3.2 Inverse Kinematics

In order to perform any kind of ID analysis the motion of the body segments from the recorded kinematics must be obtained. The motion of the body segments can be recorded using various equipments ranging from marker-based Motion Capture (MoCap) systems to markerless Inertial Measurement Units (IMUs) devices. The next step would be to determine the evolution of the generalized model coordinates that best match the experimentally recorded motion. The IK method goes through each time step of the recorded motion and computes the generalized coordinates

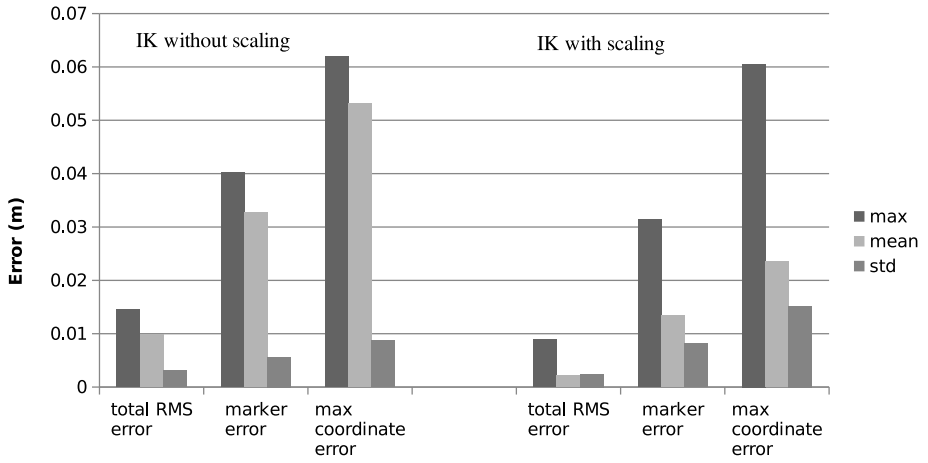


Figure 2.8: IK errors (total Root Mean Square (RMS) error, marker error and max coordinate error) improved after applying scaling to adjust the general model to subject's size.

which positions the model in a pose that best matches the experimental marker. More formally, this is expressed as a weighted least squares problem, whose solution aims to minimize both marker and coordinate errors

$$\underset{\boldsymbol{q}}{\text{minimize}} \quad \sum_{i}^{\text{markers}} w_i \|\boldsymbol{x}_i^{\text{exp}} - \boldsymbol{x}_i(\boldsymbol{q})\|^2 + \sum_{j}^{\text{coordinates}} \omega_j (q_j^{\text{exp}} - q_j)^2 \tag{2.26}$$

where w_i represents the weight for the error term i, $\boldsymbol{x}_i^{\text{exp}}$ the experimental marker position, $\boldsymbol{x}_i(\boldsymbol{q})$ the model marker position that depends on the pose, ω_j the coordinate weight factor and q_j^{exp}, q_j the experimental and model coordinates, respectively. The marker error is the distance between an experimental marker and the corresponding marker on the model. Each marker has a weight associated with it, specifying the influence of the particular term on the overall error. A coordinate error is the difference between an experimental coordinate value and the value computed by IK, that is optionally included if there is a priori knowledge on particular coordinates. The least squares problem can be solved using a general quadratic programming solver.

2.3.3 Inverse Dynamics

For the given kinematics describing the movement of a model and any externally applied force, the ID method determines the generalized forces (e.g., net forces and torques) for each coordinate that satisfy the movement. More formally, Equation 2.8 is solved for the unknown $\boldsymbol{\tau}$ provided $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$. Since $\boldsymbol{q}$ is calculated from IK, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ must be obtained using numerical differentiation. Discontinuities

in the generalized coordinates can lead to numerical singularities during the evaluation of higher order derivatives and it is thus advised to apply filters to remove noise and avoid artifacts (De Groote et al., 2008).

When ID is solved, it is preferred to ignore model constraints since $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ may not necessarily satisfy the constraint algebraic equations. For example, `OpenSim` ignores model constraints when performing ID calculations. As a result, if a model contains kinematic constraints ID may give rise to unreliable estimates of the generalized forces. In this case, FD-based methods may provide a suitable solution. Another important misconception is that ID does not necessarily depend on externally applied forces, however, it is a common sense that they should be accounted for since their omission will significantly alter the result.

2.3.4 Forward Dynamics

In contrast to ID, FD solves the EoMs for the unknown $\ddot{\boldsymbol{q}}$. For a given set of model inputs (e.g., muscle excitations) and initial conditions the system of equations is numerically integrated in order to compute the resulting generalized coordinates and velocities. Equation 2.25 assumes that the position variables are the integrals of the velocity variables, however, this is not always the case. For example, if the system contains nonholonomic constraints, then it will have more position than velocity variables. In this case, Equation 2.25 can be adapted simply by replacing $\dot{\boldsymbol{q}}$ with $\boldsymbol{\alpha}$

$$\begin{aligned} \boldsymbol{M}\dot{\boldsymbol{\alpha}} + \boldsymbol{f} + \boldsymbol{\Phi}^T\boldsymbol{\lambda} &= -\boldsymbol{R}^T\boldsymbol{f}_m \\ \dot{\boldsymbol{q}} &= \boldsymbol{Q}(\boldsymbol{q})\boldsymbol{\alpha} \\ \boldsymbol{\Phi}\ddot{\boldsymbol{q}} &= \boldsymbol{b} \end{aligned} \tag{2.27}$$

where $\boldsymbol{Q}$ is a matrix that depends on $\boldsymbol{q}$. For example, the use of quaternion representation for orientation results in there being more $\boldsymbol{q}$'s than $\boldsymbol{a}$'s, thus $\boldsymbol{Q}$ may not necessarily be an identity matrix (Sherman, 2013). As most numerical integration solvers expect a set of first-order differential equations ($\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \cdots)$), this transformation also reduces the order of the differential equations. Combining Equation 2.27 and the equations for muscle dynamics (Equation 2.19) the first-order FD model can be derived

$$\begin{aligned} \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\alpha}} \\ \dot{\boldsymbol{a}}_m \\ \dot{\boldsymbol{l}}_m \end{bmatrix} &= \begin{bmatrix} \boldsymbol{Q}\boldsymbol{\alpha} \\ -\boldsymbol{M}^{-1}(\boldsymbol{R}^T\boldsymbol{f}_m + \boldsymbol{f} + \boldsymbol{\Phi}^T\boldsymbol{\lambda}) \\ \boldsymbol{g} \\ \boldsymbol{h} \end{bmatrix} \\ \boldsymbol{\Phi}\ddot{\boldsymbol{q}} &= \boldsymbol{b} \end{aligned} \tag{2.28}$$

which can be numerically integrated provided appropriate initial conditions $\boldsymbol{x}_0$ and muscle excitation $\boldsymbol{u}_m$ inputs (Sherman et al., 2011; Seth et al., 2011).

The practical use of FD for musculoskeletal simulation may not be evident at first sight, because muscle excitations are not easily measured or it is more diffi-

cult to model the process that generates them. FD methods for multibody systems are not only technically more difficult to implement as compared to ID methods, but they are also more demanding from a computational point of view (Featherstone, 2007). However, FD is a key component in some algorithmic design, such as Computed Muscle Control (CMC) (subsection 2.3.5), providing a solution that is unattainable purely by an ID scheme (Erdemir et al., 2007). In chapter 4, we present a MD scheme to solve the motion planning problem, where we employ an ID model-based controller that drives a musculoskeletal model in a FD manner. Another class of algorithms, that is worth mentioning, is the Hybrid Dynamics (HD) method (Featherstone, 2007), in which some accelerations and force variables are given and the task is to calculate the rest. A useful approach in cases where it is desirable to evaluate the effect of forces, such as knee ligaments and contact impulses, on the motion of the bodies predicatively.

2.3.5 Muscle Optimization

Muscle optimization is the process used to calculate the muscle excitation or activation patterns required by the model in order to reproduce the experimental kinematics and kinetics. There are two major categories of algorithms, namely static and dynamic optimization (Anderson and Pandy, 2001). Static Optimization (SO) is an extension of the ID method that further resolves the net joint moments into individual muscle forces at each instant in time independently. The muscle forces are resolved by minimizing the sum of squared (or other power) muscle activations or some other objective criterion. Dynamic Optimization (DO) is a FD method that integrates system dynamics in the solution process. Quantities like muscle forces and performance criterion are treated as time-dependent state variables, whose behavior is governed by the differential equations. The resulting optimization problem becomes computationally challenging to solve, due to non-linearity and stiff differential equations (De Groote et al., 2009).

SO is an ID method for determining the muscle forces required to satisfy the motion and forces of the model

$$-\boldsymbol{R}^T \boldsymbol{f}_m = \boldsymbol{\tau} \tag{2.29}$$

which is a system of linear equations. There are infinitely many solutions of muscle forces satisfying this equation provided that the generalized forces are known, as the transposed moment arm is always a fat matrix. Furthermore, the muscles are unilateral elements that can only pull and produce a finite, limited force. From a mathematical point of view, this description cannot be encoded solely into a set of linear equations. SO is traditionally solved by forming an optimization problem

$$\begin{aligned} \underset{\boldsymbol{a}_m}{\text{minimize}} \quad & f(\boldsymbol{a}_m) \\ \text{subject to} \quad & -\boldsymbol{R}^T \boldsymbol{f}_m(\boldsymbol{a}_m) = \boldsymbol{\tau} \\ & \boldsymbol{0} \preceq \boldsymbol{a}_m \preceq \boldsymbol{1} \end{aligned} \tag{2.30}$$

where $f(\boldsymbol{a}_m)$ is some linear or nonlinear function that is being minimized. Many studies have shown that muscles tend to be recruited systematically (Steele et al., 2015; Razavian et al., 2015; Kutch and Valero-Cuevas, 2012). Traditional solutions of the muscle redundancy problem, such as minimum effort criterion, are very popular and have roots in a compelling evolutionary hypothesis, which states that motor control systems evolve to minimize energy expenditure during movement. The sum of muscle activations raised to the power exponent p

$$f(\boldsymbol{a}_m) = \sum_i^m a_i^p \tag{2.31}$$

is commonly adopted, where p is typically given a value of 2.

In contrast to SO, CMC is an DO method (Figure 2.9) that computes the muscle excitation levels that drive the generalized coordinates of a dynamic musculoskeletal model towards a desired kinematic trajectory, using a combination of Proportional Derivative (PD) tracking controller and SO (Thelen and Anderson, 2006). More specifically, the desired acceleration $\ddot{\boldsymbol{q}}(t+T)$ is computed by comparing the experimental and simulated generalized coordinates and derivatives using a PD tracking scheme

$$\ddot{\boldsymbol{q}}(t+T) = \ddot{\boldsymbol{q}}^{\text{exp}}(t+T) + k_p(\boldsymbol{q}^{\text{exp}}(t) - \boldsymbol{q}(t)) + k_d(\dot{\boldsymbol{q}}^{\text{exp}}(t) - \dot{\boldsymbol{q}}(t)) \tag{2.32}$$

where k_p and k_d are feedback tracking gains. This acceleration is provided to the SO method that in this case solves the following problem

$$\begin{aligned} \underset{\boldsymbol{u}_m}{\text{minimize}} \quad & \sum_i^m u_i^2 \\ \text{subject to} \quad & \ddot{\boldsymbol{q}}^* = \ddot{\boldsymbol{q}}(\boldsymbol{u}_m) \\ & \mathbf{0} \preceq \boldsymbol{u}_m \preceq \mathbf{1} \end{aligned} \tag{2.33}$$

where $\boldsymbol{u}_m$ are the muscle excitations and $\ddot{\boldsymbol{q}}(\boldsymbol{u}_m)$ the induced acceleration. The computed excitations are used to actuate the musculoskeletal model and the resulting motion is predicted in a FD manner.

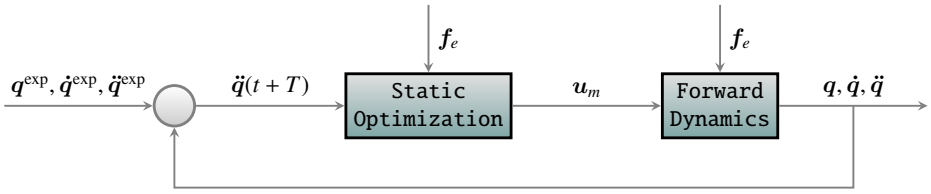


Figure 2.9: Schematic of the CMC method (Thelen and Anderson, 2006).

Although, both SO and DO can find a possible solution to the muscle recruitment problem, a critical assumption must be made, namely to select the appropriate

objective function f that is to be minimized. While the minimum effort criterion may seem reasonable, it is not necessarily the best choice, since the CNS does not orchestrate the muscles necessary in this manner. This is especially evident in the case of rigidity in Parkinson's disease, which is characterized by the inability of the muscles to relax. Clearly, a particular choice of f will not only bias the results, but also will affect the calculation of other quantities that depend on the muscle forces. This motivated us to explore the possibility of formulating the muscle recruitment problem differently, without introducing any prior assumption for f in order to identify the feasible solution space (chapter 5).

2.4 An Open-Source Educational Example

In this section, we will present an open-source educational example that utilizes the concepts of musculoskeletal modeling and simulation for the study of saccadic eye movements. The source code along with any related material for this study are publicly available (section A.1), providing simple examples so that the readers can reproduce, understand and reuse the presented methods.

2.4.1 Eye Modeling

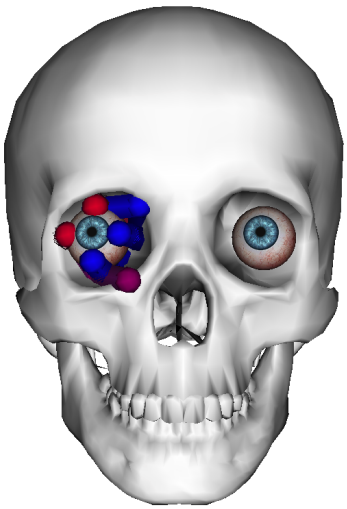


Figure 2.10: An eye model containing six EMs and three DoFs.

The eye model consists of the eye globe, three pairs of EMs and the connective passive tissues (Figure 2.10). The size of an emmetropic human adult eye is approximately 0.0242m (transverse, horizontal), 0.0237m (sagittal, vertical), 0.022 – 0.0248m (axial, anteroposterior) with no significant difference between sexes and age groups. In the transverse diameter, the eye may vary from 0.021m to 0.027m, thus it can be approximated by a solid sphere of radius $r = 0.012$m. The

weight of an average human eye is $m = 0.0075$kg and the moment of inertia can be calculated assuming a spherical homogeneous and isotropic model $I = 2/5mr^2$. The eye has three rotational DoFs, namely incyclotosion-excyclotosion ($\boldsymbol{x}$-axis), adduction-abduction ($\boldsymbol{y}$-axis) and supraduction-infraduction ($\boldsymbol{z}$-axis). More formally, the orientation of the eyes' local frame E with respect to the reference frame G, using homogeneous transformations, is

$$
{}^{G}\boldsymbol{T}_{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_1 & -s_1 & 0 \\ 0 & s_1 & c_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.34}
$$

where $s_i = \sin(q_i)$, $c_i = \cos(q_i)$ and q_1, q_2, q_3 are the generalized coordinates of the model. In this example, the model has only one body and joint, unlike models like the arm (Figure 2.7) that contain more links and joints. When the system is kinematically assembled the joint transformations ${}^{P}\boldsymbol{T}_{S}$ between the predecessor and successor bodies in the kinematic tree are calculated in terms of the model coordinates. Given that, the position and orientation of each body can be measured with respect to a reference frame by applying the chain multiplication rule (${}^{0}\boldsymbol{T}_{5} = {}^{0}\boldsymbol{T}_{1}{}^{1}\boldsymbol{T}_{2} \cdots {}^{4}\boldsymbol{T}_{5}$).

The six EMs, including four rectus muscles and two oblique muscles, are controlled by the cranial nerves in order to track a visual target and stabilize the image of the object of interest on the retina. The Lateral Rectus (LR) and Medial Rectus (MR) muscles form an agonist/antagonist pair that produces horizontal eye movements. The Superior Rectus (SR) and Inferior Rectus (IR) muscles form the vertical agonist/antagonist pair, which mainly controls vertical eye movement and also affects rotation about the line of sight (secondary action) and the horizontal plane (tertiary action). The Superior Oblique (SO) muscle passes through the cartilaginous trochlea attached to the orbital wall, which reflects the SO path by 51°. The Inferior Oblique (IO) muscle originates from the orbital wall anteroinferior to the globe center and inserts on the sclera posterior to the globe's equator. The primary actions of SO and IO causes rotation of the globe around the visual axis, but also affects vertical (secondary action) and horizontal (tertiary action) movements.

The model relies on the passive pulley assumption, which states that the pulleys are fixed to the orbit pulley points. Table 2.1 shows the positions of the origin, insertion and passive muscle pulleys for the EMs, defined in the local body coordinates of the eye globe. The data are based on physiological measurements (Iskander et al., 2018), with some minor modification to prevent unrealistic muscle-surface penetration.

The Millard muscle model has been adopted for the modeling of the EMs, permitting parameterization of the characteristic curves according to the experimental measured data. The muscles were modeled using the rigid tendon assumption that ignores the elasticity of the tendon. This means that the series element of the muscle model is not included (the tendon length l^T is equal to the tendon slack length

Table 2.1: Muscle path points for the six EMs defined in the local frame of the eye globe (dimensions are given in meters).

Muscle	Origin			Pulley			Insertion		
	Ox	***Oy***	***Oz***	***Px***	***Py***	***Pz***	***Ix***	***Iy***	***Iz***
LR	-0.034	0.0006	-0.013	-0.0102	0.0003	0.012	0.0065	0	0.0101
MR	-0.030	0.0006	-0.017	-0.0053	0.00014	-0.0146	0.0088	0	-0.0096
SR	-0.0317	0.0036	-0.016	-0.0092	0.012	-0.002	0.0076	0.0104	0
IR	-0.0317	-0.0024	-0.016	-0.0042	-0.0128	-0.0042	0.00805	-0.0102	0
SO	0.0082	0.0122	-0.0152	-0.030834	0.001145	-0.01644	0.0044	0.011	0.0029
IO	0.0113	-0.0154	-0.0111	-0.00718	-0.0135	0	-0.008	0	0.009

l_s^T). EMs are considered parallel-fibered muscles, so the pennation angle is assumed zero ($\alpha = 0$). The values for the maximum isometric force f_o^M, optimal fiber length l_o^M and tendon length l^T are presented in Table 2.2.

Table 2.2: Millard muscle parameters for the EMs.

Muscle	Maximum Isometric Force (N)	Optimal Fiber Length (m)	Tendon Slack Length (m)	Maximum Contraction Velocity (m / s)
LR	1.4710	0.04898	0.0084	3.8483
MR	1.5740	0.04084	0.0038	4.6155
SR	1.1768	0.04487	0.0054	4.2009
IR	1.4269	0.04549	0.0048	4.1437
SO	0.6031	0.03956	0.0265	4.7648
IO	0.5590	0.04110	0.0015	4.5863

The active F-L and F-PE characteristic curves of the EMs differ significantly from those of a skeletal muscle. As shown in Figure 2.5, we can fine-tune the curve parameters to fit the experimental data available for the LR muscle. The values for the active F-L and F-PE characteristic curves are summarized in Tables 2.3 and 2.4, respectively. We safely assume that the parameters of the characteristic curves for the other EMs are the same.

EMs have a higher fraction of fast twitch fibers and thus different F-V behavior,

Table 2.3: Parameters of the active F-L characteristic curve.

Parameter	Value
min norm active fiber length	0.55
transition norm fiber length	0.7
max norm active fiber length	1.8
shallow ascending slope	2.4
minimum value	0.0

Table 2.4: Parameters of the F-PE characteristic curve.

Parameter	Value
strain at zero force	-0.18
strain at one norm force	0.4

due to different structures compared to skeletal muscles. Despite that, the default Millard F-V curve was used for the six EMs, since the behavior of the selected muscle model depends heavily on the maximum contraction velocity $v^{\max}$. The maximum muscle contraction velocity is tuned to match the peak velocity of saccadic eye movement $\omega^{\max} = 15.7\text{rad/s}$ (900°/s). Following this definition, the maximum muscle contraction velocity is given in optimal fiber length per seconds and it is thus different for each EMs, as their optimal fiber length is different ($v^{\max} = \omega^{\max} r / l_o^M$). Furthermore, since the optic nerve is much shorter that the average muscle nerve, activation and deactivation delays ($\tau_a = \tau_d = 5\text{ms}$) are smaller. Finally, two separate wrapping spheres for the rectus muscles and the oblique muscles were created, to avoid abnormal changes on the F-L curve as the eye rotates.

The passive connective tissues of the orbit apply a restoring force, which brings the eye back to the central position when the net force from the EMs is zero. These tissues include all non-muscular suspensory tissues, such as Tenon's capsule, the optic nerve, the fat pad and the conjunctiva. The force-displacement elasticity force can be represented as

$$\boldsymbol{f}_t = -k\boldsymbol{q} - c\boldsymbol{q}^3 - d\dot{\boldsymbol{q}} \tag{2.35}$$

where, $\boldsymbol{f}_t$ is the passive tissue forces, $k = 2.225 \cdot 10^{-3}\text{Nm/rad}$, $c = 34.53 \cdot 10^{-3}\text{Nm/rad}^3$ and $d = 2 \cdot 10^{-3}\text{Nm/(rad/s)}$ the physiological constants and $\boldsymbol{q}, \dot{\boldsymbol{q}} \in \Re^3$ the rotational coordinates and velocities of the model (Robinson et al., 1969; Collins et al., 1981). These forces are modeled using `OpenSim`'s expression based coordinate force.

2.4.2 Eye Simulation

In the previous section, an anatomically realistic eye model was developed. Here, we will present a fixation controller that calculates the EMs excitations necessary to track a desired saccade. The controller actuates the model in a closed-loop FD manner. The parameters of the controller are the desired horizontal and vertical fixation angles, the saccade onset and velocity, and the gains of PD tracking controller. A sigmoid function is used for generating smooth saccade trajectories in the horizontal and vertical direction, while the torsional component is maintained close to zero. More formally,

$$\begin{aligned} \theta_d(t) &= \frac{a}{2}(\tanh(b(t - t_0)) + 1) \\ \dot{\theta}_d(t) &= \frac{ab}{2}(1 - \tanh^2(b(t - t_0))) \end{aligned} \tag{2.36}$$

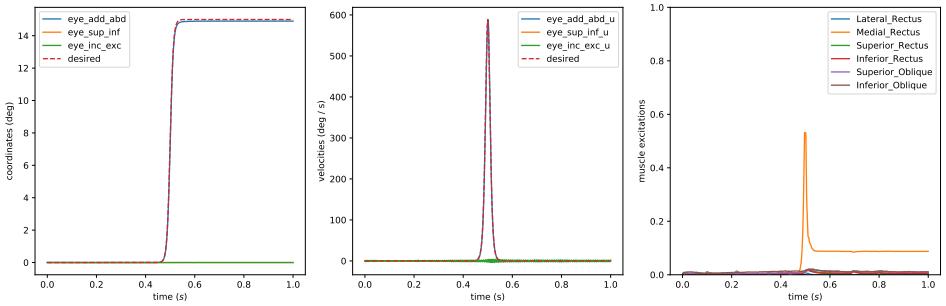
where $\theta_d(t)$ and $\dot{\theta}_d(t)$ represent the desired orientation and velocity at time t, a the magnitude of the trajectory, b the slope and t_0 a time shift constant. Provided a fixation goal θ_g, a desired saccade velocity $\dot{\theta}_g$ and a saccade onset t_g, the parameters

of the sigmoid function are defined as $a = \theta_g$, $b = 2\dot{\theta}_g/\theta_g$ and $t_0 = t_s$. The PD tracking controller has the following form

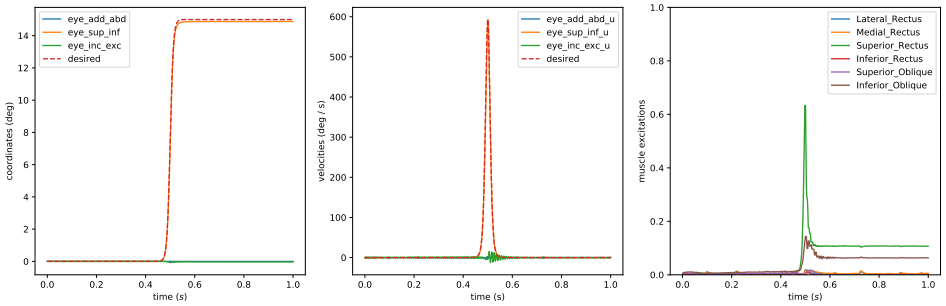
$$u(t) = k_p(\theta_d(t) - \theta(t)) + k_d(\dot{\theta}_d(t) - \dot{\theta}(t)) \tag{2.37}$$

where k_p, k_d are the tracking gains, and $\theta(t)$, $\dot{\theta}(t)$ the simulated response of the model.

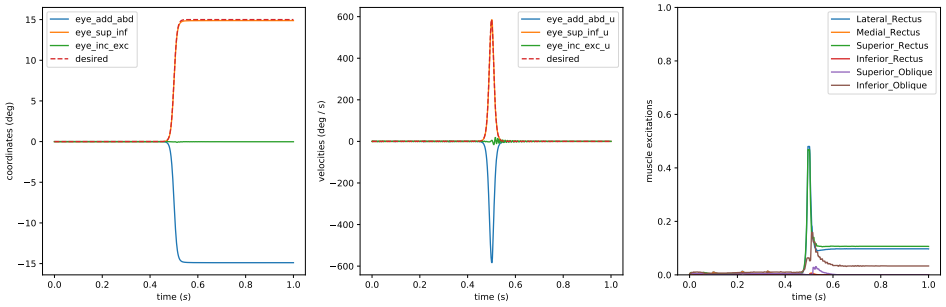
The sign and magnitude of $u(t)$, representing the deviation from the fixation target for each axis of rotation respectively, are used to calculate the muscle excitation levels, by assuming that each individual muscle rotates the eye globe in a particular direction. Figure 2.11 depicts the simulated coordinates, speeds and estimated EMs excitation levels that reproduce the desired saccade trajectory for different model parameters.



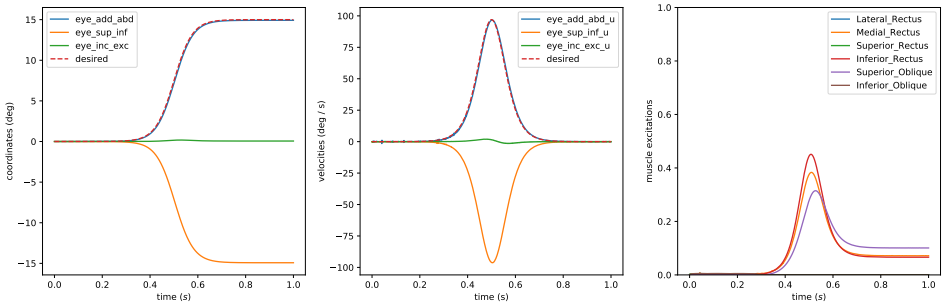
(a) $\theta_H = 15°$, $\theta_V = 0°$, $b = 600°/s$ and $d = 0\text{Nm}/(\text{rad}/\text{s})$



(b) $\theta_H = 0°$, $\theta_V = 15°$, $b = 600°/s$ and $d = 0\text{Nm}/(\text{rad}/\text{s})$



(c) $\theta_H = -15°$, $\theta_V = 15°$, $b = 600°/s$ and $d = 0\text{Nm}/(\text{rad}/\text{s})$



(d) $\theta_H = -15°$, $\theta_V = 15°$, $b = 100°/s$ and $d = 0.002\text{Nm}/(\text{rad}/\text{s})$

Figure 2.11: Simulated saccade response for different model parameters. Left subplot represents the simulated generalized coordinates, middle the generalized velocities and right the estimated EMs excitation levels.

Chapter 3

Anterior Cruciate Ligament Reconstruction Through Personalized Simulation

The objective of this chapter is to develop a Decision Support System (DSS) that can help clinicians to optimally plan the ACL reconstruction procedure based on computational musculoskeletal models. A full body model with twenty-three DoFs and ninety-three muscles was used for the analysis. The knee ligaments were modeled as nonlinear spring elements and a tibiofemoral contact model was utilized. The parameters of the ligaments were calibrated from experimental measurements using an optimization-based procedure. FD was utilized to predict the model's response to a given set of external forces, posture configuration and physiological parameters. The predictions are compared against measurements of the well-known Lachman test on several patients with a torn ACL. The potential of this framework is demonstrated in the context of flexion-extension, gait and jump activities. The clinician is able to modify and fine-tune several parameters, such as the number of bundles, insertion position on the tibia or femur and the resting length that correspond to the choices of the surgical procedure and study their effect on the biomechanical behavior of the knee. Computational knee models can be used to predict the effect of surgical decisions and providing insights on how different parameters can affect the stability of the knee.

Following this case study, we will highlight and discuss two open issues that can make application of musculoskeletal modeling and simulation frequently inappropriate in clinical practice. This cultivated the development of novel methods that address these issues in the following chapters. More specifically, predicting the outcome of a surgical procedure is possible only in a FD setting, yet the muscle excitations that generate a coordinated movement are not known a priori. In chapter 4, we will develop methods for planning virtual simulations of constrained musculoskeletal systems without necessarily relying on experimentally measured kinematics and muscle activity. Moreover, we will discuss the problem of muscu-

loskeletal redundancy and how it can negatively affect the validity of the obtained predictions. In chapter 5, we will develop methods for handling redundancy in the process of modeling, simulation and analysis of musculoskeletal systems, enabling the calculation of the feasible set of solutions. Consequently, we will be able to properly interpret results obtained from the redundant musculoskeletal systems.

3.1 Introduction

The knee joint is distinguished by its compound, three-dimensional geometry and multibody articulations that generate complex mechanical responses under moderate loads (Fregly et al., 2011). The joint compliance and stability required for optimal daily functioning are provided by various articulations, the menisci, ligaments and muscle forces. A complete understanding of knee joint biomechanics can potentially improve the prevention and treatment of disorders and injuries. Total Knee Arthroplasty (TKA), osteoarthritis and prosthetic ligament replacement are examples that directly benefit from such knowledge (Weiss and Gardiner, 2001).

Computational models of human movement provide a framework for integrating facts about the biology of movement, that can be used to predict biomechanics performance, prevent injury and customize treatment methodology in the context of everyday activities. The latter can clearly help as a decision support mechanism, as far as surgical reconstruction is concerned, by evaluating the macroscopic effects of surgical decision in a simulated manner. Subject-specific modeling is a necessity for predicting the reaction loads of the human knee under different loading conditions. Concerning ACL tear, even if surgical reconstruction practice can be nowadays considered advanced, it is mainly based on expert judgment and less on objective biomechanical evaluation of the patient's physiology and everyday activities, while subject-specific surgery customization is very rare (Fardoun et al., 2015). With realistic ligament models and computational modeling techniques, ACL injuries could be potentially analyzed and optimally reconstructed using simulated decision support.

Presurgical objective evaluations is on of the main objective of this study. The proposed scheme aims to provide a feasibility study, demonstrating how simulated Virtual Physiological Humans (VPH) could help towards the aforementioned grand challenge. The main contribution is the development of a six DoFs knee joint, as part of a full 3D multibody model, comprised of ten ligaments and a tibiofemoral contact model (Stanev et al., 2015) (section A.1). The model is customizable, meaning it can be further parameterized based on subject-specific in vivo measurements. Furthermore, it can be used for simulation of different tasks, such as walking, running and jumping, thus making it a valuable tool for assessment of presurgical decision.

3.1.1 Related Work

Finite Element (FE) and rigid body modeling are two common approaches for gaining insights into the musculoskeletal system. In this study, we will utilize the latter approach mainly because we are interested in analyzing the response of the knee under complex activities such as gait, which are hard to solve through a FE analysis. Multibody systems are typically analyzed in a forward or inverse dynamics manner. In the former, the calculation of the acceleration response of a given musculoskeletal system to a given applied force is performed. In the latter, the calculation of the force that must be applied to a given musculoskeletal system in order to produce a given acceleration response is assessed in the context of experimentally measured kinematics and externally applied forces.

Several studies related to the modeling of the knee have been presented, utilizing analytical models with different degrees of sophistication and accuracy. They have mainly attempted to model the tibiofemoral joint (Blankevoort et al., 1991; Blankevoort and Huiskes, 1996; Kar and Quesada, 2013), while a few have included the patellofemoral joint (Piazza and Delp, 2001; Bloemker et al., 2012; Xu et al., 2014). Most of the studies concentrate on modeling partly the knee joint and did not take into account the full body in their simulations. Moreover, few studies consider the influence of the muscle contribution to the laxity of the tibiofemoral joint (Kar and Quesada, 2013).

Various representations of the knee ligaments have been proposed, ranging from simplified nonlinear elastic springs to more sophisticated FE models (Weiss and Gardiner, 2001). Modeling the ligaments as elastic springs is more inexpensive and several studies have looked into their physiological definition (Blankevoort et al., 1991; Blankevoort and Huiskes, 1996). Ligaments have a nonlinear toe region, which occurs because of the initial crimping of the fibers. The toe region ends when all the fibers have become taut (Weiss et al., 2005). At that point the ligament behaves as a linear spring. The proposed framework uses a nonlinear one dimensional spring method to define the cruciate and collateral ligaments.

3.2 Methods

3.2.1 Ligaments Modeling

The knee ligaments, which attach the femur to the tibia or fibula, are very important in stabilizing the joint and prevent injuries. There are four major groups of ligaments distinguished by their functionality: 1) the ACL that primary restraints anterior tibial translation and tibia rotation (relative to the femur), 2) the Posterior Cruciate Ligament (PCL) which restrains posterior translation of the tibia, 3) the Medial Collateral Ligament (MCL) which counteracts valgus instabilities and 4) the Lateral Collateral Ligament (LCL) which restrains varus movements and tibial external rotations.

The ligament has nonlinear and elastic properties and the tension is mainly a

function of its length, where the non-linearity is present for low strains (Blankevoort et al., 1991). Furthermore, its behavior becomes linear for strains higher than a certain level. The force-strain curve is defined as

$$f_l = \begin{cases} 0, & \epsilon \leq 0 \\ \frac{1}{4} k \frac{\epsilon^2}{\epsilon_l}, & 0 \leq \epsilon \leq 2\epsilon_l \\ k(\epsilon - \epsilon_l), & \epsilon > 2\epsilon_l \end{cases} \tag{3.1}$$

where k represents the ligament stiffness and $2\epsilon_l$ the limit at which the ligament moves from the nonlinear to linear region. The strain ϵ is defined by

$$\epsilon = \frac{l - l_0}{l_0} \tag{3.2}$$

where l denotes the length of the ligament and l_0 the zero-load length (the length of the ligament when it first becomes taut). In addition, the resting length can be defined

$$l_0 = \frac{l_r}{e_r + 1} \tag{3.3}$$

where l_r is the reference length of the ligament, usually measured at a reference posture and e_r the reference strain.

3.2.2 Musculoskeletal Modeling and Simulation

The developed model is based on full body `OpenSim` musculoskeletal model (Delp and Loan, 1995; Delp et al., 2007). It consists of twelve rigid segments, twenty-three DoFs and ninety-three muscles partially illustrated in Figure 3.1. The model has been modified in order to enable flexion-extension, varus-valgus, tibial and femoral rotations, anterior-posterior, medial-lateral and inferior-superior translations, that are of significant importance in the proposed framework. In addition to the main muscles, extra patella-tendon muscles were inserted since their contribution to the stability of the joint cannot be neglected. The tibiofemoral contact was modeled using elastic foundation model (Seth et al., 2011), which uses a mesh to represent arbitrary surfaces in contact and calculates deformations and forces using a simplified bed of springs elastic model.

The EoMs of the musculoskeletal system, following the notation presented in section 2.2

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = -\boldsymbol{R}^T(\boldsymbol{q})\boldsymbol{f}_m \tag{3.4}$$

where $\boldsymbol{M} \in \Re^{n \times n}$ denotes the symmetric, positive definite joint space inertia mass matrix, n the number of model DoFs and $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \Re^n$ the joint space generalized coordinates and their derivatives with respect to time. The term $\boldsymbol{f} \in \Re^n$ represents the sum of all internal (e.g., gravity, Coriolis, etc.) and externally applied forces.

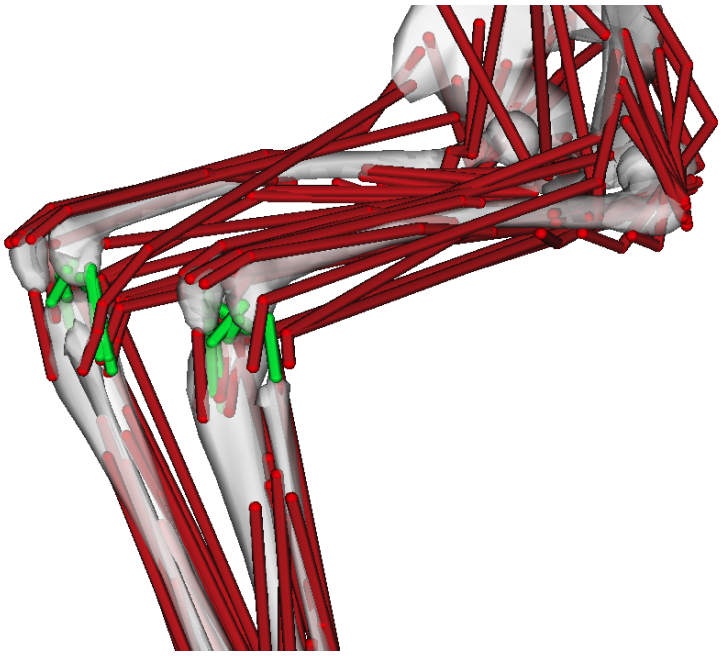


Figure 3.1: A portion of the developed model, with the muscles visualized with red and the ligaments with green.

Term $\boldsymbol{R}(\boldsymbol{q}) \in \Re^{m \times n}$ is the muscle moment arm matrix (Sherman et al., 2013) and $\boldsymbol{f}_m \in \Re^m$ a vector of muscle forces with m being the number of muscles.

The muscle force production is influenced by the activation and contraction mechanisms (subsection 2.2.3). These effects are typically expressed as a set of first-order differential equations

$$\begin{aligned} \dot{\boldsymbol{a}}_m &= \boldsymbol{g}(\boldsymbol{u}_m, \boldsymbol{a}_m; \tau_a, \tau_d) \\ \dot{\boldsymbol{l}}_m &= \boldsymbol{h}(\boldsymbol{l}_m; \boldsymbol{\theta}_0) \end{aligned} \tag{3.5}$$

where $\boldsymbol{a}_m$ represents the muscle activation, which is a function of the excitations $\boldsymbol{u}_m$, τ_a, τ_d the activation and deactivation time constants and $\boldsymbol{l}_m$, $\dot{\boldsymbol{l}}_m$ the muscle length and lengthening speed. Vector $\boldsymbol{\theta}_0$ stands for a multitude of parameters, such as pennation angle, maximum velocity and muscle routing. The muscle force $\boldsymbol{f}_m$ depends on these quantities

$$\boldsymbol{f}_m = \boldsymbol{d}(\boldsymbol{a}_m, \boldsymbol{l}_m, \dot{\boldsymbol{l}}_m; \boldsymbol{\theta}_1) \tag{3.6}$$

and vector $\boldsymbol{\theta}_1$ denotes another set of parameters such as maximum isometric force.

If the motion is known $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ along with the externally applied forces, then Equation 3.4 can be solved for the unknown muscle force in an ID manner (subsection 2.3.3). On the other hand, if the muscle activation patterns are known, then Equation 3.4 can be solved in a FD manner for the unknown accelerations (subsec-

tion 2.3.4). In this case, the accelerations are integrated twice in order to acquire motion trajectories, for a given set of initial conditions.

3.2.3 Lachman Test

The Lachman test is a clinical test used to diagnose injury of the ACL. It is recognized as reliable, sensitive and usually superior to the anterior drawer test (van Eck et al., 2013). The patient should be relaxed for this test, especially the tested extremity. The examiner positions the tested leg into about 20° of flexion, by placing the examiner's knee under the patient's thigh. One hand is used to stabilize the distal femur near the joint line on the anterior side, while palpating the joint line. Then the thumb of the other hand is placed on the anterior side of the tibia and the fingers grasp the posterior side of the tibia near the joint line. Finally, quick posterior-to-anterior directed forces are applied through the tibia. There should be a firm end-feel. A positive test is excessive movement or the lack of a firm end-feel. An alternate method involves holding the femur and tibia without the examiner's knee under the patient's thigh. During the procedure, it is important that the correct joint angle is used for this test, since a position closer to full extension, naturally has less anterior translation of the tibia and can result in a false endpoint (Benjaminse et al., 2006).

The Lachman test is imperative for the prognosis, follow-up and scientific comparison of ACL surgery results. The most commonly used tools, that can quantify the Lachman test results, by measuring the respective displacements are the KT-1000, KT-2000 (MEDmetric, San Diego, Calif., USA), and the Striker knee laxity testers. An alternative and less expensive device, the Rolimeter knee tester (Aircast Europa, Neubeuern, Germany) has also shown exact and simple quantification of anterior knee joint instability (Muellner et al., 2001). The Rolimeter knee tester is a reliable device for quantifying knee joint laxity and is sensitive enough to identify anterior cruciate ligament deficiency (Hatcher et al., 2005). Consequently, the Rolimeter provides a cost effective and simple operating device for quantifying anterior knee joint instability. Recently, the diagnostic efficiency of all these devices has been confirmed in several investigations. Statistical evaluation showed no significant difference in the exactitude of measurement between the Rolimeter and the KT-1000 arthrometer which currently is the gold standard for such measurements (Balasch et al., 1999; Ganko et al., 2000).

3.2.4 Clinical Data

Experimental data from 30 young patients, age between 20–28 years old, average height and weight 1.78m and 78.3kg respectively with an excellent physical condition was used for model calibration. The patients were amateur football players, who suffered from ACL tear, where optimal surgery and rehabilitation was necessary. Different anthropometric measurements were carried, along with the Magnetic Resonance Imaging (MRI) recordings and the presurgical Lachman test

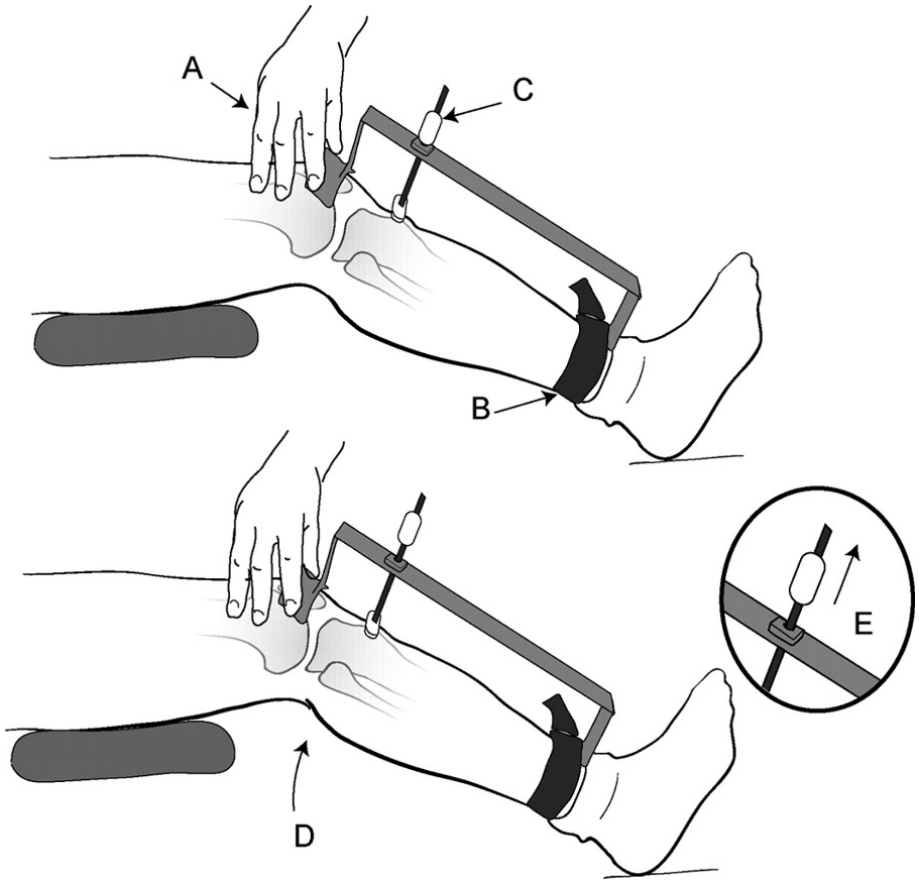


Figure 3.2: Rolimeter configuration during Lachman test.

in order to scale and calibrate the model accordingly.

Measurements of the tibiofemoral gap along with the length of the ligaments Table 3.1 have been obtained from MRI. The average, min and max values for the four ligaments collected from different patients are in agreement with data reported in (Cohen et al., 2009). Similarly, the stiffness coefficient of the ligaments was estimated based on the area of the ligament (Blankevoort et al., 1991; Liu et al., 2010). Measurements of the anterior displacement with respect to various magnitudes of the applied force were performed on patients with deficient knee, in the context of the presurgical Lachman test.

3.3 Results

3.3.1 Ligament Parameter Calibration

The ligament parameters were calibrated in order to produce a similar force - displacement curve as the one measured from the presurgical Lachman test, neglecting

Table 3.1: Average, min and max values for the five ligaments determined from MRI measurements from 30 young patients.

Ligament	Avg. Length (cm)	Min (cm)	Max (cm)
ACL	3.8	2.3	4.1
PCL	3.5	3.2	3.8
MCL	9.5	6	12.7
LCL	5.5	5	6

the contribution of ACL. The description of the Lachman test was used to configure the model posture. An external force is applied to the posterior aspect of the proximal tibia. The anterior displacement, muscle contribution and ligament stress and strains were obtained in a simulated manner, by utilizing FD in order to predict the response of the model to the loading conditions and ligament parameter changes. An optimization, that minimizes the distance between the measured and simulation-obtained anterior displacement of the Lachman test, was formulated. The error function is expressed as follows

$$\text{error}(\boldsymbol{\theta}_l) = \|(\boldsymbol{\delta}_m(\boldsymbol{f}_p; \boldsymbol{\theta}_l) - \boldsymbol{\delta}_e(\boldsymbol{f}_p))\|^2 \tag{3.7}$$

where $\boldsymbol{\delta}_m$ is the simulated model displacement, which is a function of the tibial posterior force $\boldsymbol{f}_p$ and the parameters of the ligaments $\boldsymbol{\theta}_l$ (Table 3.2), $\boldsymbol{\delta}_e$ the experimental displacement from the presurgical Lachman test. The ligament parameters that are under investigation are summarized in Table 3.2 along with their initial values (Xu et al., 2014). The clinical data from Table 3.1 was used to constrain the search space of the optimization. The optimization procedure was performed several times, restarting with different initial conditions to avoid sub-optimal solutions.

Figure 3.3 depicts the experimental anterior-posterior displacement as a function of the applied force obtained from the Lachman test and the model's response after the ligament parameter calibration, neglecting the contribution of aACL and pACL bundles. We observe that the calibrated model achieves a similar response under the same conditions of the presurgical Lachman test. The overshoot at the beginning of the simulated time series exhibits a transient stable response of the knee complex to the applied force and does not convey clinical information.

3.3.2 Clinical Decision Support System

Table 3.3 indicates the parameters that can be fine-tuned by the clinician or user of this framework in order to investigate their effect on the knee. Indicatively, the parameters under investigation are the number of bundles used for reconstruction and the insertion coordinates of the aACL on the tibia. Ligament strain is estimated for various potential values of the aforementioned parameters. Some of the

Table 3.2: Initial values for the ligament parameters; l_r reference length, ϵ_r reference strain and k stiffness (a: anterior, p: posterior, i: inferior and D: deep).

Ligament	l_r (cm)	k (kN)	ϵ_r
aACL	3.23	1.5	0.02
pACL	2.47	1.6	0.01
aPCL	2.58	2.6	-0.23
pPCL	2.52	1.9	0.02
aMCL	7.22	2.5	0.02
iMCL	7.31	3.0	0.04
pMCL	8.8	2.5	0.02
aDMCL	3.63	2.0	-0.08
pDMCL	3.72	4.5	0.03
LCL	5.59	2.0	0.02

parameters presented in Table 3.3 can be presurgically measured and used as an input to the model. The stiffness and the reference length of the ligaments can be approximated based on the ligaments' geometry obtained from the MRI (Bloemker et al., 2012), while the reference strain is difficult to measure and it is thus derived from the literature (Blankevoort et al., 1991; Blankevoort and Huiskes, 1996; Yang et al., 2010). Furthermore, the model can be further personalized (Valente et al., 2014), accounting for the skeletal segment geometry, mass parameters of the body segments (mass, inertia and center of mass), knee joint axis of rotation and muscular parameters (muscle length and maximum isometric, etc.), if such measurements are available.

Table 3.3: Potential variables that can be modified in order to investigate their effect on the biomechanical behavior of the knee.

Parameter	Variable
Ligament Attachment Positions	Tibia, Femur
Ligament Parameters	Stiffness, Reference Parameters
Model Customization	Weight, Height, Definition of the Knee Joint
Muscle Parameters	Activation, Contraction Dynamics
Actions	Simulated Activities

Initially, we compare two different knee models where the ACL is reconstructed using one or two bundles respectively. Simulation of the Lachman test for both models was utilized in order to estimate the response of the models (Figure 3.4). Using a single anterior ACL bundle is enough to constrain the anterior displacement of 2.7mm for 100N and posterior bundle doesn't contribute much the anterior displacement which was well expected. A variation of the initial rest-

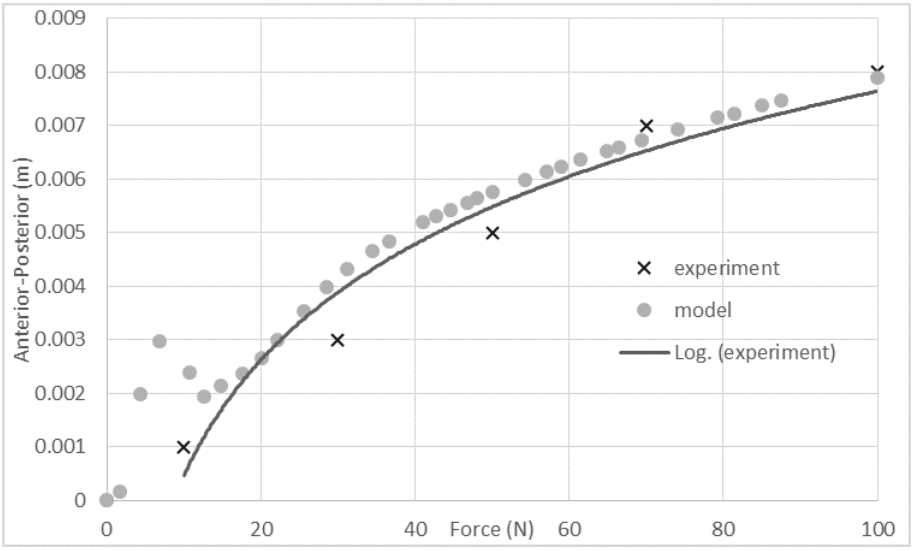


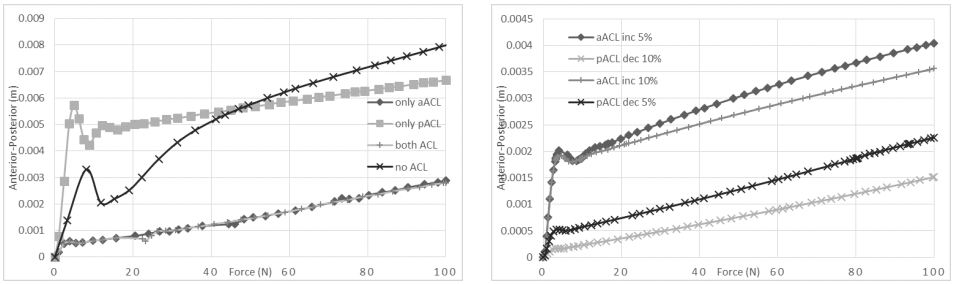
Figure 3.3: Comparison between experimentally measured and model response to the Lachman test after calibration of the ligament parameters. The diagram shows the anterior-posterior displacement with respect to the force applied to the posterior aspect of the proximal tibia.

ing length is made on both bundles, showing that the response of the model is very sensitive to changes of the ligament resting length that is also in agreement with (Bloemker et al., 2012). The resting length parameter of the bundles can be adjusted and an optimal solution can be chosen for each individual subject.

As illustrated in Figure 3.5 the passive forces of LCL and aDMCL ligaments during the simulated Lachman test are the main restrictors in the absence of ACL bundles (Shelburne et al., 2004). In both diagrams, we observe the strain with or without ACL bundles, where the peak passive force for LCL increases 11% and for aDMCL 67% during the absence of the ACL.

Extraction of comparative results in cases involving variables that cannot be explicitly observed, like for example the ligament strains in our case, is always a challenge. In order to produce comparable results a reference model for the simulations of knee flexion-extension is utilized. The reference model represents an intact knee with prescribed anterior-posterior and inferior-superior displacement during flexion which is based on the previous work of (Yamaguchi and Zajac, 1989; Delp et al., 1990; Anderson and Pandy, 1999) and is widely used in the biomechanics community. The location where the aACL is attached to the tibia is defined on a two-dimensional surface, forming a regular octagon with 10mm apothem. For every one of the eight possible positions a forward simulation was performed for a motion of knee flexion from 80° to full extension at 0°.

A potential decision criterion could be defined as the distance between the reference strains (intact knee model) and the strains produced by the simulation for



(a) Effect of bundle number. **(b)** Effect of ligament rest length.

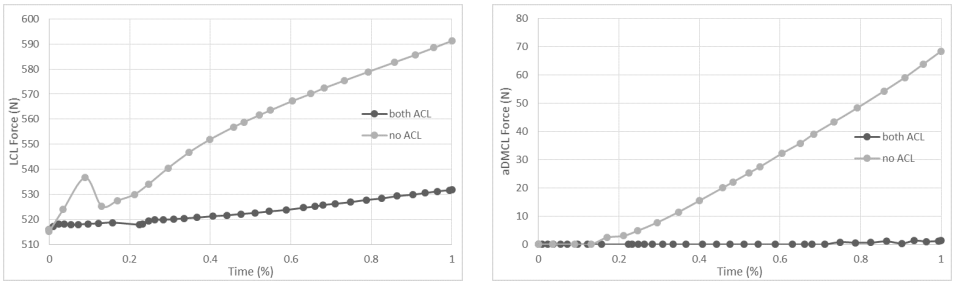
Figure 3.4: The diagrams show the anterior-posterior displacement with respect to the force applied to the posterior aspect of the proximal tibia during the simulated Lachman test for different surgical choices. On the left diagram the response of the model is investigated based on the insertion of single or double bundles for the ACL ligament. On the right diagram an investigation is made by adjusting the resting length of the two bundles, by increasing or decreasing the initial resting length.

every ligament with respect to the different attachment positions of the aACL. In Figure 3.6, we compare the simulated strain to the reference strain of each ligament for the different attachment positions. In Table 3.4, a summary of the distance error per position and per ligament is provided and we marked the smallest errors with bold. The distance error is measured by first aligning the reference strain and the simulated strain for knee flexion-extension angle. We can conclude that insertion position 8 produces the best overall matched strains with respect to the reference strains.

Table 3.4: The error between simulated and reference strains for every ligament and for every insertion position of the aACL on the tibia (the minimum errors are marked with bold). The error is defined as the accumulated absolute value between reference and simulated curves after alignment of the time series with respect to knee flexion-extension angle. Minimum values show that for the concrete insertion position the ligament strain curve is the closest to the reference strain curve of an intact knee during flexion.

Position	aACL	pACL	aPCL	pPCL	aMCL	iMCL	pMCL	aDMCL	pDMCL	LCL	Total
1	0.638	0.795	0.831	0.766	**0.252**	**0.254**	**0.217**	0.438	0.509	**0.320**	5.019
2	0.390	3.988	1.968	0.697	0.942	0.841	0.773	2.114	2.279	0.955	14.948
3	1.855	5.409	2.151	1.379	1.344	1.207	1.094	2.983	3.222	1.422	22.068
4	1.752	5.390	2.159	1.358	1.336	1.199	1.088	2.968	3.205	1.412	21.867
5	**0.322**	3.786	1.848	**0.689**	0.908	0.813	0.745	2.018	2.175	0.926	14.230
6	0.724	1.653	0.878	1.066	0.680	0.649	0.559	1.232	1.334	0.847	9.620
7	0.768	1.776	0.912	1.070	0.718	0.684	0.590	1.311	1.414	0.889	10.131
8	0.769	**0.637**	**0.718**	0.764	0.256	0.260	0.218	**0.415**	**0.489**	0.342	**4.870**

ID simulation is performed on gait and jump activities. The kinematics and



(a) LCL strain. **(b)** aDMCL strain.

Figure 3.5: Simulation of the strain force for LCL (left) and aDMCL (right) during the simulated Lachman test which are the main restrictors in the absence of ACL bundles.

ground reaction forces for the two movements were obtained from (Delp et al., 2007) and (Kar and Quesada, 2013), respectively. The parameter under investigation is the strain of the aACL ligament for the eight different insertion positions. In Figure 3.7, we present the collective results and compare the ligament strain for each insertion position with the reference strain. We observe that for some attachment position the strain of the ACL exhibits a compressive behavior, which may lead to ineffective restriction of anterior displacements. While this example, demonstrates that ID methods are useful for analyzing functional performance, it also highlights that they cannot be used to predict the result of the ligament's forces on the alterations in the knee kinematics.

3.4 Discussion-Conclusions

A computational model of the human knee was developed and subsequently used for simulation of custom activities. The proposed scheme demonstrates how VPH simulations can be used in a presurgical step for optimal planning of several parameters related to the surgical procedure. The effect of the selected parameters on the motion behavior of the knee can be evaluated in a FD manner, leading to a powerful clinical DSS. We conclude that muscles play an important role in the stability of the knee, thus neglecting their contribution can potentially lead to misleading results.

There is of course space for further improvement. Model parameterization from in vivo measurements is important in subject-specific healthcare, where geometric representation of the knee can be reconstructed from MRI (Valente et al., 2014). The ligament model can be further improved by taking into account wrapping surfaces or by modeling realistically the insertion and origin sites spanning the entire area and not just a single point. A more accurate and comprehensive model of the knee contact can be developed to account for the contribution of the menisci,

which have a significant effect of distributing the tibiofemoral contact forces. Even if, by definition, the proposed scheme aims to provide relative evaluations of different rehabilitation treatment strategies, an extensive clinical validation of the model and the simulation results are still necessary. Sensitivity and uncertainty analysis (Myers et al., 2015; Hatcher et al., 2005) can also be performed jointly with remodeling to evaluate the overall robustness of the current model.

Computational musculoskeletal models are a valuable tool for predicting the outcome of a surgical decision, demonstrating that such analyses may lead to improved clinical decision-making. On the other hand, there are a couple of aspects that need to be thoroughly examined in order to enable proper in silico medicine. ID methods are useful for analyzing settings based on the experimentally measured kinematics and kinetics, however, they cannot predict alterations in the subject kinematics resulting from a surgical intervention. Predictive simulation is possible only in a FD setting, yet the muscle excitations that generate a coordinated movement are not known a priori. The ultimate goal is to use musculoskeletal modeling and simulation to predict the result of a surgical intervention, without necessarily relying on experimental measurements of kinematics, kinetics and Electromyography (EMG). A possible solution, may lie in the combination of ID and FD methods. This is further explored in chapter 4, where the FD method is used for simulating the model response, with the input provided by an ID model-based controller representing the underlying system.

The estimation of muscle forces is essential for the analysis of the knee complex mechanics. For example, one may be interested in evaluating alterations in the tibiofemoral joint loading as a result of some surgical intervention. Clearly, the assessment of this quantity depends on the obtained combination of muscle forces, which cannot be measured directly. As there are multiple muscles spanning the knee joint, there are infinite combinations of muscle forces that can result in the same movement, due to muscle over-availability, hence predicted quantities that depend on these estimates cannot be calculated reliably. In chapter 5, a method for identifying the feasible solution space, that can help to properly interpret results obtained from redundant musculoskeletal systems, will be presented.

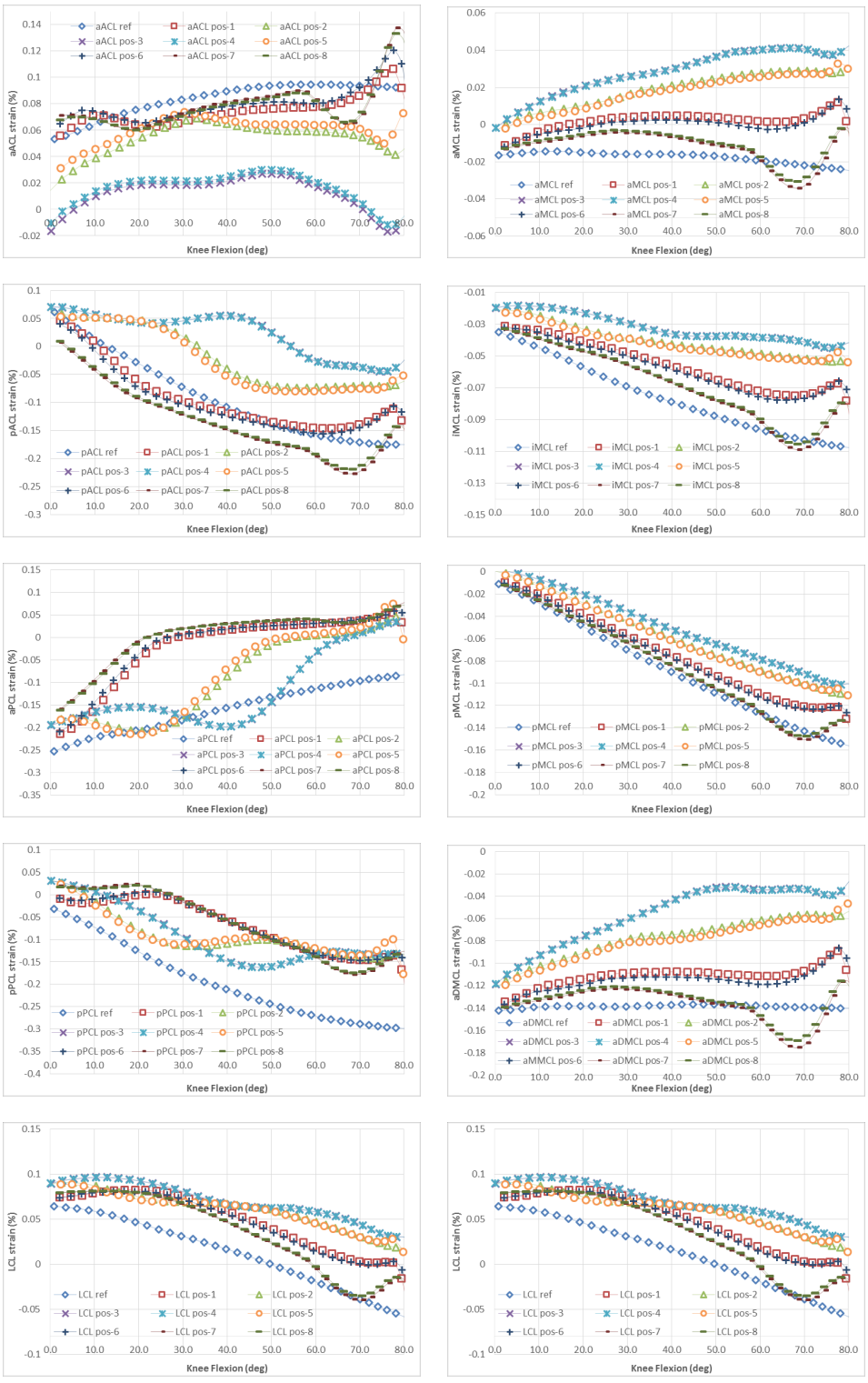


Figure 3.6: Comparison between simulated strains for all ligaments with respect to the reference strain of a healthy knee flexion motion (reference model). Investigation of 8 different attachment positions of the aACL on the tibia and their effect on the simulated strain curves.

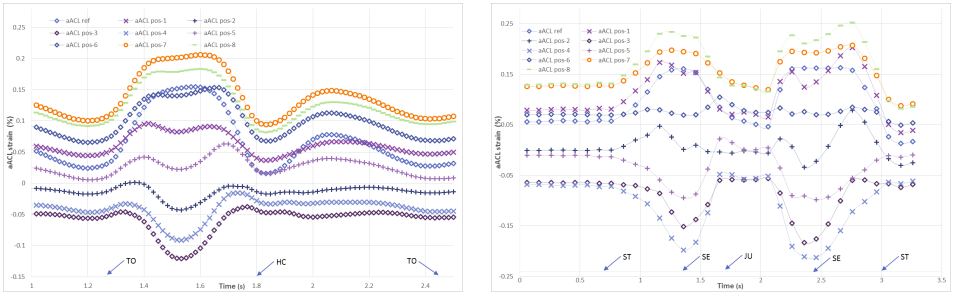


Figure 3.7: Simulation of gait (left) and jump action (right) for different insertion positions of the aACL on the tibia. For the gait simulation the annotations are TO: toe off and HC: heel contact. For the jump action the annotations are ST: stand, SE: seated and JU: jump postures.

Chapter 4

Simulation of Constrained Musculoskeletal Systems in Task Space

In this chapter, a framework for planning virtual simulations of constrained musculoskeletal systems without necessarily relying on experimentally measured kinematics and muscle activity will be presented. As was previously discussed (chapter 4), ID methods are useful for analyzing functional performance, however, they cannot be used to predict alterations in the kinematics due to a surgical intervention. On the other hand, while FD methods can predict these alterations, the muscle excitations that produce a coordinated movement are not known a priori. ID and FD methods are combined together (MD) in order to perform predictive simulation without relying on muscle excitations as an input. As the human body has many DoFs, it is easier to record a movement experimentally rather than design anatomically realistic behaviors virtually through motion planning. It turns out that, even if the motion of the human body is high-dimensional, we can still create realistic movements by working in a low-dimension subspace. This is achieved by expressing the EoMs into the space of a task using the theory of projection operators (subsection 2.1.1). We show that task-based approaches could be adopted in the design of simulation related to the study of constrained musculoskeletal systems.

The developed methods utilize task space projection, while the underlying model is extended to incorporate kinematic constrains. The change of representation requires different algorithms for solving the inverse and forward dynamics in the task space domain. We propose an extension to the Direct Marker Control (DMC) and an adaptation of the CMC algorithms for solving the IK and muscle redundancy problems, respectively. Experimental evaluation demonstrates that this framework is not only successful in dealing with the ID problem, but also provides an intuitive way of studying and designing simulations, facilitating assessment prior to any experimental data collection. The incorporation of constraints in the derivation unveils an important extension of this framework towards addressing

systems that use absolute coordinates and topologies that contain closed kinematic chains. Task space projection reveals a more intuitive encoding of the motion planning problem, allows for better correspondence between observed and estimated variables, provides the means to effectively study the role of kinematic redundancy and, most importantly, offers an abstract point of view and control, which can be advantageous towards further integration with high level models of the precommand level.

4.1 Introduction

The human coordination problem combines many levels of hierarchical organization that are performed in our body (DeWolf and Eliasmith, 2011; Wolpert, 1997). One way to model the complexity of controlling the musculoskeletal system (Valero-Cuevas et al., 2009) is to partition the problem into layers of abstraction and to encode the physiological factors that describe their functionality, providing the necessary details for the particular problem. Frequently, we find it difficult to express an abstract idea that encapsulates a problem and relate it to the underlying mathematical model. This has partly contributed to the fact that most of the algorithms are expressed in joint space, which may not always be the optimal representation.

As a matter of fact, many applications require a change of representation to solve a particular problem. Relevant examples include: the study of the mechanical capabilities of the human limbs (Valero-Cuevas, 2009a), where fundamental questions on the interaction of the nervous system with the physical world are addressed; the role of muscle synergies, their relation to the execution of a task (Kutch and Valero-Cuevas, 2012) and the influence they have in shaping the end point stiffness (Inouye and Valero-Cuevas, 2016); predictive simulation of closed-chain systems during execution of a crank-rotation task (Davoudabadi Farahani et al., 2016). In this context, researchers are interested in relating the meaningful, observable quantities (e.g., position, orientation, force, end point stiffness, etc.) exerted by a segment (typically end-effector) to the muscle recruitment distribution. To account for this, we seek to establish the equations that dictate the task characteristics and muscle excitation patterns.

The main goal of the developed methods is to provide a transparent link between the task and the corresponding muscle coordination patterns. The concept of a task has an abstract meaning and can be interpreted as a high level command that encapsulates the characteristics of a movement similar to the planning occurring in the brain (Kawato, 1999). The present study focuses mainly on motion primitive tasks (e.g., position and orientation), without necessarily restricting applications to those alone (e.g., force or impedance control (Hogan, 1984) could also be addressed). The operational task space formalization (Khatib, 1995) provides the mathematical tools to map the EoMs of the underlying system to the space of the task and control the movement in this domain.

The proposed algorithms and controllers are designed so as to handle constrained multibody systems. This is of significant importance since many biological models (e.g., anatomical joints (Seth et al., 2010)) or experimental setups (e.g., crank-rotation tasks (Davoudabadi Farahani et al., 2016)) are modeled through constraints. Consequently, the presented framework can address models that use absolute (Cartesian) coordinates (Dziewiecki et al., 2014) instead of generalized coordinates (subsection 4.3.4) or systems that contain closed kinematic chains (De Sapio and Park, 2010) (subsection 4.3.5). The effect of different joint topologies (e.g., closed kinematic chains) and the influence of constraint modeling on the required muscle forces remains a subject open to study. Results show that constraint modeling can alter the required generalized motion forces and consequently affect the muscle forces (subsection 4.3.1).

Although task space formalization has been proposed for the study of musculoskeletal systems (Khatib et al., 2009), there is a lack of tools suitable for solving the inverse and forward dynamics problems (Erdemir et al., 2007). Considering this, we present an implementation of a dynamically-based IK algorithm (Task Space Inverse Kinematics (TSIK) subsection 4.2.5) for solving the IK problem. We show that this algorithm is capable of tracking the recorded movement (subsection 4.3.2), while satisfying the imposed constraints due to its dynamic nature. Moreover, we present an algorithm (Task Space Computed Muscle Control (TSCMC) subsection 4.2.6) for performing muscle driven dynamics simulations that closely reproduce experimental measurements of kinematics and ground reaction forces, similar to the CMC algorithm (Thelen and Anderson, 2006). The latter is used to perform FD simulations by encoding the movement behavior from a set of abstract task goals, thus enabling model assessment prior to any experimental data collection. The performance of the two algorithms has been evaluated for a gait movement against the current available state-of-the-art schemes.

We employed and extended the open source tool `OpenSim` (Delp et al., 2007) for modeling and simulation of the human movement. `OpenSim` represents the open science movement, algorithm sharing and reproducibility of scientific results (Erdemir et al., 2016). Currently, `OpenSim` does not provide an integrated support for handling operational task space formulation and projection of the EoMs. The source code along with any related material for this study are publicly available (section A.1), providing simple examples so that the readers can reproduce, understand and reuse the presented methods.

4.1.1 Related Work

Operational task space (Khatib, 1995; Khatib et al., 2004) control has been successfully applied by the robotics community yielding novel control schemes, intuitive design, implementation and practical applications. Multibody theory is the basis for studying the kinematics and dynamics of the skeletal system. The underlying EoMs have been extended, so as to describe the human muscle complexity (Thelen, 2003; Millard et al., 2013). Consequently, this can be used to evaluate the cause

of an orchestrated muscle command to the observable movement. Simulation of human movement has been successfully translated in clinical treatment of cerebral palsy, lower extremity amputees and osteoarthritis (Myers et al., 2015). As well as, basic science related to the understanding of movement progression and control during dynamic tasks, contributing to the perception of important neurodegenerative diseases (Arnold et al., 2000; Pandy, 2001; Delp et al., 2007). Pioneer work in the combination of the above mentioned fields has been presented in (Khatib et al., 2009), stressing how these concepts can be applied to study the musculoskeletal complex in the task space. The results have meaningful biological interpretations and are not a product of systems engineering.

In clinical practice, there are two main approaches for computing the internal state of the musculoskeletal composite from the available measurements (Erdemir et al., 2007). During ID, the analysis starts from the effect (resultant movement) and propagates all the way up in the hierarchy to the cause (muscle excitation signals) (Buchanan et al., 2006). On the other hand, during FD a synchronous command triggers the muscles, whereas the resulting movement is estimated and observed. Different combinations of these schemes can be found in the literature (Shourijeh et al., 2016; Ong et al., 2016), building upon the problem's complexity, the available measurements and the assessment requirements. Commonly observable variables are the movement of the segments recorded by a MoCap system, the external forces that act on the system (e.g., ground reaction during walking) (Pizzolato et al., 2015) and the EMG recordings of muscle activity (Durandau et al., 2017; Dideriksen et al., 2011). These measurements are used to estimate the internal state of the model (joint space generalized forces, muscle forces, etc.) and further restrict the possible solutions of the redundant system, both in terms of kinematic and dynamic redundancy.

Many biological models require adequate constraint modeling, so that the underlying model approximates the realistic movement behavior accurately (Seth et al., 2010). The knee and shoulder complex are representative examples of such models (Mootanah et al., 2014; Chadwick et al., 2014; De Sapio and Park, 2010). The incorporation of constraints introduces many technical and scientific challenges when solving the forward and ID problem (Aghili, 2005; Righetti et al., 2011; Mistry and Righetti, 2011). Some of them are finding a solution existence of the system configuration under constraints and difficulties in solving the ODE (Eich, 1993). As an example, numerical drifts can still result in constraint violation (Baumgarte, 1972; Eich, 1993) and the introduction of constraints can result in DAE of index-3 that are hard to solve (Eich, 1993). Although constraints are an inseparable part of the model, their influence on the muscle recruitment problem is not thoroughly studied, even if any underlying simulation scheme should be able to handle these types of systems.

The proposed approach fits well into the wide spectrum of the current biomedical literature extending the task-based approach towards applications in musculoskeletal simulation. The developed algorithms solve the motion planning problem intuitively with direct translation in clinical practice. Extension towards con-

strained multibody systems facilitates the modeling of the broader type of problems found in many biological systems.

4.2 Methods

4.2.1 Equations of Motion of the Plant

In the derivation of the EoMs, we will address systems with holonomic constraints. The fundamental equations of differential variational principles for a constrained mechanical system lead to the following DAE of index-3 (Eich, 1993)

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_c = \boldsymbol{\tau} \tag{4.1}$$

$$\phi(\boldsymbol{q}) = \boldsymbol{0} \tag{4.2}$$

where $\boldsymbol{M} \in \Re^{n \times n}$ denotes the symmetric, positive definite joint space inertia mass matrix, n the number of model DoFs and $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \Re^n$ the joint space generalized coordinates and their derivatives with respect to time. The term $\boldsymbol{f} \in \Re^n$ models all internal and external applied forces (e.g., gravity, Coriolis, GRF, etc.), $\boldsymbol{\tau}_c \in \Re^n$ the generalized forces induced by constraints and $\boldsymbol{\tau} \in \Re^n$ the vector of applied generalized forces that actuate the model[1]. Equation 4.2 corresponds to a set of c constraint algebraic equations which can be differentiated twice with respect to time (the dot notation depicts a derivative with respect to time)

$$\boldsymbol{\Phi}\dot{\boldsymbol{q}} = \boldsymbol{0}, \ \boldsymbol{\Phi} = \frac{\partial \phi}{\partial \boldsymbol{q}} \tag{4.3}$$

and

$$\boldsymbol{\Phi}\ddot{\boldsymbol{q}} = -\dot{\boldsymbol{\Phi}}\dot{\boldsymbol{q}} = \boldsymbol{b} \tag{4.4}$$

where the term $\boldsymbol{\Phi} \in \Re^{c \times n}$ defines the constraint Jacobian.

At this point it is worth considering the effect of Equations 4.2, 4.3 and 4.4 on the permissible configuration solutions (Equation 4.1). The algebraic constraint equations (Equation 4.2) imply that any admissible configuration $\boldsymbol{q} \in \Re^n$ should lie on the constraint manifold (Aghili, 2005; De Sapio and Park, 2010). The first derivative of the constraints (Equation 4.3) suggests that $\dot{\boldsymbol{q}} \in \mathbb{N}(\boldsymbol{\Phi})$, following the definition of null space (Equation 2.4). As a consequence of the D'Alembert's principle, constraint forces do no work under any virtual displacement that satisfy them ($\boldsymbol{\tau}_c \delta\boldsymbol{q} = \boldsymbol{0}, \forall \delta\boldsymbol{q} \in \mathbb{N}(\boldsymbol{\Phi})$ which is equivalent to $\boldsymbol{\tau}_c \perp \delta\boldsymbol{q}, \forall \delta\boldsymbol{q} \in \mathbb{N}(\boldsymbol{\Phi})$). In turn, this implies that $\boldsymbol{\tau}_c \in \mathbb{N}(\boldsymbol{\Phi})^{\perp} = \mathbb{C}(\boldsymbol{\Phi}^T)$. Thus, the generalized constraint forces can be represented as a linear combination of columns of $\boldsymbol{\Phi}^T$

[1]Most of the quantities in the equations are a function of the generalized coordinates and their derivatives. This dependency will be omitted for simplicity.

$$\boldsymbol{\tau}_c = \boldsymbol{\Phi}^T \boldsymbol{\lambda} \tag{4.5}$$

where $\boldsymbol{\lambda} \in \Re^c$ stands for the vector of Lagrange multipliers. While $\dot{\boldsymbol{q}}_{\|} = \mathbf{0}$, where $\dot{\boldsymbol{q}}_{\|} = \{\dot{\boldsymbol{q}} : \dot{\boldsymbol{q}} \in \mathbb{N}(\boldsymbol{\Phi})^{\perp}\}$, the same does not always hold for the acceleration ($\ddot{\boldsymbol{q}}_{\|} \neq \mathbf{0}$) by inspecting the second derivative of the constraints (Equation 4.4).

Equation 4.1 can be solved in a FD manner for the unknowns $\ddot{\boldsymbol{q}}$, $\boldsymbol{\lambda}$, through the complement of Equations 4.2, 4.3 and 4.4. The constraint derivatives can reduce the DAE to index-1, making the system solvable by an ODE solver. However, numerical integration inevitably leads to drifts that eventually result in constraint violation ($\phi(\boldsymbol{q}(t)) \neq \mathbf{0}$). Coordinate projection (Eich, 1993) or Baumgarte stabilization (Baumgarte, 1972) can be introduced to ensure exponential elimination of the constraint error, given the appropriate initial conditions.

4.2.2 Inverse Dynamics Controller

Before presenting the Inverse Task Space Controller (ITSC) we will first elaborate on alternative implementations of the constrained ID model that can significantly impact the derivation of the required generalized forces. A consequence of incorporating constrains is that if the system is overconstrained, different sets of applied forces can generate the same movement behavior (Righetti et al., 2011). In the following, we will present two types of controllers, that project the underlying EoMs into the constraint manifold by two different projection operators. The derivation begins with the following model

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{f} + \boldsymbol{\Phi}^T \boldsymbol{\lambda} = \boldsymbol{\tau} \tag{4.6}$$

$$\boldsymbol{\Phi}\ddot{\boldsymbol{q}} = \boldsymbol{b}. \tag{4.7}$$

Both controllers aim to decouple the constraint and applied forces. The first approach (De Sapio and Park, 2010) eliminates the constraints by solving Equations 4.6 and 4.7 for $\boldsymbol{\lambda}$

$$\begin{aligned} \underbrace{\boldsymbol{\Phi}\boldsymbol{M}^{-1}\boldsymbol{M}\ddot{\boldsymbol{q}}}_{\boldsymbol{b}} + \boldsymbol{\Phi}\boldsymbol{M}^{-1}\boldsymbol{f} + \underbrace{\boldsymbol{\Phi}\boldsymbol{M}^{-1}\boldsymbol{\Phi}^T}_{\boldsymbol{\Lambda}_c^{-1}}\boldsymbol{\lambda} &= \boldsymbol{\Phi}\boldsymbol{M}^{-1}\boldsymbol{\tau} \\ \boldsymbol{\lambda} = \underbrace{\boldsymbol{\Lambda}_c\boldsymbol{\Phi}\boldsymbol{M}^{-1}}_{\bar{\boldsymbol{\Phi}}^T}(\boldsymbol{\tau} - \boldsymbol{f}) - \boldsymbol{\Lambda}_c\boldsymbol{b} & \end{aligned} \tag{4.8}$$

where $\boldsymbol{\Lambda}_c \in \Re^{c\times c}$ represents the constraint compliant inertia mass matrix and $\bar{\boldsymbol{\Phi}} \in \Re^{n\times c}$ the generalized inertia weighted inverse of the constraint Jacobian ($\boldsymbol{\Phi}$). We can substitute the solution obtained from Equation 4.8 into Equation 4.6 (Model-A):

$$\begin{aligned} \boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{f}_{\perp} + \boldsymbol{b}_c &= \boldsymbol{\tau}_{\perp} \\ \boldsymbol{f}_{\perp} = \boldsymbol{N}_{\Phi}^T \boldsymbol{f},\ \boldsymbol{\tau}_{\perp} &= \boldsymbol{N}_{\Phi}^T \boldsymbol{\tau} \\ \boldsymbol{N}_{\Phi}^T &= \boldsymbol{I} - \boldsymbol{\Phi}^T \bar{\boldsymbol{\Phi}}^T \\ \boldsymbol{b}_c &= -\boldsymbol{\Phi}^T \boldsymbol{\Lambda}_c \boldsymbol{b} \end{aligned} \tag{4.9}$$

where $\boldsymbol{N}_{\Phi} \in \Re^{n \times n}$ represents the constraint null space matrix ($\mathbb{C}(\boldsymbol{N}_{\Phi}) = \mathbb{N}(\boldsymbol{\Phi})$) and $\boldsymbol{b}_c \in \Re^n$ the constraint acceleration bias term (Equation 4.7). Here, the symbol $\perp$ denotes that the projected quantities lie in the null space of the constraints. This signifies that only the null space component of the generalized forces contributes to the motion of a constrained mechanical system, while the residual forces ($(\boldsymbol{I} - \boldsymbol{N}_{\Phi}^T)(\boldsymbol{\tau} - \boldsymbol{f})$) are compensated by the constraints. Moreover, $\boldsymbol{N}_{\Phi}$ is a projection operator ($\boldsymbol{N}_{\Phi}^2 = \boldsymbol{N}_{\Phi}$), which is not necessarily orthogonal ($\boldsymbol{N}_{\Phi} \neq \boldsymbol{N}_{\Phi}^T$) and $\boldsymbol{\Phi}\boldsymbol{N}_{\Phi} = \boldsymbol{0}$ holds by virtue of the relation between the null and range space (Equation 2.6).

In contrast to Model-A, Aghili (Aghili, 2005) proposed a more general derivation of the constrained ID model. The EoMs can be projected into the null space of the constraints, eliminating the constraint forces through an orthogonal projection operator $\boldsymbol{N}_{\Phi}^{'}$ ($\boldsymbol{N}_{\Phi}^{'2} = \boldsymbol{N}_{\Phi}^{'}$, $\boldsymbol{N}_{\Phi}^{'} = \boldsymbol{N}_{\Phi}^{'T}$), where $\Phi\boldsymbol{N}_{\Phi}^{'} = \boldsymbol{0}$ and $\boldsymbol{N}_{\Phi}^{'} = \boldsymbol{I} - \boldsymbol{\Phi}^{+}\boldsymbol{\Phi}$ (+ indicates the Moore-Penrose pseudoinverse). In comparison with the projection operator $\boldsymbol{N}_{\Phi}$, that depends on the inertia mass matrix, $\boldsymbol{N}_{\Phi}^{'}$ is easier to compute in practice (Righetti et al., 2011), being purely kinematic dependent. Equations 4.6 and 4.7 can be expressed as follows

$$\boldsymbol{N}_{\Phi}^{'}\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{N}_{\Phi}^{'}\boldsymbol{f} = \boldsymbol{N}_{\Phi}^{'}\boldsymbol{\tau} \tag{4.10}$$

$$\ddot{\boldsymbol{q}}_{\parallel} = (\boldsymbol{I} - \boldsymbol{N}_{\Phi}^{'})\ddot{\boldsymbol{q}} = \boldsymbol{\Phi}^{+}\boldsymbol{b}. \tag{4.11}$$

Typically, $\boldsymbol{N}_{\Phi}^{'}$ is rank deficient, thus $\boldsymbol{N}_{\Phi}^{'}\boldsymbol{M}$ is not invertible. However, since the system is constrained and Equations 4.10 and 4.11 lie in mutually orthogonal spaces (they cannot cancel each other out), they can be added (Model-B):

$$\begin{aligned} \boldsymbol{M}^{'}\ddot{\boldsymbol{q}} + \boldsymbol{f}_{\perp} + \boldsymbol{b}_c^{'} &= \boldsymbol{\tau}_{\perp} \\ \boldsymbol{M}^{'} &= \boldsymbol{M} + \boldsymbol{N}_{\Phi}^{'}\boldsymbol{M} - (\boldsymbol{N}_{\Phi}^{'}\boldsymbol{M})^T \\ \boldsymbol{f}_{\perp} = \boldsymbol{N}_{\Phi}^{'}\boldsymbol{f},\ \boldsymbol{\tau}_{\perp} &= \boldsymbol{N}_{\Phi}^{'}\boldsymbol{\tau} \\ \boldsymbol{b}_c^{'} &= -\boldsymbol{M}\boldsymbol{\Phi}^{+}\boldsymbol{b} \end{aligned} \tag{4.12}$$

where the choice of $\boldsymbol{M}^{'}$ is not unique, as there are many ways that Equations 4.10 and 4.11 can be combined. In the derivation of Equation 4.12 they chose to premultiply Equation 4.11 by $\boldsymbol{M}$ before adding to Equation 4.10. It can be shown that $\boldsymbol{M}^{'}$ is invertible as long as $\boldsymbol{M}$ is invertible (Aghili, 2005). It is evident that the choice of $\boldsymbol{b}_c^{'}$ depends on the choice of $\boldsymbol{M}^{'}$.

The constraint generalized forces ($\boldsymbol{\tau}_c$) can be derived by first projecting Equation 4.6 with ($\boldsymbol{I} - \boldsymbol{N}_\Phi^{'}$) and combine the result from Equation 4.12

$$\begin{aligned}(\boldsymbol{I} - \boldsymbol{N}_\Phi^{'})\boldsymbol{M}\ddot{\boldsymbol{q}} + (\boldsymbol{I} - \boldsymbol{N}_\Phi^{'})(\boldsymbol{f} + \boldsymbol{\tau}_c) &= (\boldsymbol{I} - \boldsymbol{N}_\Phi^{'})\boldsymbol{\tau} \\ (\boldsymbol{I} - \boldsymbol{N}_\Phi^{'})\boldsymbol{M}\boldsymbol{M}^{'-1}(\boldsymbol{\tau}_\perp - \boldsymbol{f}_\perp - \boldsymbol{b}_c^{'}) + \boldsymbol{f}_\parallel + \boldsymbol{\tau}_c &= \boldsymbol{\tau}_\parallel \\ \boldsymbol{\tau}_c &= \boldsymbol{\tau}_\parallel - \boldsymbol{f}_\parallel - \boldsymbol{\Xi}(\boldsymbol{\tau}_\perp - \boldsymbol{f}_\perp - \boldsymbol{b}_c^{'})\end{aligned} \tag{4.13}$$

where $\boldsymbol{\Xi} = (\boldsymbol{I} - \boldsymbol{N}_\Phi^{'})\boldsymbol{M}\boldsymbol{M}^{'-1}$ and $\boldsymbol{\tau}_\parallel$, $\boldsymbol{f}_\parallel$ represent the contribution of these forces in the constraint subspace. Equation 4.13 implies that the constraint generalized forces ($\boldsymbol{\tau}_c$) can be obtained uniquely, whereas the derivation of the Lagrange multipliers ($\boldsymbol{\lambda}$) is not unique, since $\boldsymbol{\Phi}$ can be rank deficient.

4.2.3 Inverse Task Space Controller for a Single Task

In this section we will describe the process of projecting the EoMs derived previously into the space of the task. In this derivation we will not consider any prior choice of $\boldsymbol{M}$, $\boldsymbol{b}_c$ and $\boldsymbol{N}_\Phi$.

Each task is associated with a task Jacobian matrix $\boldsymbol{J}_t \in \Re^{d\times n}$, where d is the dimension of the task (e.g., three for position task, three for orientation task and six for spatial task, which is the combination of the first two)[2]. For a fixed joint space configuration, $\boldsymbol{J}_t$ is a linear mapping from joint velocities $\dot{\boldsymbol{q}}$ to task velocities $\dot{\boldsymbol{x}}_t$

$$\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t\dot{\boldsymbol{q}} \tag{4.14}$$

$$\ddot{\boldsymbol{x}}_t = \dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}} + \boldsymbol{J}_t\ddot{\boldsymbol{q}}. \tag{4.15}$$

In turn, the transpose of a task Jacobian maps from task space forces $\boldsymbol{f}_t \in \Re^d$ to joint space generalized forces $\boldsymbol{\tau} \in \Re^n$

$$\boldsymbol{\tau} = \boldsymbol{J}_t^T\boldsymbol{f}_t. \tag{4.16}$$

For a given configuration, there is an infinite number of elementary displacements that could take place without altering the configuration of the effector (Khatib, 1995), because in general $d < n$. Those displacements correspond to motion in the null space associated with the generalized inverse of the Jacobian matrix. With the addition of the null space forces, the relationship between task forces and manipulator joint space generalized forces takes the following general form

$$\boldsymbol{\tau} = \boldsymbol{J}_t^T\boldsymbol{f}_t + \boldsymbol{N}_{J_t}^T\boldsymbol{\tau}_0,\ \boldsymbol{N}_{J_t}^T = (\boldsymbol{I} - \boldsymbol{J}_t^T\bar{\boldsymbol{J}}_t^T) \tag{4.17}$$

where $\boldsymbol{N}_{J_t} \in \Re^{n\times n}$ is the null space of $\boldsymbol{J}_t$ ($\boldsymbol{J}_t\boldsymbol{N}_{J_t} = \boldsymbol{0}$), $\boldsymbol{\tau}_0 \in \Re^n$ a vector of arbitrarily selected generalized forces, which will be projected in the null space direction of $\boldsymbol{J}_t$ and $\bar{\boldsymbol{J}}_t \in \Re^{n\times d}$ the generalized inverse of $\boldsymbol{J}_t$.

[2]Note that the dimension of the task can be $1 \leq d \leq 6$, since in general we can choose which DoFs to utilize. A task that does not have DoFs ($d = 0$) is a constraint.

Khatib (Khatib, 1995) showed that the generalized inverse for a single task and an unconstrained system exists and is uniquely defined as follows

$$\bar{\boldsymbol{J}}_t^T = \boldsymbol{\Lambda}_t \boldsymbol{J}_t \boldsymbol{M}^{-1} \tag{4.18}$$

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

where $\boldsymbol{\Lambda}_t \in \Re^{d \times d}$ represents the task compliant inertia mass matrix. The dynamically consistent generalized inverse ($\bar{\boldsymbol{J}}_t$) is the only generalized inverse that results in zero task acceleration for any $\boldsymbol{\tau}_0$. In the presence of a single task the control law has the form of Equation 4.17. We can apply this control law to the constrained equations of the previous section and solve for the task forces

$$\begin{gathered} \underbrace{\boldsymbol{J}_t \boldsymbol{M}^{-1} \boldsymbol{M} \ddot{\boldsymbol{q}}}_{\ddot{\boldsymbol{x}}_t - \dot{\boldsymbol{J}}_t \dot{\boldsymbol{q}}} + \boldsymbol{J}_t \boldsymbol{M}^{-1} (\boldsymbol{f}_\perp + \boldsymbol{b}_c) = \underbrace{\boldsymbol{J}_t \boldsymbol{M}^{-1} \boldsymbol{N}_\Phi^T \boldsymbol{J}_t^T}_{\boldsymbol{\Lambda}_{t|c}^{-1}} \boldsymbol{f}_t \\ \boldsymbol{\Lambda}_{t|c} (\ddot{\boldsymbol{x}}_t + \boldsymbol{b}_t) + \bar{\boldsymbol{J}}_{t|c}^T (\boldsymbol{f}_\perp + \boldsymbol{b}_c) = \boldsymbol{f}_t \\ \boldsymbol{J}_t \boldsymbol{M}^{-1} \boldsymbol{N}_\Phi^T (\boldsymbol{I} - \boldsymbol{J}_t^T \bar{\boldsymbol{J}}_t^T) \boldsymbol{\tau}_0 \stackrel{!}{=} \boldsymbol{0} \end{gathered} \tag{4.20}$$

where $\boldsymbol{\Lambda}_{t|c} \in \Re^{d \times d}$ is the constrained task compliance inertia mass matrix, $\boldsymbol{b}_t = -\dot{\boldsymbol{J}}_t \dot{\boldsymbol{q}}$ the task acceleration bias term and $\bar{J}_{t|c}^T = \Lambda_{t|c} J_t M^{-1}$ the constrained task Jacobian generalized inverse. The symbol $t|c$ denotes that the task t quantity is "filtered out" (prioritized) by the null space of the constraints c. Note that we required that the null space forces $\boldsymbol{\tau}_0$ should not interfere with the task acceleration. To achieve this for any $\boldsymbol{\tau}_0 \neq \boldsymbol{0}$ we must solve equation $\boldsymbol{J}_t \boldsymbol{M}^{-1} \boldsymbol{N}_\Phi^T (\boldsymbol{I} - \boldsymbol{J}_t^T \bar{\boldsymbol{J}}_t^T) = \boldsymbol{0}$ for $\bar{\boldsymbol{J}}_t^T$

$$\bar{\boldsymbol{J}}_t^T = \bar{\boldsymbol{J}}_{t|c}^{'T} = \boldsymbol{\Lambda}_{t|c} \boldsymbol{J}_t \boldsymbol{M}^{-1} \boldsymbol{N}_\Phi^T \neq \bar{\boldsymbol{J}}_{t|c}^T \tag{4.21}$$

where, in turn the total residual null space of the task subject to the constraints is given by

$$\boldsymbol{N}_{t|c}^T = \boldsymbol{N}_\Phi^T (\boldsymbol{I} - \boldsymbol{J}_t^T \bar{\boldsymbol{J}}_{t|c}^{'T}) = (\boldsymbol{I} - \boldsymbol{J}_{t|c}^T \bar{\boldsymbol{J}}_{t|c}^T) \boldsymbol{N}_\Phi^T \tag{4.22}$$

where $\boldsymbol{J}_{t|c}^T = \boldsymbol{N}_\Phi^T \boldsymbol{J}_t^T$. Finally, the computed generalized forces for a given desired command $\ddot{\boldsymbol{x}}_t$ are

$$\boldsymbol{\tau}_\perp = \boldsymbol{J}_{t|c}^T \boldsymbol{f}_t + \boldsymbol{N}_{t|c}^T \boldsymbol{\tau}_0. \tag{4.23}$$

4.2.4 Inverse Task Space Controller for Multiple Tasks

An important aspect of working in the task space is that in the presence of multiple tasks (Sentis, 2007), we can further assign a priority level for each task. Consequently, lower level tasks will not produce acceleration that will interfere with the performance of higher priority tasks. This is achieved by projecting the individual task into the null space of the higher priority tasks through the null space matrix.

This is very important for two reasons: 1) helps to assign an importance factor in the execution of tasks and 2) helps to further partition and decompose the control into individual tasks that do not interfere with higher priority ones. The following control scheme is adopted

$$\boldsymbol{\tau} = \underbrace{\sum_{i=1}^{g} \boldsymbol{J}_{i|i-1*}^{T} \boldsymbol{f}_i}_{\text{task forces}} + \underbrace{\boldsymbol{N}_{g*}^{T} \boldsymbol{\tau}_0}_{\text{null space forces}} \quad . \tag{4.24}$$

There are g tasks in total and each task force (f_i) is projected onto the aggregate null space ($\boldsymbol{J}_{i|i-1*} = \boldsymbol{J}_i \boldsymbol{N}_{i-1*}$) of the higher priority tasks with respect to i ($i*$ denotes an aggregation). For the purpose of controlling the musculoskeletal system we partition the applied generalized forces ($\boldsymbol{\tau}$), into those that are responsible for driving the task goals and those that operate in the aggregate null space of all tasks ($\boldsymbol{N}_{g*}^{T}$). This notation is very similar to the posture control separation proposed in (Sentis, 2007), but we can choose to utilize the kinematic redundancy in a different manner. More specifically, the term $\boldsymbol{\tau}_0$ can be appropriately selected to compensate for any residual force in the null space (e.g. $\boldsymbol{\tau}_0 = \boldsymbol{f}_\perp + \boldsymbol{b}_c$). Alternatively, this redundancy can be exploited to solve the case of underactuated systems (Mistry and Righetti, 2011; Aghili, 2005). We can substitute Equation 4.24 into the ID model and solve for any arbitrary task force $\boldsymbol{f}_k,\ 1 \leq k \leq g$

$$\begin{aligned}
\boldsymbol{J}_k \boldsymbol{M}^{-1} \boldsymbol{M} \ddot{\boldsymbol{q}} + \boldsymbol{J}_k \boldsymbol{M}^{-1} (\boldsymbol{f}_\perp + \boldsymbol{b}_c) &= \underbrace{\boldsymbol{J}_k \boldsymbol{M}^{-1} \boldsymbol{J}_{k|k-1*}^{T}}_{\boldsymbol{\Lambda}_{k|k-1*}^{-1}} \boldsymbol{f}_k + \boldsymbol{J}_k \boldsymbol{M}^{-1} \sum_{i \neq k}^{g} \boldsymbol{J}_{i|i-1*}^{T} \boldsymbol{f}_i + \\
&\cancelto{\mathbf{0}}{\boldsymbol{J}_k \boldsymbol{M}^{-1} \boldsymbol{N}_{g*}^{T} \boldsymbol{\tau}_0} \\
\boldsymbol{\Lambda}_{k|k-1*} (\ddot{\boldsymbol{x}}_k + \boldsymbol{b}_k) + \boldsymbol{\Lambda}_{k|k-1*} \boldsymbol{J}_k \boldsymbol{M}^{-1} (\boldsymbol{f}_\perp + \boldsymbol{b}_c) &= \boldsymbol{f}_k + \boldsymbol{\Lambda}_{k|k-1*} \boldsymbol{J}_k \boldsymbol{M}^{-1} \sum_{i=1}^{k-1} \boldsymbol{J}_{i|i-1*}^{T} \boldsymbol{f}_i \\
\boldsymbol{f}_k &= \boldsymbol{\Lambda}_{k|k-1*} (\ddot{\boldsymbol{x}}_k + \boldsymbol{b}_k) + \bar{\boldsymbol{J}}_{k|k-1*}^{T} (\boldsymbol{f}_\perp + \boldsymbol{b}_c - \boldsymbol{\tau}_{k-1*})
\end{aligned} \tag{4.25}$$

where $\boldsymbol{\tau}_{k-1*}$ is the contribution of the higher priority tasks. Note that the summation term, which represents the force contribution of each task, stops at index $k-1$ (not g) because all lower priority tasks ($i > k$) will yield zero interference, complying with our requirement. The same applies to the null space term $\boldsymbol{J}_k \boldsymbol{M}^{-1} \boldsymbol{N}_{g*}^{T} \boldsymbol{\tau}_0 = \mathbf{0}$. It can be shown that the null space of task k, subject to the higher priority tasks, is defined in a similar manner to the single task case.

$$\begin{aligned}
\boldsymbol{N}_{k*}^{T} = \boldsymbol{N}_{k-1*}^{T} (\boldsymbol{I} - \boldsymbol{J}_k^{T} \bar{\boldsymbol{J}}_{k|k-1*}^{T}) &= (\boldsymbol{I} - \boldsymbol{J}_{k|k-1*}^{T} \bar{\boldsymbol{J}}_{k|k-1*}^{T}) \boldsymbol{N}_{k-1*}^{T} \\
\bar{\boldsymbol{J}}_{k|k-1*}^{T} &= \boldsymbol{\Lambda}_{k|k-1*} \boldsymbol{J}_k \boldsymbol{M}^{-1} \boldsymbol{N}_{k-1*}^{T}.
\end{aligned} \tag{4.26}$$

By working in task space, the input to the controller is a set of desired task goals. For a given set of goals, we can compute the task forces and map them

to joint space forces through Equation 4.24. The iteration starts with the higher priority task and propagates through the task priority graph $\mathcal{G}$, that is a priority ordered set (De Sapio and Park, 2010) (Algorithm 4.1).

Algorithm 4.1 Computes the driving torques for a set of goals.

Input: $\ddot{\boldsymbol{x}}_t,\ \forall t \in \mathcal{G}$
Output: $\boldsymbol{\tau},\ \boldsymbol{\tau}_c$
1: Select $\boldsymbol{M}$, $\boldsymbol{b}_c$ and $\boldsymbol{N}_\Phi$ model
2: $\boldsymbol{N}_{0*} = \boldsymbol{N}_\Phi$
3: $\boldsymbol{\tau}_0 = \boldsymbol{0}$
4: **for** $i \in \mathcal{G}$ **do**
5: Compute $\boldsymbol{f}_i$ from Equation 4.25
6: $\boldsymbol{\tau}_i = \boldsymbol{\tau}_{i-1*} + \boldsymbol{J}^T_{i|i-1*}\boldsymbol{f}_i$
7: Update $\boldsymbol{N}_{i*}$ from Equation 4.26
8: **end for**
9: $\boldsymbol{\tau} = \boldsymbol{\tau}_g + \boldsymbol{N}^T_{g*}\boldsymbol{\tau}_0$
10: Compute $\boldsymbol{\tau}_c$ from Equations 4.5 and 4.8 or 4.13
11: **return** $\boldsymbol{\tau}, \boldsymbol{\tau}_c$

4.2.5 Task Space Inverse Kinematics

There are multiple of ways to perform IK given a set of experimental recordings. The first is to use an optimization-based approach that minimizes an objective function. On the other hand, dynamics (Equation 4.16) can be utilized by computing a set of joint space forces that drive the model, which is subsequently integrated numerically. Optimization-based solutions can achieve better tracking in terms of the RMS marker error. In contrast, the dynamic solutions can take advantage of the structure of the model, e.g., by addressing different types of constraints that are commonly handled at a dynamic level.

Recently, different types of MoCap systems have been employed to record the movement of the subject. In this derivation, it is possible to combine marker-based and Inertial Measurement Unit (IMU) based MoCap systems to record the spatial properties of body segments. The following implementation, inspired by the approach suggested by (Khatib et al., 2009) (DMC), is enriched with some major improvements/modifications in order to accommodate for applications on more demanding movements (e.g., walking).

As presented in Figure 4.1, the input is the trajectory of the markers and/or spatial motion of the body segments in the case of an IMU based system. The task decomposition and generation module is responsible for constructing the desired task goals based on one or more marker per body, or to prescribe the movement of the segment if IMU measurements are available. The desired tasks are used by the ITSC to compute the generalized forces that actuate the model. For a given actuation, the plant is numerically integrated and advanced in time. The resulting simu-

lated tasks are compared with the measured ones and corrections are performed by the tracking controller.

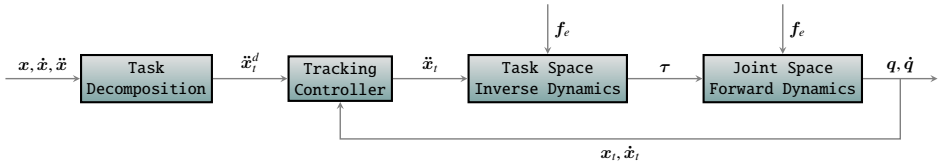


Figure 4.1: Architecture diagram of the TSIK method. The input to the system is a set of marker trajectories or other types of measurements (e.g. IMU). Markers are grouped based on their attachment location on the body and a set of task goals is extracted from the recorded motion. A tracking scheme is used to track the task goals (e.g. PD controller) and the desired goals are transformed into generalized forces by the ITSC. Finally, these forces actuate the model in a FD manner. The product of this algorithm is both a set of task goals and the simulated motion that reproduces the observations.

In case of a marker based system, the marker trajectories are recorded and the corresponding virtual markers are attached to the body segments, each containing zero or more markers. We denote the position of the markers attached to a body b and measured in the ground frame G at time t as

$$ {}^G\boldsymbol{p}_{:}^b(t) = \{{}^G\boldsymbol{p}_1^b(t), \ldots, {}^G\boldsymbol{p}_{m_b}^b(t)\},\ {}^G\boldsymbol{p}_i^b(t) \in \Re^3. \tag{4.27} $$

In order to generate a set of task goals, the markers are grouped per body segment and a tracking task is assigned. Some segments don't contain any marker, while others can have less than three markers. To define the 6D motion of a segment in space, at least three markers per body are required. Fortunately, since some segments are constrained with respect to their position in the body tree kinematic chain, their configuration can be determined even with less markers.

If a body contains a single marker, then a position type task is assigned ($\boldsymbol{p}(t) \in \Re^3$). If a segment can only rotate relative to its parent body, then an orientation type task is assigned ($\boldsymbol{\theta(t)} \in \Re^3$). When a segment is permitted to move freely (e.g., the floating base of the model) a spatial type task is assigned ($[\boldsymbol{\theta}(t),\ \boldsymbol{p}(t)]^T \in \Re^6$). As shown in Figure 4.2, which depicts two frames of gait, a 6D spatial task is assigned to the pelvis and a 3D orientation task is assigned to the femur and the tibia.

Regardless of the task type, the final goal is a desired acceleration that the task should achieve. In order to compute the desired goal a PD tracking controller is adopted

$$ \boldsymbol{a}(t) = \boldsymbol{a}_d(t) + k_p(\boldsymbol{x}_d(t) - \boldsymbol{x}(t)) + k_d(\boldsymbol{v}_d(t) - \boldsymbol{v}(t)) \tag{4.28} $$

where $\boldsymbol{a}(t)$ is the task goal acceleration, $\boldsymbol{x}_d(t)$, $\boldsymbol{v}_d(t)$ and $\boldsymbol{a}_d(t)$ the desired position, velocity and acceleration that are derived from the experimental marker trajecto-

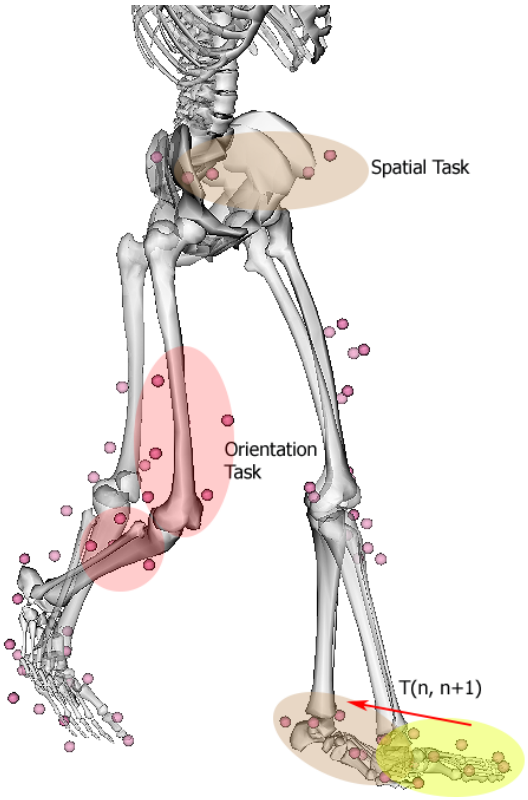


Figure 4.2: This figure presents two frames of recorded gait. An ellipse is placed to denote the grouping of the markers, where the color defines the type of the task assigned to each group as an example. The transition, transformation ($\boldsymbol{T}(n, n+1)$) between the two frames for the foot segment has been annotated.

ries, $\boldsymbol{x}(t)$ and $\boldsymbol{v}(t)$ the current position and velocity of the task and k_p, k_d are the tracking gains.

In order to track the goal effectively, the first (${}^G\boldsymbol{v}_i^b(t)$) and the second (${}^G\boldsymbol{a}_i^b(t)$) derivatives of the marker position have to be evaluated, by fitting smooth splines to the recorded trajectories. Estimating the orientation of the segment and the higher order derivatives from the marker positions requires a different strategy. We seek to find a transition, transformation matrix that maps a set of marker positions on frame n to a set of positions on the next frame $(n+1)$. The optimal rotation and translation between the corresponding 3D points can be obtained from the Kabsch algorithm (Horn, 1987) (Algorithm 4.2).

By precomputing the transition, transformation matrix ${}^G\boldsymbol{T}^b(n, n+1)$ and for a given initial pose ${}^G\boldsymbol{T}^b(0, 1)$, the transformation of frame k with respect to the ground frame can be evaluated.

$$ {}^G\boldsymbol{T}^b(k) = {}^G\boldsymbol{T}^b(k-1, k) \cdots {}^G\boldsymbol{T}^b(0, 1). \tag{4.29} $$

Algorithm 4.2 Kabsch algorithm for finding the optimal rotation and translation between two corresponding 3D point clouds.

Input: ${}^{G}\boldsymbol{p}_{\cdot}^{b}(n)$,${}^{G}\boldsymbol{p}_{\cdot}^{b}(n+1)$
Output: ${}^{G}\boldsymbol{T}^{b}(n, n+1) = \{\boldsymbol{R}, \boldsymbol{t}\},\ \boldsymbol{R} \in \Re^{3\times3},\ \boldsymbol{t} \in \Re^{3}$
1: Calculate the centroid of each point set $\boldsymbol{c}_A$ and $\boldsymbol{c}_B$ using $\boldsymbol{c} = \frac{1}{N}\sum_{i=1}^{N} {}^{G}\boldsymbol{p}_{i}^{b}$
2: Calculate the cross covariance matrix $\boldsymbol{H} = \sum_{i=1}^{N}({}^{G}\boldsymbol{p}_{i}^{b}(n) - \boldsymbol{c}_A)({}^{G}\boldsymbol{p}_{i}^{b}(n+1) - \boldsymbol{c}_B)^{T}$
3: Perform SVD $[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}]^{T} = \mathtt{svd}(\boldsymbol{H})$ on $\boldsymbol{H}$
4: Calculate the rotation matrix $\boldsymbol{R} = \boldsymbol{V}\boldsymbol{U}^{T}$
5: **if** $def(\boldsymbol{R}) < 0$ **then**
6: multiply 3rd column of $\boldsymbol{R}$ by -1 –Handle reflection case˝
7: **end if**
8: Find the translation $\boldsymbol{t} = -\boldsymbol{R}\boldsymbol{c}_A + \boldsymbol{c}_B$
9: **return** $\{\boldsymbol{R}, \boldsymbol{t}\}$

Unfortunately, the quantities of interest are the angular velocity ${}^{G}\boldsymbol{\omega}^{b}(t)$ and angular acceleration ${}^{G}\boldsymbol{\alpha}^{b}(t)$ of the frame. Quaternion representation is used, so that interpolation between consequent frames is more natural. For the interpolation a smooth curve should be always situated on the unit quaternion hypersphere $\|\boldsymbol{q}\| = 1,\ \boldsymbol{q} \in \Re^{4}$ (here $\boldsymbol{q}$ is a quaternion). By constructing quaternion splines that are twice differentiable in C^2 (Nielson, 2004), the higher order derivatives of the orientation $\boldsymbol{q}(t)$, $\dot{\boldsymbol{q}}(t)$ and $\ddot{\boldsymbol{q}}(t)$ can be computed. Then the physical quantities of the rotational motion are deduced

$$\boldsymbol{\omega} = 2\dot{\boldsymbol{q}}\boldsymbol{q}^{*} \tag{4.30}$$

$$\boldsymbol{\alpha} = 2(\ddot{\boldsymbol{q}}\boldsymbol{q}^{*} - (\dot{\boldsymbol{q}}\boldsymbol{q}^{*})^{2}) \tag{4.31}$$

where $\boldsymbol{q}^{*} = q_0 - q_1\hat{\boldsymbol{i}} - q_2\hat{\boldsymbol{j}} - q_3\hat{\boldsymbol{k}}$ is the quaternion conjugate.

In Equation 4.31 the power property of a quaternion is used to compute the term $(\dot{\boldsymbol{q}}\boldsymbol{q}^{*})^{2}$ (quaternions are closed over the product operator)

$$\begin{aligned} \boldsymbol{q} &= |\boldsymbol{q}|e^{\hat{\boldsymbol{u}}\theta} \\ \boldsymbol{q}^{p} &= |\boldsymbol{q}|^{p}(e^{\hat{\boldsymbol{u}}\theta})^{p} = |\boldsymbol{q}|^{p}(\cos(p\theta) + \hat{\boldsymbol{u}}\sin(p\theta)) \end{aligned} \tag{4.32}$$

where $\hat{\boldsymbol{u}}$ depicts a direction. After the task goals are precomputed, the ITSC can be used to derive the necessary joint space.

4.2.6 Task Space Computed Muscle Control

The TSCMC accepts as input a set of desired task goals, which may originate from a high level controller that encapsulates the logic of producing a synchronous movement. Alternatively, the goals computed by TSIK can be used as input. The

output of the algorithm consists of the observable variables, such as factors related to the tasks (task Jacobian, task compliant inertia mass matrix, task forces, etc.), joint space forces, muscle forces, excitation patterns and the simulated movement. The flow begins with the task goals, which are compared against the actual (simulated) goals and a corrective action is issued to the ITSC. In turn, the ITSC computes the joint space forces, accounting for any prioritization scheme between the individual tasks. These forces are projected onto muscle excitation patterns through an optimization procedure (subsection 4.2.7). Finally, the plant is numerically integrated and advanced in time for the given command and initial states. Figure 4.3 presents the block diagram of the TSCMC.

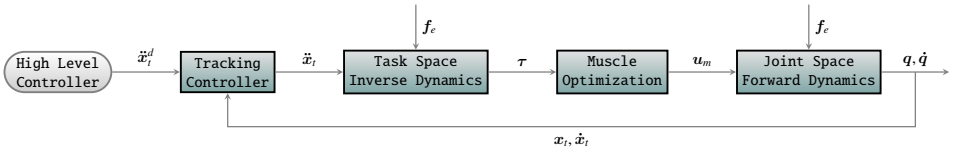


Figure 4.3: Architecture diagram of the TSCMC. The input can be the task goals originating from the TSIK or a set of goals from a different objective controller. The goals are tracked by an internal tracking scheme (e.g. PD controller), while the desired goals are used by ITSC to derive the required generalized forces. These forces are then mapped to muscle excitation patterns through an optimization procedure and the model is numerically integrated for the given muscle activation patterns.

In contrast to the proposed algorithm, the CMC (Thelen and Anderson, 2006) accepts joint trajectories as its input. The controller computes a set of muscle excitation patterns, by forming and optimization procedure, and drives the FD plant simulating a movement (joint space). In turn, the simulated trajectories are compared against the desired ones by a PD tracking controller. In a very similar manner, instead of tracking trajectories, the TSCMC algorithm tracks task goals.

4.2.7 Muscle Effort Assessment

Typically, more than one muscles span a single DoF. Thus, a unique mapping between muscle and joint space forces cannot be strictly defined. Formally, the relationship is given by

$$\boldsymbol{\tau} = -\boldsymbol{R}^T \boldsymbol{f}_m \tag{4.33}$$

where $\boldsymbol{R} \in \Re^{n \times m}$ is the moment arm fat matrix (Sherman et al., 2013; Deshpande et al., 2010), that maps from muscle space to joint space and $\boldsymbol{f}_m \in \Re^m$ is a vector of muscle forces with m the number of muscles. In terms of solution existence, $m > n$, thus there is an infinite number of solutions of muscle forces that can produce the required generalized forces

$$\boldsymbol{f}_m = -R^{+T}\boldsymbol{\tau} + (\boldsymbol{I} - \boldsymbol{R}^{+T}\boldsymbol{R}^T)\boldsymbol{f}_{m0} \tag{4.34}$$

where $\boldsymbol{f}_{m0} \in \Re^m$ is an arbitrarily selected vector that is being projected into the null space of $\boldsymbol{R}$. In the present, we address the muscle redundancy (dynamic redundancy) problem through optimization.

The Hill type muscle model has been widely adopted by the biomechanics community (Zajac, 1989; Thelen, 2003; Millard et al., 2013) mainly for efficient simulation and reliable parameter estimation. The muscle force is governed by its activation and contraction dynamics

$$\begin{aligned} \dot{\boldsymbol{a}}_m &= \boldsymbol{g}(\boldsymbol{u}_m, \boldsymbol{a}_m; \tau_a, \tau_d) \\ \dot{\boldsymbol{l}}_m &= \boldsymbol{h}(\boldsymbol{l}_m; \boldsymbol{\theta}_0) \end{aligned} \tag{4.35}$$

where $\boldsymbol{a}_m$ represents the muscle activation, which is a function of the excitation $\boldsymbol{u}_m$, τ_a, τ_d the activation and deactivation time constants and $\boldsymbol{l}_m$, $\dot{\boldsymbol{l}}_m$ the muscle length and lengthening speed. Vector $\boldsymbol{\theta}_0$ stands for a multitude of parameters, such as pennation angle, maximum velocity and muscle routing. The muscle force $\boldsymbol{f}_m$ depends on these quantities

$$\boldsymbol{f}_m = \boldsymbol{d}(\boldsymbol{a}_m, \boldsymbol{l}_m, \dot{\boldsymbol{l}}_m; \boldsymbol{\theta}_1) \tag{4.36}$$

and vector $\boldsymbol{\theta}_1$ denotes another set of parameters such as maximum isometric force.

For a given set of desired goals that are computed by the ITSC, the muscles must equilibrate the required generalized forces ($\boldsymbol{\tau}$). SO is a common technique for solving the muscle redundancy problem (Pandy, 2001). Unfortunately, models are imperfect and common modeling choices may introduce simulation instabilities. For a stable simulation the equality $\boldsymbol{\tau} = -\boldsymbol{R}^T\boldsymbol{f}_m(\boldsymbol{a}_m)$ should hold true, but the muscles may not be actually able to produce the right amount of force. To avoid problems of this kind, a set of reserve actuators is introduced to compensate for this defect. Furthermore, their contribution is adjusted using a penalty factor, so as to minimize their use with respect to muscles

$$\begin{aligned} \underset{\boldsymbol{a},\boldsymbol{\beta}}{\text{minimize}} \quad & \frac{1}{m}\sum_i^m a_i^2 + \gamma\frac{1}{n}\sum_i^n \left(\frac{\beta_i}{\beta_i^{\max}}\right)^2 \\ \text{subject to} \quad & \boldsymbol{\tau} = -\boldsymbol{R}\boldsymbol{f}_m(\boldsymbol{a}) + \boldsymbol{\beta}, \\ & \boldsymbol{0} \preceq \boldsymbol{a} \preceq \boldsymbol{1}, \\ & -\boldsymbol{\beta}^{\max} \preceq \boldsymbol{\beta} \preceq \boldsymbol{\beta}^{\max} \end{aligned} \tag{4.37}$$

where $\boldsymbol{\beta} \in \Re^n$ are the complementary residual activations, γ is the penalty constant and $\beta_i^{\max}$ is the maximum allowed value for the residual force i. Although the introduction of residuals does not have a direct physiological counterpart, their use allows for investigation of model weaknesses. The process of choosing γ and $\boldsymbol{\beta}^{\max}$ is as follows: 1) assign high values to $\boldsymbol{\beta}^{\max}$, $\gamma = 1$ and perform a first pass simulation, 2) update $\beta_i^{\max} = \max(\beta_i(t = start : end))\ \forall i$ and 3) experimentally

sample $\gamma \in [0, \infty)$ in order to assess whether residual forces are further reduced. We adopt an interior point method (Wachter and Biegler, 2006) with constraint and convergence tolerance of 10^{-5}. A final note is that the dynamic redundancy is handled in a forward manner (Shourijeh et al., 2016) (e.g., the muscles are activated and the developed forces are compared against the required generalized forces).

4.3 Results

In subsection 4.3.1, we present a comparison between the two alternative constrained models (A and B) derived in subsection 4.2.2. More specifically, we show that the choice of the constrained model can alter the required generalized forces without altering the motion, in the sense that the constraints can be used to reduce the command applied by the controller. Moreover, we demonstrate how to plan a movement behavior in task space and perform assessments without any experimental measurements.

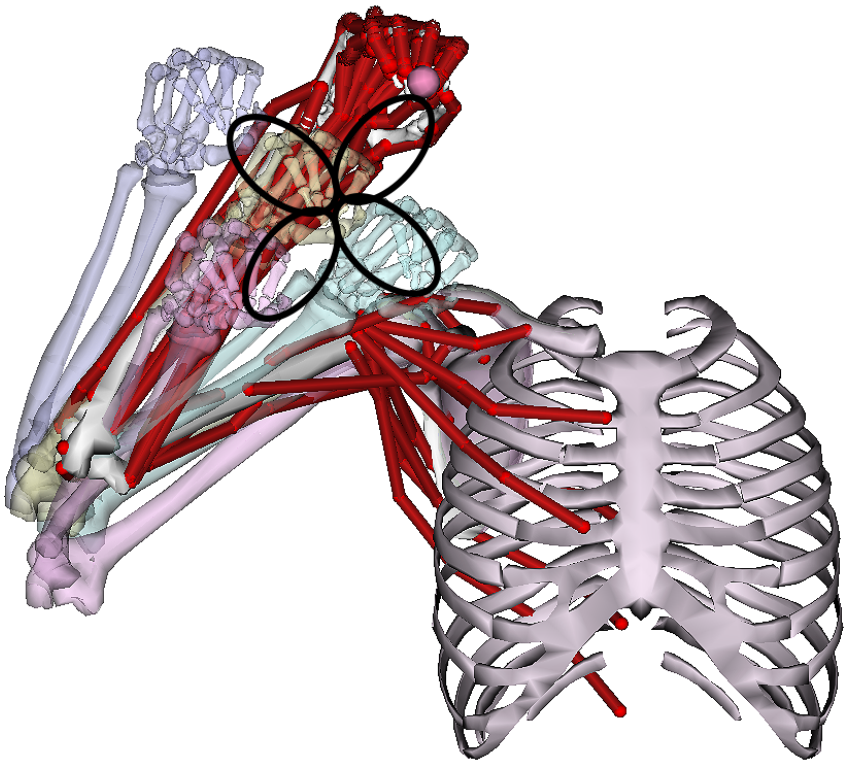


Figure 4.4: This figure depicts the time instances of a customly planned trajectory. The upper extremity model (Saul et al., 2014) consists of seven DoFs and forty-seven muscles. The model defines a set of constraint equations that regulate the movement of the shoulder complex.

For this particular experiment the upper extremity model (Saul et al., 2014) is used (Figure 4.4). It consists of seven DoFs, including shoulder rotation and elevation (thoracohumeral angle), wrist flexion, wrist deviation, elbow flexion, elevation plane of the shoulder and forearm rotation. There are forty-seven muscles (Schutte muscle model (Schutte et al., 1993)) spanning these joints. More importantly, it defines the movement of the shoulder complex through a set of constraints, making it a good candidate to evaluate the constrained models.

Next, we demonstrate that the two presented algorithms can be employed to solve the ID problem. A complex behavior of a gait movement along with the ground reaction forces recordings is used as a benchmark. For the IK problem, the marker error metric is adopted as a baseline for comparison, the tracking error of the proposed algorithm is relatively low compared to the state-of-the-art optimization-based approach (subsection 4.3.2). Finally, we estimate the required muscle patterns through TSCMC and compare the results against CMC algorithm (subsection 4.3.3).

Moreover, we demonstrate two additional case studies that are of particular interest. The developed controllers were proved to be coordinate invariant, thus they can also handle models that are defined in terms of absolute coordinates subsection 4.3.4. The proposed controllers can model systems of closed kinematic chains subsection 4.3.5, since they are commonly modeled by constraints through the virtual kinematic chain principle.

The `gait2354` model (Figure 4.2 shown without muscles), which has twenty-three DoFs and fifty-four muscles (Thelen muscle model (Thelen, 2003)), is used for the second experiment. The model features lower extremity joint definitions adapted from (Delp et al., 1990), low back joint and anthropometry from (Anderson and Pandy, 1999) and a planar knee model from (Yamaguchi and Zajac, 1989).

4.3.1 Comparison of the Constrained Models

We define the following trajectory to demonstrate the task space planning mechanism

$$\begin{gathered} x(t) = x(0),\ y(t) = y(0) + r\sin(\theta),\ z(t) = z(0) + r\cos(\theta) \\ \theta = \pi t,\ r = A\sin(2\theta) \end{gathered} \tag{4.38}$$

which generates the 3D movement presented in Figure 4.4. We can assign a position task on the hand and prescribe the desired trajectory following Equation 4.38. The muscle excitation patterns and other variables of interest can be assessed utilizing the TSCMC algorithm. Here we compare the two constrained models from Equations 4.9 and 4.12 with respect to the simulated generalized forces.

The simulated trajectory and generalized coordinates are shown in Figures 4.5 and 4.7, respectively. It is evident by these figures that the same movement is generated by the two models, both in terms of goal trajectory and model coordinates, something that is in good agreement with the motion-constraint separation principle (subsection 4.2.2).

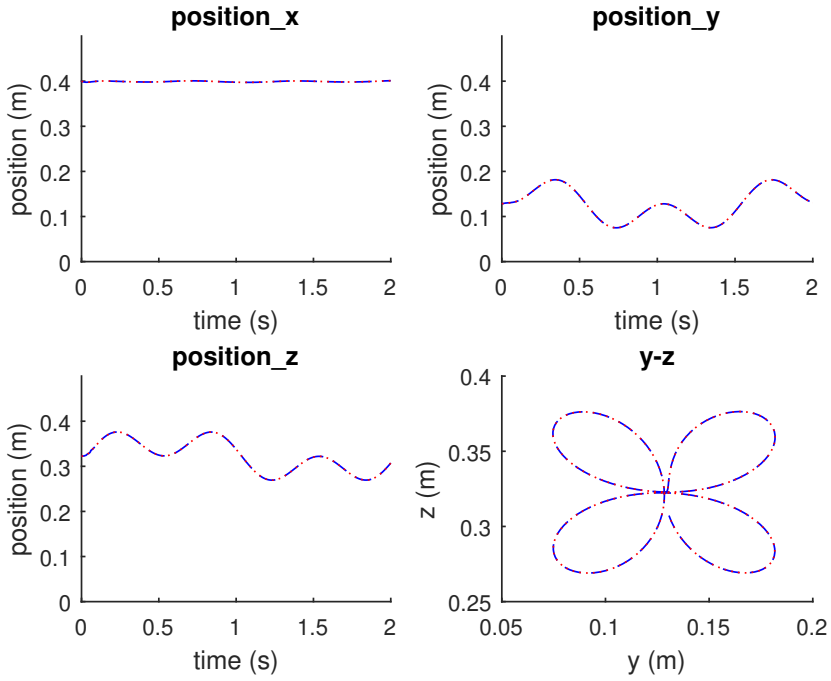


Figure 4.5: The simulated task trajectory of the upper limb model for the finger point task. The $y-z$ plane defines the resulting pattern. The trajectories of Model-A (Equation 4.9) are drawn with red dotted lines, while the blue dashed lines denote the trajectories of Model-B (Equation 4.12). Both models result in identical motion.

It is worth noting that the two models define different projection operators, meaning they issue different joint space forces despite resulting in identical movement behaviors. From the derivation, the total generalized forces is the sum of the motion related component (task and null space forces Equation 4.24). Figure 4.6 depicts the magnitude ($\|\boldsymbol{\tau}\|$) of each component for the two alternative models. Model-A requires larger motion forces compared to Model-B, although Model-A generates smaller constraint forces. This suggests that Model-B utilized the constraint forces more rigorously, minimizing the required motion forces ($\boldsymbol{\tau}^T\boldsymbol{\tau}$), while Model-A minimizes the inertia weighted command ($\boldsymbol{\tau}^T\boldsymbol{M}^{-1}\boldsymbol{\tau}$) (Righetti et al., 2011). Conclusively, the inverse model can impact the required generalized forces for tracking a movement behavior and consequently the muscle effort.

The fact that the two models produce different forces, but generate identical movement implies that this particular model is overconstrained. Let's assume that a model contains n unconstrained DoFs, c constraint algebraic equations and n_a active DoFs. Then the actual DoFs of the constrained model are of dimension $n_c = n - c$. We can distinguish three cases ((Righetti et al., 2011)):

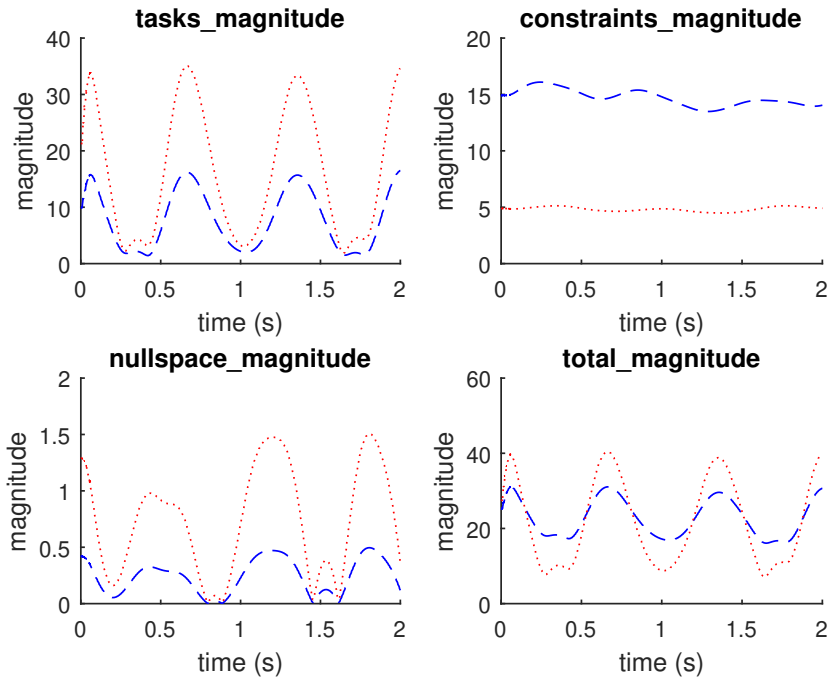


Figure 4.6: The computed forces that drive the model to perform the required movement. These forces are decoupled into motion related forces (task and null space forces) and constraint forces. A comparison of the magnitude values for the two models is presented. The trajectories of Model-A (Equation 4.9) are drawn with red dotted lines, while the blue dashed lines denote the trajectories of Model-B (Equation 4.12). Model-A results in larger motion forces command compared to Model-B, while Model-B results in larger constraint forces than Model-A.

1. $n_c > n_a$, the system is underactuated. There is at most one solution to the ID problem.
2. $n_c = n_a$, the system is fully actuated. There is a unique solution.
3. $n_c < n_a$, the system is overconstrained. There are infinite solutions (τ) that will achieve the desired goals (this is the case here).

4.3.2 Inverse Kinematics on Gait

The evaluation of the IK algorithm is presented for a single gait cycle. A comparison between the available optimization-based algorithm (`OpenSim`) and the TSIK is shown in Figure 4.8, in terms of the total marker RMS, min and max error. In general, the optimization-based algorithm behaves better in terms of tracking. Despite the presented algorithm does not minimize the tracking error by definition, results show relatively low and acceptable values for the total RMS (< 2cm).

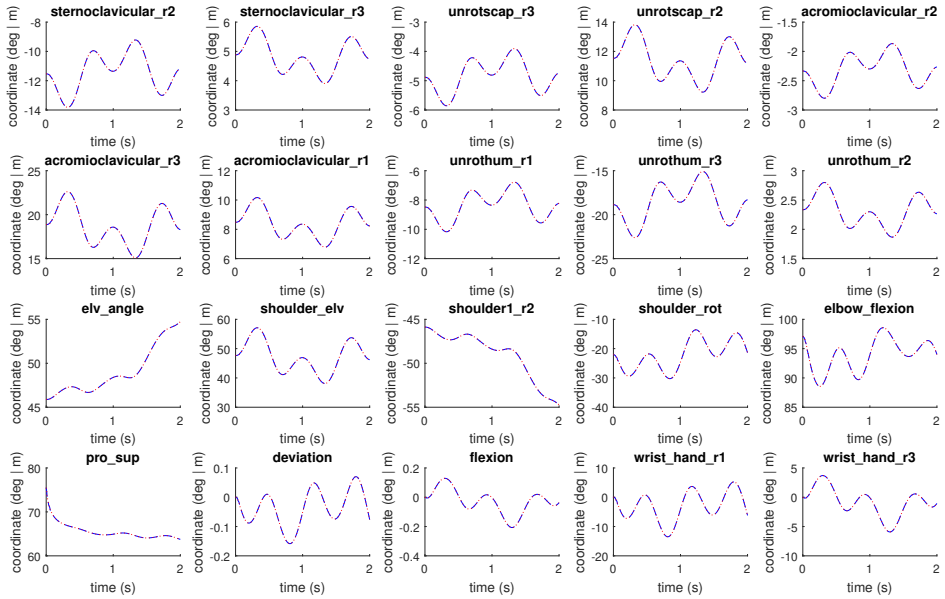


Figure 4.7: The simulated generalized coordinates ($\boldsymbol{q}$) of the upper limb model as a result for the defined custom trajectory. The trajectories of Model-A (Equation 4.9) are drawn with red dotted lines, while the blue dashed lines denote the trajectories of Model-B (Equation 4.12). Both models result in identical motion (residuals are zero).

As far as simulation performance is concerned, our approach is limited by its dynamic nature. More specifically, the explicit numerical integration scheme and the large accelerations required to track the movement takes up about 95% of the total execution time. Implicit integration schemes are not available in the current version of `OpenSim`, nevertheless it has been shown that they can improve the execution time dramatically (van den Bogert et al., 2011).

In the case of the optimization-based algorithms the weight of each error term contributed by each marker can be controlled separately, while the proposed algorithm relies on the task prioritization. The priority graph used for the gait movement follows the tree structure of the multibody system (high priority for the proximal segments and lower priority for the distal). It can be argued that this kind of priority structure is dependent on the type of movement that is analyzed as well as the presence of uncertainty in the measurements. For example, in gait related movements the floating base should have a higher priority, as was observed during simulation. Otherwise, in the case of a reach movement the hand tracking should be prioritized higher than the more proximal segments. Therefore, the user has to decide on the priority graph with respect to the movement under study.

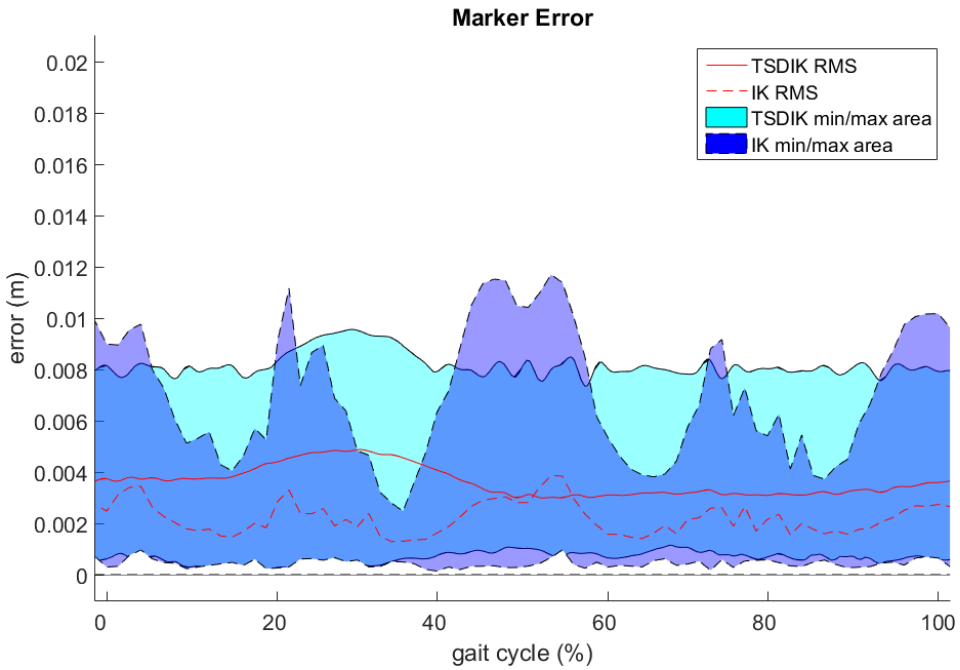


Figure 4.8: Comparison of the total RMS, min and max error between the TSIK and the optimization-based IK algorithms for a single gait cycle. The shaded area encloses the min and max marker error as its upper and lower bound respectively.

4.3.3 Inverse Muscle Driven Simulation on Gait

A comparison of the TSCMC with the CMC method for the same gait movement was performed. The task goals generated by the TSIK algorithm were supplied to the TSCMC method and similarly the output movement of the optimization-based IK was used by CMC. Figure 4.9 depicts the computed controls of the two algorithms. The estimated muscle patterns show very good agreement between the two methods, despite their different internal implementations. The execution time performance of the two algorithms was approximately the same.

The residual forces/torques as computed by the two algorithms are shown in Figure 4.10. The goal is to keep these residuals as low as possible. Major differences are present in the residual forces, where for some coordinates the proposed algorithm performs better and vice versa. The Residual Reduction Algorithm (RRA) (Delp et al., 2007; Hamner et al., 2010) method has not been applied to reduce the residuals prior to the simulation, since we didn't want to bias the comparison between the two methods. We can conclude that the proposed method is generally equivalent to CMC in terms of assessing the muscle effort distribution.

4.3.4 Absolute (Cartesian) Coordinates

In order to evaluate whether this framework can be used to simulate models that use absolute (Cartesian) coordinates, we introduce two identically functional models Figure 4.11. The model on the left (Figure 4.11a) consists of two bodies and two DoFs (generalized coordinates are used). The first and second bodies are permitted to rotate around the z-axis by a pin joint (one rotation Degree of Freedom (DoF)). The model on the right (Figure 4.11b) treats each body as a free body that is permitted to move freely with respect to the ground frame. Furthermore, the first body is constrained to the ground by a point constraint, which restricts relative translation of a point on the two bodies (three algebraic equations), and two constant angle constraints for restricting rotation around $\boldsymbol{x}$ and $\boldsymbol{y}$ axis (two algebraic equations). Similarly, the second body is constrained with respect to the first body. This model contains twelve DoFs and ten constraint algebraic equations. In terms of permissible movements behaves identically to the first model.

The goal is to use the task driven controller (ITSC) to drive the tip of the second body in performing a sinusoidal movement in the $\boldsymbol{y}$ direction

$$\boldsymbol{x}_d(t) = [x(0), y(0) + A\sin(2\pi t), z(0)]^T \tag{4.39}$$

where $\boldsymbol{x}_d(t)$ is the desired position at time t and $A = 0.2m$.

Results show that the ITSC is able to handle systems that use absolute coordinates. Figure 4.12 demonstrates that the two models (Figure 4.11) produce identical movement. We conclude that the presented task space controller is coordinate invariant as long as the fundamental quantities of the model (e.g., inertia mass matrix, Coriolis and centrifugal forces, gravity forces, constraint Jacobian, body Jacobian, etc.) are available.

4.3.5 Simulation of Closed Kinematic Chains

Closed kinematic chains are commonly handled (Jain, 2011) by cutting the kinematic chain and introducing a constraint that connects the two (the virtual kinematic chain principle (Campos et al., 2005)). When generalized coordinates are used the closed chain can be separated by cutting a body instead of a joint, so that the original coordinates of the model are preserved. When absolute coordinates are used the closed chain can be segmented at the joint level. By doing so instead of having a closed chain two separate open chains are constructed, that are appropriately constrained with each other. As a consequence, a controller that accounts for the constraint forces can handle closed kinematic chains.

As presented in Figure 4.13 the model consist of three bodies that are permitted to rotate around the z-axis. The top most body is connected by two joints with the two supporting bodies (left, right). The underlying multibody dynamics engine (Simbody) implicitly transforms closed kinematic chains by cutting a body rather than a joint. Mass properties are divided between the two halves. Then a weld constraint (six algebraic equations) is used to reattach the two halves.

To prove that the proposed controller (ITSC) is capable of controlling this system an orientation task is assigned to the leftmost body (body 1)

$$\boldsymbol{\theta}_d(t) = [\theta_x(0), \theta_y(0), \theta_z(0) + \frac{\pi}{2}\sin(2\pi t)]^T \tag{4.40}$$

where $\boldsymbol{\theta}_d(t)$ is the desired orientation at time t. Figure 4.14 depicts the simulated body orientations, which validate the correct behavior of the controller. Therefore, the proposed controller is capable of addressing topologies that contain closed kinematic chains.

4.4 Discussion-Conclusions

Although joint space representation is the de facto standard for formulating the underlying EoMs and dynamics simulation methods, it may be sub-optimal for a certain range of problems. Common movement behaviors can be described through a well-defined set of interrelating task goals, which is especially evident in the process of planning a movement in the task space. Projecting the EoMs in the domain of the task provides a straightforward mechanism to translate a movement behavior to the biomechanical counterpart. The underlying complexity of mapping the abstract task primitive to a movement is effectively handled by the underlying controllers. This work mainly focuses on motion primitive tasks, without necessarily restricting applications to those alone (Sentis, 2007). Extension to other types of tasks can shed some light towards understanding the complex interactions between the nervous system and the execution of multiple task goals.

The underlying controller (ITSC) was designed to handle constrained multibody systems, facilitating the study of broader types of problems. It is straightforward to verify that constrained task space projection is coordinate invariant, in the sense that the provided algorithms can address systems that use absolute coordinates (Dziewiecki et al., 2014). This is true since the underlying projection operators are formed from the fundamental system quantities (e.g., the inertial mass matrix, the constraint and task Jacobian) that can be derived under any coordinate system. Another virtue of constraint modeling is that the presented controllers can address systems of closed kinematic chains, since they are commonly modeled by constraints as a consequence of the virtual kinematic chain principle (Campos et al., 2005; Jain, 2011). It is important to note that closed kinematic chains are found to be the rule in many biological systems rather than the exception (Levin et al., 2017). As constraints can be used to model biological systems more closely, results demonstrate that they can alter the required muscle forces. Consequently, the effect of constraint modeling on the assessment of muscle forces remains a subject open to study.

As a proof of concept we showed that task-based schemes can be used to solve the IK problem. Experimental evaluation indicates an acceptably low tracking error for a complex gait movement. The execution time of the algorithm is inferior as

compared to the optimization-based approach, due to a combination of an explicit numerical integration scheme and the high acceleration goals required to track the movement. The authors of (van den Bogert et al., 2011) suggested that implicit numerical integration can dramatically improve the execution time of the FD methods. Dynamics-based IK approaches can be of great interest, since constraints are primarily resolved at a dynamic level as opposed to kinematic methods, they can be paired with temporal filtering techniques for reducing marker error artifacts ((De Groote et al., 2008)) and allow for a combination of different types of input (e.g. IMU and marker recordings).

We showed that the TSCMC algorithm is able to establish a link between the task(s) goals and muscle excitation patterns. Moreover, we compared its performance to the CMC algorithm and demonstrated negligible differences. The algorithm can be used in combination with a high level controller that generates a set of abstract task goals in order to perform simulation prior to any experimental data collection. This is of great importance, since simulations can be arranged effortlessly and intuitively. Of equal importance is the fact that our method allows deduction of important variables that can be used to improve the underlying model and the experimental setup a priori. Finally, it offers a novel, abstract point of view and control, which can prove to be advantageous towards further integration with high level models of the precommand level (Zhang et al., 2009; DeWolf and Eliasmith, 2011).

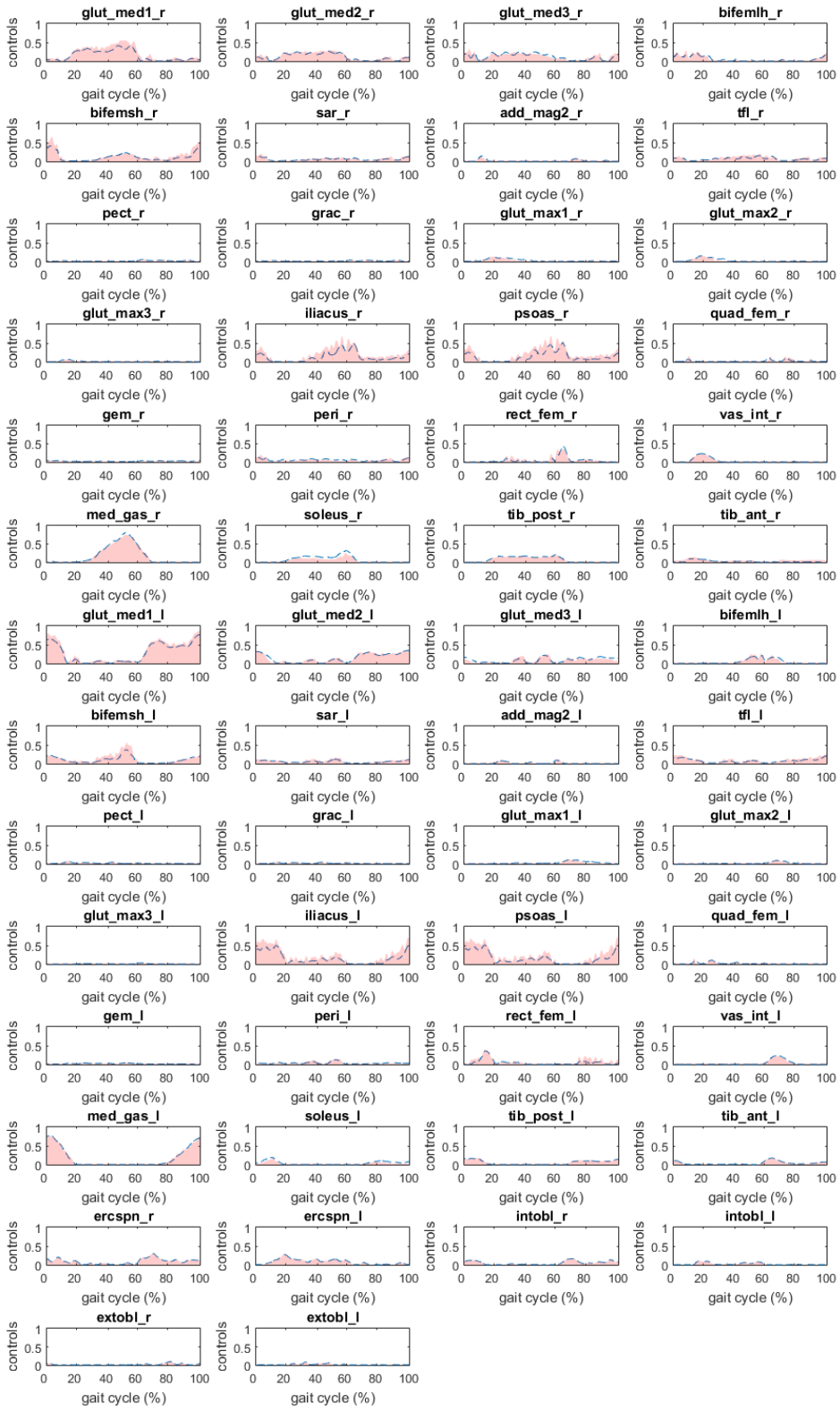


Figure 4.9: This figure presents the muscles excitations for a single gait cycle in comparison between CMC (red solid line) method and the proposed TSCMC (blue dashed line). There are some minor differences, as the two methods have their implementation specifics, but in general they agree.

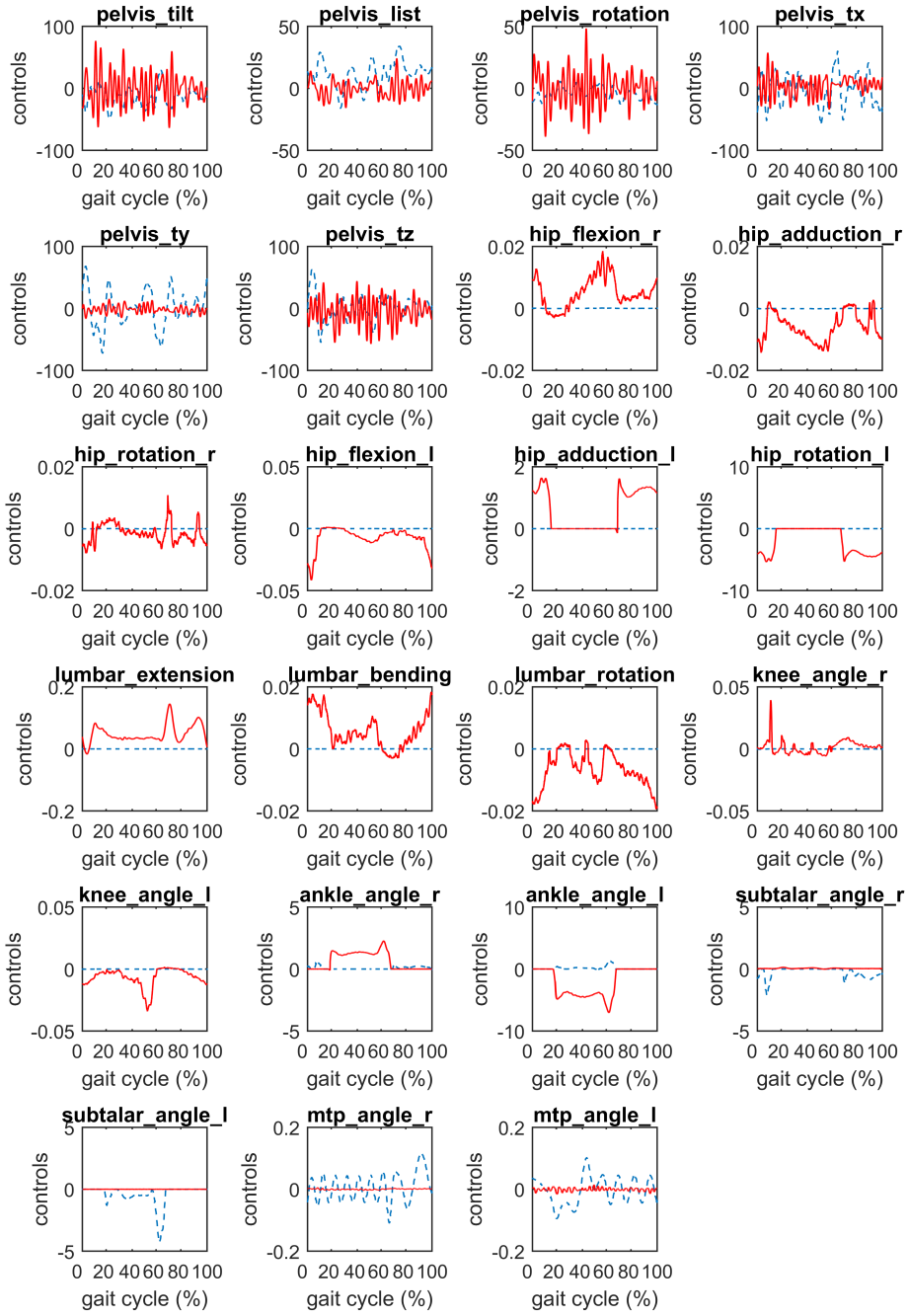
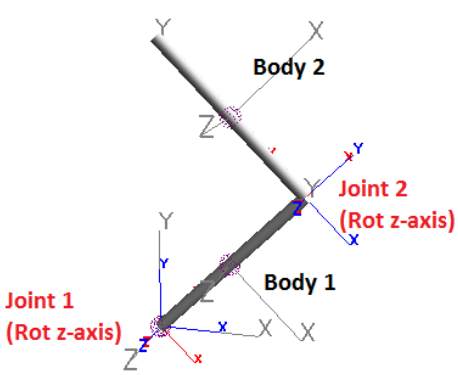
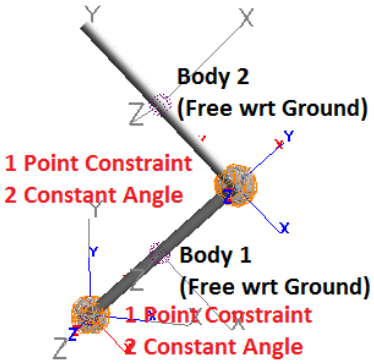


Figure 4.10: This figure presents the residual forces/torques for a single gait cycle in comparison between CMC (red solid line) method and the proposed TSCMC (blue dashed line).



(a) The unconstrained model is composed of two bodies and two pin joints (1 and 2). A pin joint permits a rotational movement around an axis (in this case is the z-axis).



(b) The free body model is composed of two bodies that are permitted to move freely with respect to the ground frame (absolute coordinates are used). To enforce similar functional behavior with respect to the model on the left, the bodies are further constrained at the joint level by a point constraint and two constant angle constraints (for each joint).

Figure 4.11: Left: an unconstrained model (generalized coordinates) with two DoFs. Right: constrained model that uses absolute coordinates twelve DoFs and ten constraint algebraic equations.

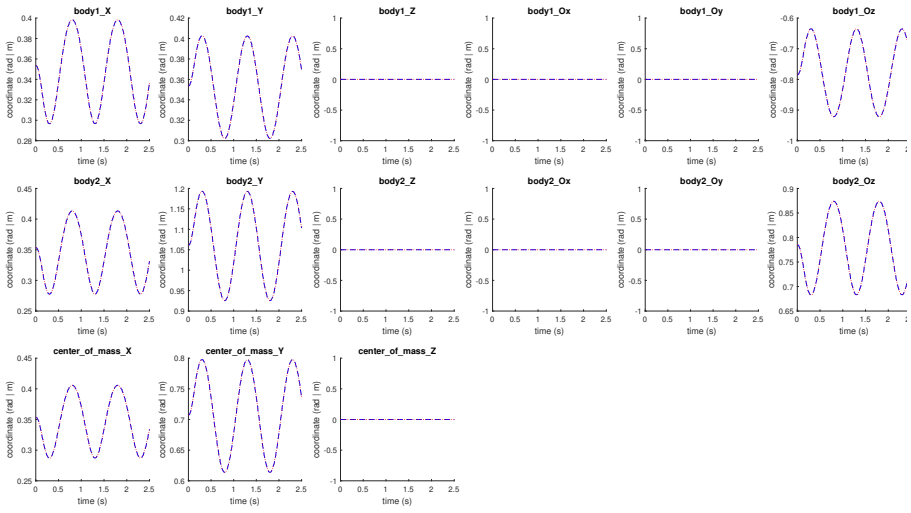


Figure 4.12: Body kinematics analysis of the two DoF (generalized coordinates) (blue) and the model that uses absolute coordinate (red). XYZ quantities correspond to the translational components, while $Oxyz$ quantities represent the orientation of the bodies. The task goal is a sinusoidal movement of the tip of the second body in the y direction. The two models produce identical movements (residuals are zero).

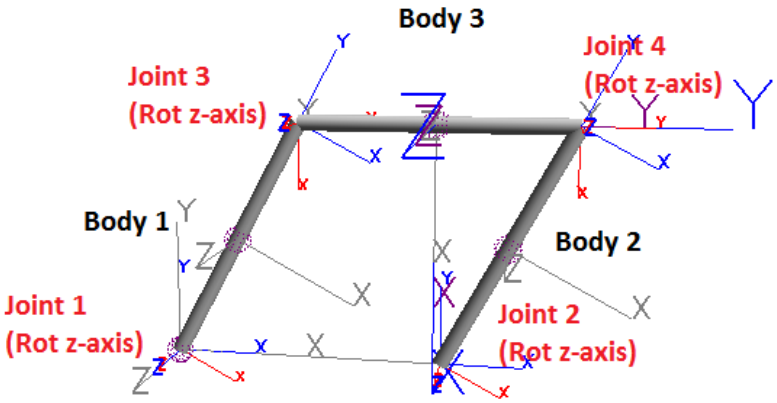


Figure 4.13: A model of a closed kinematic chain topology. The first body (left) is permitted to rotate around the z-axis with respect to the ground frame. The second body (right) is connected to the ground by an offset of the length of the third body (top) and it is also permitted to rotate around the z-axis. The third body is connected with the first and second bodies by two pin joints.

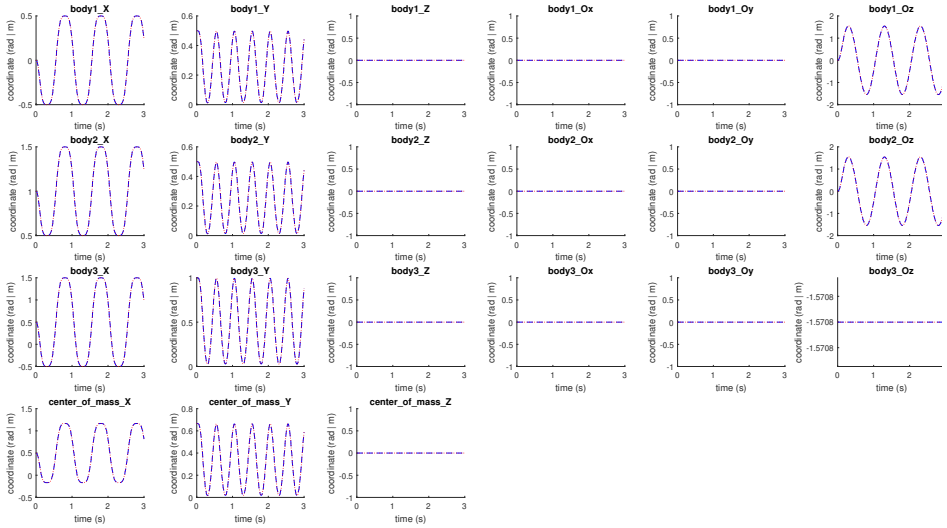


Figure 4.14: Body kinematics analysis of the two models Model-A (blue), Model-B (red) on a closed kinematic chain model. XYZ quantities correspond to the translational components, while $Oxyz$ quantities represent the orientation of the bodies. The goal is to rotate body 1 along the z-axis. Since body 2 is connected to body 1 by body 3 (top) their orientation is identical. Furthermore, because of the structure we expect that body 3 do not rotate at all for this particular experiment. The two models produce identical movements (residuals are zero).

Chapter 5

Modeling Kinematic and Dynamic Redundancy Using Null Space Projection

The problem of musculoskeletal redundancy is examined in this chapter, by developing methods that can cope with indeterminacy. As musculoskeletal systems are inherently redundant, i.e., an underdetermined system of equations, the validity of the obtained results is in question. The redundant nature of the musculoskeletal system introduces variability/uncertainty in simulated quantities leading to misinterpretation of the results if ignored. Therefore, this chapter lay the theoretical foundations to study and interpret results obtained from redundant musculoskeletal systems.

The coordination of the human musculoskeletal system is deeply influenced by its redundant structure, in both kinematic and dynamic terms. Noticing a lack of a relevant, thorough treatment in the literature, we formally address the issue in order to understand and quantify factors affecting the motor coordination. We employed well-established techniques from linear algebra and projection operators to extend the underlying kinematic and dynamic relations by modeling the redundancy effects in null space. We distinguish three types of operational spaces, namely task, joint and muscle space, which are directly associated with the physiological factors of the system. A method for consistently quantifying the redundancy on multiple levels in the entire space of feasible solutions is also presented. We evaluate the proposed muscle space projection on segmental level reflexes and the computation of the feasible muscle forces for arbitrary movements. The former proves to be a convenient representation for interfacing with segmental level models or implementing controllers for tendon driven robots, while the latter enables the identification of force variability and correlations between muscle groups, attributed to the system's redundancy. Furthermore, the usefulness of the proposed framework is demonstrated in the context of estimating the bounds of the joint reaction loads, where we show that misinterpretation of the results is possible if the null

space forces are ignored. This chapter presents a theoretical analysis of the redundancy problem, facilitating application in a broad range of fields related to motor coordination, as it provides the groundwork for null space characterization. The proposed framework rigorously accounts for the effects of kinematic and dynamic redundancy, incorporating it directly into the underlying equations using the notion of null space projection, leading to a complete description of the system.

5.1 Introduction

As highlighted previously, the human musculoskeletal system has a redundant structure, i.e., there are more DoFs than those required to perform a certain task (kinematic redundancy) and each DoF is actuated by multiple muscles (dynamic redundancy). This over-availability poses numerous challenges in the process of modeling and simulation that can negatively affect the validity of the models and the obtained results, rendering their application frequently inappropriate for clinical practice. Despite the fact that mathematical tools for studying redundant systems do exist, they are still not widely adopted in the field of musculoskeletal simulation. Proper formalization and interpretation of redundancy can significantly improve our understanding of the motor coordination problem.

We introduce an extended derivation of the underlying equations that govern the kinematic and dynamic evolution of the system, accounting explicitly for the different types of redundancy presented. The derivations rely on the theory of linear projection operators and the properties of the associated subspaces. An intuitive analogy is the projection of a point from 3D space to a point on a plane (2D). When mapping from a high-dimensional subspace the problem is straightforward, however, the projection from 2D to 3D is not unique. By incorporating the null space one can identify the feasible solution space and outline the factors that influence it.

What makes musculoskeletal systems challenging is not only the process of creating accurate mathematical representations, but also the assumptions that are introduced when they are analyzed. Traditional solutions of the muscle redundancy problem (Anderson and Pandy, 2001), such as minimum effort criterion, are very popular and have roots in a compelling evolutionary hypothesis, which states that motor control systems evolve to minimize energy expenditure during movement. Undeniably, single solution methods are of great importance, however, the assumptions that are introduced can severely hinder the validity of the obtained results. This is especially evident in the case of rigidity (Le Cavorzin et al., 2003) in Parkinson's disease, which is characterized by the inability of the muscles to relax. Clearly, a particular solution will not only bias the results, but will also affect the calculation of other quantities that depend on the muscle forces, like joint reaction loads. On the other hand, identification of the feasible solution space can help to properly interpret results obtained from the redundant musculoskeletal systems.

The main contributions can be summarized below:

- Identify the properties of kinematic redundancy when performing a specific

task and establish the kinematic and dynamic relations (Subsections 5.2.1 and 5.2.2). Determine the properties of dynamic redundancy and their relationship to joint and muscle space quantities both at a kinematic and dynamic level (Subsections 5.2.3 and 5.2.4). This description can potentially enable the modeling and evaluation of hypotheses related to normal and pathological conditions presented in the coordination process (e.g., slack muscle disorder, increase in joint stiffness, rigidity, etc.). We also show that these disorders manifest themselves in the null space.

- Propose an alternative representation of the EoMs in muscle space (subsection 5.2.4). Muscle space projection is a convenient representation for interfacing with segmental level models (e.g., reflexes) (Zhang et al., 2009) or implementing controllers for tendon driven robots (Deshpande et al., 2010). Furthermore, it has the highest number of DoFs and permits coordination of different aspects of the movement (e.g., co-contraction) that are uncontrollable in subspaces of lower dimensionality. Task and joint space projections are specializations of muscle space.

- Establish an effective model for quantifying redundancy with respect to the movement task, the neuro-muscular model (e.g., linear/nonlinear muscle model, synergy encoding (Steele et al., 2015)) and the anatomical characteristics of the muscle routing (the muscle moment arm and its null space) (subsection 5.2.5). This approach is advantageous as compared to the local techniques (e.g., minimum effort optimization (Anderson and Pandy, 2001)) that are suitable for finding only a particular solution, because it identifies the entire solution space of possible muscle force realizations as well as the factors that influence its structure. The importance of identifying the entire solution space when estimating the bounds of the joint reaction loads is demonstrated and we further show that misinterpretation of the results is possible if local techniques are used as compared to the proposed approach.

The results demonstrate working examples of the developed models. The projected EoMs in muscle space provide a new perspective for studying the musculoskeletal system in this domain. In subsection 5.3.1, we design a posture controller that encapsulates the characteristics of the internal regulation process performed in the spinal cord and study the response of the system under external disturbances. Muscle space projection proves to be a convenient representation to solve this problem, since no conversion to muscle activation is required.

In subsection 5.3.2, we present a case study in which we determine the force variability and correlations between different muscle groups for arbitrary movements and show that there is a significant systematic correlation between different muscle pairs that emerges from the muscle routing properties and the functional constraints of the task. This can be used to estimate the importance of actors/muscles during the movement, to identify synergies and classify the various control strategies available (Kutch and Valero-Cuevas, 2012; Inouye and Valero-Cuevas,

2016), considering the entire solution space. This not only demonstrates an effective approach for finding the family of possible solutions, but also exposes the structural relations of the null space solutions.

In subsection 5.3.3, we perform a joint reaction analysis on a gait, using a realistic full body model, appropriately accounting for the null space muscle forces that do not alter the movement. Accurate joint reaction force estimation requires reliable assessment of the muscle forces. As this is an impossible task, we demonstrate the usefulness of identifying the feasible muscle forces in order to determine the limits of the reaction loads. The large variance that is evident from the results confirms that misinterpretation of the reaction loads is possible if the null space contribution is ignored.

The kinematic and dynamic relationships were extended in order to model the musculoskeletal redundancy effects, using the notion of null space. We investigate the properties that govern the existence and uniqueness of these projections with emphasis on their physiological counterparts. Thus a complete theoretical analysis is constructed, which can be extended for many practical applications related to the motor coordination. The source code along with any related material for this chapter are publicly available (section A.1), providing simple examples so that the readers can reproduce, understand and reuse the presented methods.

5.1.1 Related work

Motor control is the process by which humans and animals use their nervous system to activate and coordinate the muscles and limbs involved in the performance of a movement. The redundant nature of the musculoskeletal system was noted over fifty years ago (Bernshtein, 1967), highlighting the over-availability of DoFs for most common tasks. Subsequently, researchers have created formal descriptions of the physiological process in order to identify the main variables responsible for coordinated movement (Latash et al., 2010; Loeb, 2001; Houk and Rymer, 2011; Stein, 1982). Important hypotheses relative to these have been developed, improving our understanding of the central coordination problem (Feldman, 1986; Latash et al., 2002).

It has been shown that the musculoskeletal system can act as a "filter" that transforms the feasible muscle activations into a set of feasible wrenches (Valero-Cuevas et al., 2009; Valero-Cuevas, 2009a), revealing an important direction towards a holistic identification of the system capabilities. We propose a more general approach (subsection 5.2.5), where the computed feasible muscle forces are both action-specific, accounting for the dynamic evolution of the motion, and satisfy the physiological constraints of the muscles, outlining the various factors that affect the solution space. Since the entire solution space is obtained using our framework, we can quantifiably explore the available strategies to the CNS. This is useful because it is still unclear how the CNS explores the musculoskeletal redundancy by controlling a low-dimensional subset of synergies (Steele et al., 2015; Razavian et al., 2015) and whether there is a correlation with the structure of the

musculoskeletal system and the functional constraints each task (Valero-Cuevas, 2009a; Kutch and Valero-Cuevas, 2011). We will show that both the structure of the model and the task (movement) indirectly impose some degree of muscle (anti-)synergies (subsection 5.3.2).

The use of linear projection operators have been successfully utilized in the field of robotics (Khatib, 1995; Fisher and Mujtaba, 1991; Aghili, 2005) in order to separate and simplify the control problem into multiple objectives. It has been shown that task-based projection can be used effectively for planning and simulation of constrained musculoskeletal systems (Khatib et al., 2009; Stanev and Moustakas, 2018) and furthermore, it provides the means to identify the kinematically redundant DoFs. In a similar vein, we explore the null space properties as well as the potential applications that exploit not only the kinematic redundancy, but also the dynamic redundancy of the system. Detailed derivations are presented in order to provide the reader with the appropriate background so as to understand the implications of the proposed models for addressing redundant systems.

5.2 Methods

5.2.1 Relation Between Task and Joint Space Quantities

This section focuses on the study of kinematic redundancy, which in turn will pave the way towards addressing the dynamic redundancy. Task space projection, a method that can reveal the properties of kinematic redundancy, has been thoroughly studied (Khatib, 1995; Khatib et al., 2009) and we will briefly introduce the basic concept here. Establishing a bilateral link between task execution (e.g., movement of the hand) and muscle coordination is essential, as it is more convenient to interpret observations in task space rather than joint space. Furthermore, it is of great importance to transform the complex movement of the musculoskeletal system as a composition of multiple interleaved task goals, which may then be studied separately (Baerlocher, 2001).

The task space position ($\boldsymbol{x}_t$) is given as a function of the generalized coordinates ($\boldsymbol{q}$)

$$\boldsymbol{x}_t = \boldsymbol{g}(\boldsymbol{q}),\ \boldsymbol{x}_t \in \Re^d,\ \boldsymbol{q} \in \Re^n,\ d \leq n \tag{5.1}$$

implying that the generalized coordinates fully describe the task. A task has an abstract meaning, it can be interpreted either as a position, an orientation or a spatial primitive, which is a combination of position and orientation. The first derivative of Equation 5.1 (the dot notation depicts a derivative with respect to time) is given by

$$\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t(\boldsymbol{q})\dot{\boldsymbol{q}},\ \boldsymbol{J}_t(\boldsymbol{q}) = \begin{bmatrix} \frac{\partial \boldsymbol{g}_1}{\partial q_1} & \cdots & \frac{\partial \boldsymbol{g}_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{g}_d}{\partial q_1} & \cdots & \frac{\partial \boldsymbol{g}_d}{\partial q_n} \end{bmatrix} \tag{5.2}$$

where the task Jacobian matrix ($\boldsymbol{J}_t \in \Re^{d \times n}$) defines a mapping from joint to task space ($\Re^n \rightarrow \Re^d$) and in most cases is a fat matrix (Proposition A.4).

Remark 5.1. *[Task Jacobian matrix (fat)]*

(a) The system $\boldsymbol{J}_t\dot{\boldsymbol{q}} = \boldsymbol{0}$ has infinite solutions.

(b) If rank$(\boldsymbol{J}_t) = d$ then the system $\boldsymbol{J}_t\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}_t$ is consistent (one or more solutions).

(c) If rank$(\boldsymbol{J}_t) < d$ then the system $\boldsymbol{J}_t\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}_t$ is inconsistent, e.g., $\exists\dot{\boldsymbol{x}}_t : \boldsymbol{J}_t\dot{\boldsymbol{q}} \neq \dot{\boldsymbol{x}}_t \Rightarrow \dot{\boldsymbol{x}}_t \notin \mathbb{C}(\boldsymbol{J}_t)$.

(d) The MPP matrix of $\boldsymbol{J}_t$ is the right inverse ($\boldsymbol{J}_t\boldsymbol{J}_t^+ = \boldsymbol{I}$).

The inverse of Equation 5.2 must be augmented so as to account for the kinematic redundancy (Equation 2.7) when mapping from task space ($\Re^d$–low-dimensional) to joint space ($\Re^n$–high-dimensional)

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\|} + \dot{\boldsymbol{q}}_{\perp} = \boldsymbol{J}_t^+\dot{\boldsymbol{x}}_t + \boldsymbol{N}_{\boldsymbol{F}_{J_t}}\dot{\boldsymbol{q}}_0,\ \boldsymbol{N}_{\boldsymbol{F}_{J_t}} = (\boldsymbol{I} - \boldsymbol{J}_t^+\boldsymbol{J}_t) \tag{5.3}$$

where $\boldsymbol{N}_{\boldsymbol{F}_{J_t}} \in \Re^{n \times n}$ represents the right null space of $\boldsymbol{J}_t$ (Definition 2.1) and $\dot{\boldsymbol{q}}_0$ an arbitrarily selected vector in $\Re^n$. The derivative of Equations 5.2 and 5.3 with respect to time are provided below

$$\ddot{\boldsymbol{x}}_t = \dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}} + \boldsymbol{J}_t\ddot{\boldsymbol{q}} \tag{5.4}$$

$$\ddot{\boldsymbol{q}} = \dot{\boldsymbol{J}}_t^+\dot{\boldsymbol{x}}_t + \boldsymbol{J}_t^+\ddot{\boldsymbol{x}}_t + \dot{\boldsymbol{N}}_{\boldsymbol{F}_{J_t}}\dot{\boldsymbol{q}}_0 + \boldsymbol{N}_{\boldsymbol{F}_{J_t}}\ddot{\boldsymbol{q}}_0 \tag{5.5}$$

where $\ddot{\boldsymbol{q}}_0$ is also an arbitrarily selected vector in $\Re^n$.

The task Jacobian defines a dual relation between motion and force quantities. The virtual work principle can be used to establish the link between task and joint space forces

$$\begin{aligned} \boldsymbol{\tau}^T\delta\boldsymbol{q} &= \boldsymbol{f}_t^T\delta\boldsymbol{x}_t \\ \boldsymbol{\tau}^T\delta\boldsymbol{q} &= \boldsymbol{f}_t^T\boldsymbol{J}_t\delta\boldsymbol{q} \\ \boldsymbol{\tau} &= \boldsymbol{J}_t^T\boldsymbol{f}_t \end{aligned} \tag{5.6}$$

where $\boldsymbol{J}_t^T : \Re^d \rightarrow \Re^n$ defines a mapping and its properties in the tall matrix case (Proposition A.5).

Remark 5.2. *[Task Jacobian Transpose (tall)]*

(a) If rank$(\boldsymbol{J}_t^T) = d$ the homogeneous system $\boldsymbol{J}_t^T\boldsymbol{f}_t = \boldsymbol{0}$ has a unique, trivial solution.

(b) If rank$(\boldsymbol{J}_t^T) < d$ the homogeneous system $\boldsymbol{J}_t^T\boldsymbol{f}_t = \boldsymbol{0}$ has infinite solutions.

(c) The system $\boldsymbol{J}_t^T \boldsymbol{f}_t = \boldsymbol{\tau}$ is inconsistent for some $\boldsymbol{\tau} \in \Re^n$ (a solution exists iff $\boldsymbol{\tau} \in \mathbb{C}(\boldsymbol{J}_t^T)$).

(d) The MPP matrix of $\boldsymbol{J}_t^T$ is the left inverse ($\boldsymbol{J}_t^{+T} \boldsymbol{J}_t^T = \boldsymbol{I}$).

In a similar manner as before, the generalized forces $\boldsymbol{\tau}$ can be expressed as the sum of the two mutually orthogonal vectors $\boldsymbol{\tau}_{\parallel} \in \mathbb{C}(\boldsymbol{J}_t^T)$ and $\boldsymbol{\tau}_{\perp} \in \mathbb{N}(\boldsymbol{J}_t)$ in order to account for the null space contribution

$$\boldsymbol{\tau} = \boldsymbol{J}_t^T \boldsymbol{f}_t + (\boldsymbol{I} - \underbrace{(\boldsymbol{J}_t^+ \boldsymbol{J}_t)^T}_{\boldsymbol{J}_t^+ \boldsymbol{J}_t}) \boldsymbol{\tau}_0 = \boldsymbol{J}_t^T \boldsymbol{f}_t + \boldsymbol{N}_{\boldsymbol{F}_{\boldsymbol{J}_t}} \boldsymbol{\tau}_0 \tag{5.7}$$

$$\boldsymbol{f}_t = \boldsymbol{J}_t^{+T} \boldsymbol{\tau} \tag{5.8}$$

where $\boldsymbol{\tau}_0 \in \Re^n$ is an arbitrarily selected vector and Equation 5.8 the inverse mapping of Equation 5.7. Note that the null space complements the kinematic and dynamic relations in the sense that there are several combinations of joint space velocities, accelerations and generalized forces that bear no effect to the corresponding task space velocities, accelerations and forces. Furthermore, this redundancy is explored by the same null space term ($\boldsymbol{N}_{\boldsymbol{F}_{\boldsymbol{J}_t}}$), both at a motion and force level by a virtue of the MPP properties (Definition A.2).

Remark 5.3. *[Task Motion-Force Duality]*
The projection operators derived from $\boldsymbol{J}_t^+$ and $\boldsymbol{J}_t^T$ span the same subspaces.

Proof. Given that $\boldsymbol{J}_t^+$ is the MPP matrix and the definition of SVD ($\boldsymbol{J}_t = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$), the following hold true:

(a) $\boldsymbol{J}_t^T = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T$

(b) $\boldsymbol{J}_t^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^T$

thus their projection operators span the same subspaces. □

5.2.2 Task Space Equations of Motion

In the previous section, the various relations between motion and force quantities of the kinetically redundant system were established. The current section continues with the derivation of the EoMs that relate the task acceleration to task space forces. Although, the task space projection was thoroughly examined in chapter 4, we would like to reexamine the properties of the null space under the perspective of this chapter. Let the joint space EoMs have the following form

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \boldsymbol{\tau} \\ \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \boldsymbol{\tau}_g(\boldsymbol{q}) + \boldsymbol{\tau}_c(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_o(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{aligned} \tag{5.9}$$

where $\boldsymbol{M} \in \Re^{n \times n}$ denotes the symmetric, positive definite joint space inertia mass matrix, n the number of model DoFs and $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \Re^n$ the joint space generalized

coordinates and their derivatives with respect to time. The term $\boldsymbol{f} \in \Re^n$ represents the sum of all joint space forces, $\boldsymbol{\tau}_g \in \Re^n$ the gravity, $\boldsymbol{\tau}_c \in \Re^n$ the Coriolis and centrifugal, $\boldsymbol{\tau}_o \in \Re^n$ other generalized forces, whereas $\boldsymbol{\tau} \in \Re^n$ the vector of applied generalized forces that actuate the model. Most of the quantities in the equations are a function of the generalized coordinates and derivatives. This dependency will be omitted for simplicity.

We can project Equation 5.9 onto the task space by multiplying both sides from the left with $\boldsymbol{J}_t\boldsymbol{M}^{-1}$ and using Equations 5.4 and 5.7

$$\begin{aligned} \boldsymbol{J}_t\boldsymbol{M}^{-1}\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{J}_t\boldsymbol{M}^{-1}\boldsymbol{f} &= \boldsymbol{J}_t\boldsymbol{M}^{-1}\boldsymbol{\tau} \\ \ddot{\boldsymbol{x}}_t - \dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}} + \boldsymbol{J}_t\boldsymbol{M}^{-1}\boldsymbol{f} &= \boldsymbol{J}_t\boldsymbol{M}^{-1}(\boldsymbol{J}_t^T\boldsymbol{f}_t + \boldsymbol{N}_{\boldsymbol{F}_{\boldsymbol{J}_t}}\boldsymbol{\tau}_0) \\ \boldsymbol{\Lambda}_t(\ddot{\boldsymbol{x}}_t + \boldsymbol{b}_t) + \bar{\boldsymbol{J}}_t^T\boldsymbol{f} &= \boldsymbol{f}_t + \bar{\boldsymbol{J}}_t^T\boldsymbol{N}_{\boldsymbol{F}_{\boldsymbol{J}_t}}\boldsymbol{\tau}_0 \end{aligned} \tag{5.10}$$

where $\boldsymbol{\Lambda}_t = (\boldsymbol{J}_t\boldsymbol{M}^{-1}\boldsymbol{J}_t^T)^{-1} \in \Re^{d\times d}$ denotes the task space inertia mass matrix, $\boldsymbol{b}_t = -\dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}}$ the task bias term and $\bar{\boldsymbol{J}}_t^T = \boldsymbol{\Lambda}_t\boldsymbol{J}_t\boldsymbol{M}^{-1} \in \Re^{d\times n}$ the generalized inverse transpose of $\boldsymbol{J}_t$ that is used to project joint space quantities in the task space.

At this point it is worth commenting on the different types of null space operators. If the null space is derived using the MPP (as in Equation 5.7), then the projection operator $\boldsymbol{N}_{\boldsymbol{F}_{\boldsymbol{J}_t}}$ does not ensure fully decoupled control in both task and null space. It has been shown that from the many projection operators that map on the null space (Righetti et al., 2011), there exists a unique generalized inverse ($\bar{\boldsymbol{J}}_t^T$) that ensures this decoupling (Khatib, 1995), which is a useful property in the presence of multiple prioritized tasks (Sentis, 2007). More specifically, suppose that

$$\boldsymbol{\tau} = \boldsymbol{J}_t^T\boldsymbol{f}_t + (\boldsymbol{I} - \boldsymbol{J}_t^T\bar{\boldsymbol{J}}_t^T)\boldsymbol{\tau}_0 \tag{5.11}$$

then we seek to find the generalized inverse that ensures zero contribution of the null space forces to the acceleration of a task

$$\begin{aligned} \boldsymbol{\Lambda}_t(\ddot{\boldsymbol{x}}_t + \boldsymbol{b}_t) + \bar{\boldsymbol{J}}_t^T\boldsymbol{f} &= \boldsymbol{f}_t \\ \boldsymbol{J}_t\boldsymbol{M}^{-1}(\boldsymbol{I} - \boldsymbol{J}_t^T\bar{\boldsymbol{J}}_t^T)\boldsymbol{\tau}_0 &\stackrel{!}{=} \boldsymbol{0},\ \forall\boldsymbol{\tau}_0 \end{aligned} \tag{5.12}$$

where the last requirement ($\stackrel{!}{=}$) implies that

$$\bar{\boldsymbol{J}}_t^T = \boldsymbol{\Lambda}_t\boldsymbol{J}_t\boldsymbol{M}^{-1}. \tag{5.13}$$

Remark 5.4. *[Task Space Generalized Inverse]*
The null space projection operator $\boldsymbol{N}_{\boldsymbol{F}_{\boldsymbol{J}_t}} = \boldsymbol{I} - \boldsymbol{J}_t^+\boldsymbol{J}_t$ cannot decouple task and null space forces, owing to the uniqueness of the generalized inverse Jacobian that achieves this decoupling.

Proof. (Khatib, 1995) □

5.2.3 Relation Between Muscle and Joint Space Quantities

Interestingly, we can follow a similar approach by establishing the corresponding relations between the muscle and joint space quantities in order to address the dynamic redundancy problem. The musculotendon length is given as a function of the generalized coordinates ($\boldsymbol{q}$) and depends on the muscle routing parameters (ϕ)

$$\boldsymbol{l}_m = \boldsymbol{f}(\boldsymbol{q}; \phi),\ \boldsymbol{l}_m \in \Re^m,\ \boldsymbol{q} \in \Re^n,\ n < m \tag{5.14}$$

implying that the generalized coordinates fully define the musculotendon lengths, assuming that the muscles are pretensioned. The derivative of Equation 5.14 with respect to time is given by

$$\dot{\boldsymbol{l}}_m = \boldsymbol{R}(\boldsymbol{q})\dot{\boldsymbol{q}},\ \boldsymbol{R}(\boldsymbol{q}) = \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial q_1} & \cdots & \frac{\partial \boldsymbol{f}_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{f}_m}{\partial q_1} & \cdots & \frac{\partial \boldsymbol{f}_m}{\partial q_n} \end{bmatrix} \tag{5.15}$$

where the muscle moment arm ($\boldsymbol{R} : \Re^n \rightarrow \Re^m$) is always a tall matrix (Proposition A.5).

Remark 5.5. *[Moment Arm Matrix (tall)]*

(a) If rank($\boldsymbol{R}$) = n the homogeneous system $\boldsymbol{R}\dot{\boldsymbol{q}} = \boldsymbol{0}$ has a unique, trivial solution.

(b) If rank($\boldsymbol{R}$) < n the homogeneous system $\boldsymbol{R}\dot{\boldsymbol{q}} = \boldsymbol{0}$ has infinite solutions.

(c) The system $\boldsymbol{R}\dot{\boldsymbol{q}} = \dot{\boldsymbol{l}}_m$ is inconsistent for some $\dot{\boldsymbol{l}}_m \in \Re^m$ (a solution exists iff $\dot{\boldsymbol{l}}_m \in \mathbb{C}(\boldsymbol{R})$).

(d) The MPP matrix of $\boldsymbol{R}$ is the left inverse ($\boldsymbol{R}^+\boldsymbol{R} = \boldsymbol{I}$).

Due to the over-availability of muscles, Equation 5.15 can be reformulated to account for any change in $\dot{\boldsymbol{l}}_m$ that does not contribute to a change in $\dot{\boldsymbol{q}}$

$$\dot{\boldsymbol{l}}_m = \dot{\boldsymbol{l}}_m^{\parallel} + \dot{\boldsymbol{l}}_m^{\perp} = \boldsymbol{R}\dot{\boldsymbol{q}} + \boldsymbol{N}_{\boldsymbol{T}_{\boldsymbol{R}}}\dot{\boldsymbol{l}}_{m0},\ \boldsymbol{N}_{\boldsymbol{T}_{\boldsymbol{R}}} = (\boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^+) \tag{5.16}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{R}^+\dot{\boldsymbol{l}}_m \tag{5.17}$$

where $\boldsymbol{N}_{\boldsymbol{T}_{\boldsymbol{R}}} \in \Re^{m \times m}$ represents the null space of $\boldsymbol{R}$ (left null space Definition 2.1), $\dot{\boldsymbol{l}}_{m0}$ an arbitrarily selected vector in $\Re^m$ and Equation 5.17 the inverse mapping of Equation 5.16. The derivative of Equations 5.16 and 5.17 with respect to time are given by

$$\ddot{\boldsymbol{l}}_m = \dot{\boldsymbol{R}}\dot{\boldsymbol{q}} + \boldsymbol{R}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{N}}_{\boldsymbol{T}_{\boldsymbol{R}}}\dot{\boldsymbol{l}}_{m0} + \boldsymbol{N}_{\boldsymbol{T}_{\boldsymbol{R}}}\ddot{\boldsymbol{l}}_{m0} \tag{5.18}$$

$$\ddot{\boldsymbol{q}} = \dot{\boldsymbol{R}}^+\dot{\boldsymbol{l}}_m + \boldsymbol{R}^+\ddot{\boldsymbol{l}}_m \tag{5.19}$$

with $\ddot{\boldsymbol{l}}_{m0}$ being an arbitrarily selected vector in $\Re^m$. The null space contribution indicates that the CNS may regulate the lengthening/shortening of the muscles without altering the corresponding joint space quantities. This property can be used in coordinating different aspects of the tendon driven limbs or in the modeling of several pathological conditions of the muscle coordination process (e.g., slack muscles).

From the virtual work principle (as in Equation 5.6) we can establish a relationship between joint and muscle space forces

$$\begin{aligned} \boldsymbol{\tau}^T \delta \boldsymbol{q} &= -\boldsymbol{f}_m^T \delta \boldsymbol{l}_m \\ \boldsymbol{\tau}^T \delta \boldsymbol{q} &= -\boldsymbol{f}_m^T \boldsymbol{R} \delta \boldsymbol{q} \\ \boldsymbol{\tau} &= -\boldsymbol{R}^T \boldsymbol{f}_m \end{aligned} \tag{5.20}$$

where the mapping ($\boldsymbol{R}^T : \Re^m \rightarrow \Re^n$) is a fat matrix (Proposition A.4). The negative sign convention is introduced, admitting that a muscle induces positive work during shortening ($\dot{\boldsymbol{l}}_m < 0$).

Remark 5.6. *[Moment Arm Transpose (fat)]*

(a) The system $\boldsymbol{R}^T \boldsymbol{f}_m = \boldsymbol{0}$ has infinite solutions.

(b) If rank$(\boldsymbol{R}^T) = n$ then the system $\boldsymbol{R}^T \boldsymbol{f}_m = \boldsymbol{\tau}$ is consistent (one or more solutions).

(c) If rank$(\boldsymbol{R}^T) < n$ then the system $\boldsymbol{R}^T \boldsymbol{f}_m = \boldsymbol{\tau}$ is inconsistent, e.g., $\exists \boldsymbol{\tau}$: $\boldsymbol{R}^T \boldsymbol{f}_m \neq \boldsymbol{\tau} \Rightarrow \boldsymbol{\tau} \notin \mathbb{C}(\boldsymbol{R}^T)$.

(d) The MPP matrix of $\boldsymbol{R}^T$ is the right inverse ($\boldsymbol{R}^T \boldsymbol{R}^{+T} = \boldsymbol{I}$).

The muscle forces $\boldsymbol{f}_m$ can be written as the sum of the two mutually orthogonal vectors $\boldsymbol{f}_m^{\parallel} \in \mathbb{C}(\boldsymbol{R}^{+T})$ and $\boldsymbol{f}_m^{\perp} \in \mathbb{N}(\boldsymbol{R}^{+})$ (Equation 2.7). In contrast to Equation 5.20, the direct mapping does not require any reformulation, but the inverse requires reformulation in order to account for the redundancy effect

$$\boldsymbol{f}_m = -\boldsymbol{R}^{+T} \boldsymbol{\tau} + (\boldsymbol{I} - \underbrace{(\boldsymbol{R}\boldsymbol{R}^{+})^T}_{\boldsymbol{R}\boldsymbol{R}^{+}}) \boldsymbol{f}_{m0} = -\boldsymbol{R}^{+T} \boldsymbol{\tau} + \boldsymbol{N}_{\boldsymbol{T}_{\boldsymbol{R}}} \boldsymbol{f}_{m0} \tag{5.21}$$

where $\boldsymbol{N}_{\boldsymbol{T}_{\boldsymbol{R}}}$ is the same matrix defined in Equation 5.16, revealing the dual relationship between the subspaces of $\boldsymbol{R}$ and $\boldsymbol{R}^{+T}$ (Remark 5.7). Note, however, that this definition spans the entire $\Re^m$, whereas in reality muscle forces are strictly positive (contraction) and bounded (limited force), which makes the above definition physiologically ill conditioned. In subsection 5.2.5, we will provide an effective solution to this problem. If a solution satisfying the required torque exists, then there are infinitely many combinations of muscle forces ($\boldsymbol{N}_{\boldsymbol{T}_{\boldsymbol{R}}} \boldsymbol{f}_{m0}$) that do not alter the overall required torque. On a physiological level, this strategy is explored by the CNS in order to regulate joint stiffness through co-contraction (Inouye and Valero-Cuevas, 2016).

Remark 5.7. *[Moment Arm Motion-Force Duality]*
The projection operators of $\boldsymbol{R}$ defines the same subspaces as the projection operators of $\boldsymbol{R}^{+T}$.

Proof. Similar to Remark 5.3. □

To summarize, it is of particular interest, whether pathological conditions can be encoded in the choice of $\dot{\boldsymbol{l}}_{m0}$, $\ddot{\boldsymbol{l}}_{m0}$ and $\boldsymbol{f}_{m0}$. Examples include the investigation of the cause-effect of slack muscle tone (e.g., hypotonia or hypertonia) or the increase in co-contraction causing rigidity. The incorporation of the null space complements the original equations and unveils the structure of dynamic redundancy. Additional steps have to be performed to ensure that Equation 5.21 provides physiologically correct interpretations.

5.2.4 Muscle Space Equations of Motion

In the previous section, the various relations between motion and force quantities of the dynamically redundant system were established. In this section we will derive the EoMs that relate the muscle lengthening/shortening accelerations to muscle space forces.

As was previously shown (Equation 5.21), the muscle forces $\boldsymbol{f}_m$ can be expressed as the sum of two mutually orthogonal subspaces that span $\Re^m$ ($\boldsymbol{f}_m = \boldsymbol{f}_m^{\parallel} + \boldsymbol{f}_m^{\perp}$). We can project Equation 5.9 onto the muscle space by multiplying both sides from the left with $-\boldsymbol{R}\boldsymbol{M}^{-1}$, also taking Equations 5.18 and 5.20 into account

$$\begin{aligned} -\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{M}\ddot{\boldsymbol{q}} - \boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{f} &= -\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{\tau} \\ -(\ddot{\boldsymbol{l}}_m - \dot{\boldsymbol{R}}\dot{\boldsymbol{q}} - \dot{\boldsymbol{N}}_{\boldsymbol{T_R}}\dot{\boldsymbol{l}}_{m0} - \boldsymbol{N}_{\boldsymbol{T_R}}\ddot{\boldsymbol{l}}_{m0}) - \boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{f} &= \boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T\boldsymbol{f}_m \\ -\boldsymbol{\Lambda}_m(\ddot{\boldsymbol{l}}_m + \boldsymbol{b}_m) - \boldsymbol{R}^{+T}\boldsymbol{f} &= \boldsymbol{f}_m^{\parallel} + \boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0} \end{aligned} \tag{5.22}$$

where $\boldsymbol{\Lambda}_m = (\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T)^+ \in \Re^{m\times m}$ represents the muscle space inertia mass matrix, $\boldsymbol{b}_m = -\dot{\boldsymbol{R}}\dot{\boldsymbol{q}} - \dot{\boldsymbol{N}}_{\boldsymbol{T_R}}\dot{\boldsymbol{l}}_{m0}$ the muscle bias term and $\boldsymbol{\Lambda}_m\boldsymbol{R}\boldsymbol{M}^{-1} = \boldsymbol{R}^{+T} \in \Re^{m\times n}$ is used to project joint space quantities onto the muscle space. Note that the null space term $\boldsymbol{N}_{\boldsymbol{T_R}}\ddot{\boldsymbol{l}}_{m0}$ cancel out when multiplied by $\boldsymbol{\Lambda}_m$ (Proposition 5.1). When Equation 5.22 is solved in an ID manner, the quantities of interest are the muscle forces $\boldsymbol{f}_m$, while $\dot{\boldsymbol{l}}_{m0}$, $\boldsymbol{f}_{m0}$ are free variables which are selected so as to satisfy additional modeling criteria.

Proposition 5.1. *[Muscle Space Inertia Mass Matrix]*
Given that $\boldsymbol{T_R} = \boldsymbol{R}\boldsymbol{R}^+$ and $\boldsymbol{\Lambda}_m = (\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T)^+$ the following assertions hold:

(a) *$\boldsymbol{\Lambda}_m$ is symmetric ($\boldsymbol{\Lambda}_m = \boldsymbol{\Lambda}_m^T$) iff $\boldsymbol{M}$ is symmetric.*

(b) *$\boldsymbol{\Lambda}_m$ remains invariant under the projection operator $\boldsymbol{T_R}$.*

(c) *$\boldsymbol{\Lambda}_m\boldsymbol{\Lambda}_m^+ = \boldsymbol{T_R}$.*

(d) $\boldsymbol{\Lambda}_m \boldsymbol{N}_{\boldsymbol{T_R}} = \mathbf{0}$.

Proof.

(a) $\boldsymbol{\Lambda}_m$ is symmetric ($\boldsymbol{\Lambda}_m = \boldsymbol{\Lambda}_m^T$) iff $\boldsymbol{M}$ is symmetric.

$$\boldsymbol{\Lambda}_m^T = (\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T)^{+T} = (\boldsymbol{R}\boldsymbol{M}^{-T}\boldsymbol{R}^T)^{+} = (\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T)^{+} = \boldsymbol{\Lambda}_m, \; \textit{iff } \boldsymbol{M} = \boldsymbol{M}^T$$

(b) $\boldsymbol{\Lambda}_m$ remains invariant under the projection operator $\boldsymbol{T}_{\boldsymbol{R}}$.

$$\boldsymbol{\Lambda}_m \boldsymbol{T}_{\boldsymbol{R}} = (\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T)^{+}\boldsymbol{R}\boldsymbol{R}^{+} = \boldsymbol{R}^{+T}\boldsymbol{M}\underbrace{\boldsymbol{R}^{+}\boldsymbol{R}\boldsymbol{R}^{+}}_{\boldsymbol{R}^{+}} = (\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T)^{+} = \boldsymbol{\Lambda}_m$$

(c) $\boldsymbol{\Lambda}_m \boldsymbol{\Lambda}_m^{+} = \boldsymbol{T}_{\boldsymbol{R}}$.

$$\begin{aligned}\boldsymbol{\Lambda}_m \boldsymbol{\Lambda}_m^{+} &= (\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T)^{+}(\boldsymbol{R}\boldsymbol{M}^{-1}\boldsymbol{R}^T) = \boldsymbol{R}^{+T}\boldsymbol{M}\underbrace{\boldsymbol{R}^{+}\boldsymbol{R}}_{\boldsymbol{I}}\boldsymbol{M}^{-1}\boldsymbol{R}^T \\ &= \boldsymbol{R}^{+T}\boldsymbol{R}^T = (\boldsymbol{R}\boldsymbol{R}^{+})^T = \boldsymbol{R}\boldsymbol{R}^{+} = \boldsymbol{T}_{\boldsymbol{R}}\end{aligned}$$

(d) $\boldsymbol{\Lambda}_m \boldsymbol{N}_{\boldsymbol{T_R}} = \mathbf{0}$. This can be shown using (b).

□

The introduction of the null space term ($\boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0}$) complements the relationship between muscle and joint space forces (Equation 5.21) and the muscle space EoMs (Equation 5.22). Without this term, the relations are ill conditioned due to physiological restrictions of the muscles ($\mathbf{0} \preceq \boldsymbol{f}_m \preceq \boldsymbol{f}_{\max}$). Although a particular solution $\boldsymbol{f}_m$ or $\boldsymbol{f}_m^{\parallel}$ for some arbitrary $\boldsymbol{\tau}$ or $\ddot{\boldsymbol{\imath}}_m$, respectively does not necessarily satisfy these restrictions, a suitable $\boldsymbol{f}_{m0}$ can be selected. Furthermore, several aspects and patterns of muscle co-contractions can be modeled by appropriately selecting $\boldsymbol{f}_{m0}$.

5.2.5 Exploitation of Kinematic and Dynamic Redundancy

In the previous section, we derived the EoMs in muscle space. Furthermore, we decomposed the muscle forces ($\boldsymbol{f}_m$) into two mutually orthogonal subspaces that span the entire muscle space ($\Re^m$). Here we will define an effective approach for taking advantage of the null space contribution in order to satisfy the physiological restrictions of the muscles.

Muscle redundancy manifests itself during ID, whereas the null space term does not contribute to the movement in a FD setting, because $\boldsymbol{R}^T\boldsymbol{N}_{\boldsymbol{T}_R}\boldsymbol{f}_{m0} = \mathbf{0}$. For a given action a particular solution of muscle forces ($\boldsymbol{f}_m^{\parallel}$) can be found by initially

ignoring the null space contribution ($\boldsymbol{f}_m^{\perp}$), which is not guaranteed to be physiologically correct (namely $\exists i : f_m^i > f_{\max}^i$ or $f_m^i < 0$). However, the null space can provide a suitable correction in order to satisfy the physiological constraints. A solution of muscle forces $\boldsymbol{f}_m^{\parallel}$ in the task, joint or muscle space respectively can be obtained as follows (note that the MPP solves the least squares problem Definition A.2)

$$\begin{aligned} \boldsymbol{f}_m^{\parallel} &= -\boldsymbol{\Theta}^T((\boldsymbol{\Lambda}_t(\ddot{\boldsymbol{x}}_t + \boldsymbol{b}_t) + \bar{\boldsymbol{J}}_t^T \boldsymbol{f}),\ \boldsymbol{\Theta} = \boldsymbol{J}_t \boldsymbol{R}^+ \in \Re^{d \times m} \\ \boldsymbol{f}_m^{\parallel} &= -\boldsymbol{R}^{+T}(\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{f}) \\ \boldsymbol{f}_m^{\parallel} &= -\boldsymbol{\Lambda}_m(\ddot{\boldsymbol{l}}_m + \boldsymbol{b}_m) - \boldsymbol{R}^{+T}\boldsymbol{f}. \end{aligned} \tag{5.23}$$

It is well-known that muscle dynamics are described by a complex nonlinear system (Zajac, 1989; Thelen, 2003; Millard et al., 2013) that depends on a multitude of factors, such as the strength of the muscles, the muscle activation dynamics, the pennation angle, the elastic properties of the muscle fibers and tendon, the force-length and the force-velocity characteristics, etc. Consequently, these factors will also affect the null space solution component $\boldsymbol{f}_{m0}$. Assuming the simplest possible muscle model

$$\boldsymbol{f}_m = \boldsymbol{f}_{\max} \circ \boldsymbol{a}_m,\ \boldsymbol{0} \preceq \boldsymbol{a}_m \preceq \boldsymbol{1} \tag{5.24}$$

where $\boldsymbol{a}_m \in \Re^m$ represents a vector of muscle activations, $\boldsymbol{f}_{\max} \in \Re^m$ a vector specifying the maximum muscle forces and $\circ$ the Hadamard (elementwise) product. We can impose bounds on the possible solutions of $\boldsymbol{f}_{m0}$ in the form of linear inequalities, by noting that Equations 5.21 and 5.24 must be equal

$$\begin{aligned} \boldsymbol{f}_m = \boldsymbol{f}_{\max} \circ \boldsymbol{a}_m = \boldsymbol{f}_m^{\parallel} + \boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0},\ \boldsymbol{0} \preceq \boldsymbol{a}_m \preceq \boldsymbol{1} \rightarrow \\ \begin{bmatrix} -\boldsymbol{N}_{\boldsymbol{T_R}} \\ \hdashline \boldsymbol{N}_{\boldsymbol{T_R}} \end{bmatrix} \boldsymbol{f}_{m0} \preceq \begin{bmatrix} \boldsymbol{f}_m^{\parallel} \\ \hdashline \boldsymbol{f}_{\max} - \boldsymbol{f}_m^{\parallel} \end{bmatrix}. \end{aligned} \tag{5.25}$$

Given an appropriately selected cost function, a unique solution of the linear inequalities can be obtained by solving the constrained optimization problem. A generalization for a nonlinear muscle model (Zajac, 1989) is presented below

$$\begin{aligned} \boldsymbol{f}_m = \boldsymbol{f}_{\max} \circ \left(\boldsymbol{a}_m \circ \boldsymbol{f}_l \circ \boldsymbol{f}_v + \boldsymbol{f}_{pe}\right) = \boldsymbol{f}_m^{\parallel} + \boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0},\ \boldsymbol{0} \preceq \boldsymbol{a}_m \preceq \boldsymbol{1} \rightarrow \\ \begin{bmatrix} -\boldsymbol{N}_{\boldsymbol{T_R}} \\ \hdashline \boldsymbol{N}_{\boldsymbol{T_R}} \end{bmatrix} \boldsymbol{f}_{m0} \preceq \begin{bmatrix} \boldsymbol{f}_m^{\parallel} - \boldsymbol{f}_{\max} \circ \boldsymbol{f}_{pe} \\ \hdashline \boldsymbol{f}_{\max} \circ \left(\boldsymbol{f}_l \circ \boldsymbol{f}_v + \boldsymbol{f}_{pe}\right) - \boldsymbol{f}_m^{\parallel} \end{bmatrix} \end{aligned} \tag{5.26}$$

where $\boldsymbol{f}_l,\ \boldsymbol{f}_v,\ \boldsymbol{f}_{pe} \in \Re^m$ are the force-length, force-velocity and passive muscle forces.

It is of great interest to understand the structural dependencies of these inequalities in order to outline the feasible muscle force space. For a muscle model $\mathcal{M}$ and an arbitrary action $\mathcal{A}$, the problem is expressed in its most general form (generalization of Equation 5.26):

$$\boldsymbol{Z}(\boldsymbol{N}_{\boldsymbol{T_R}})\boldsymbol{f}_{m0} \preceq \boldsymbol{\beta}(\mathcal{M},\ \mathcal{A}) \tag{5.27}$$

where the solutions ($\boldsymbol{f}_{m0}$) lie in a convex subspace of $\Re^r$ ($r \leq m$) and the feasible space is bounded (Proposition 5.2). Note that the action ($\mathcal{A}$) implicitly encodes the kinematic redundancy (Equation 5.23). Finally, the feasible muscle force set is computed as the sum of the particular solution and the null space forces

$$\boldsymbol{f}_m^{\oplus} = \{\boldsymbol{f}_m^{\parallel} + \boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0}^{i},\ \forall i\}. \tag{5.28}$$

Proposition 5.2. *[Feasible Null Space Forces]*
The following properties hold true:

(a) Equation 5.27 defines a convex set $\mathbb{C}$.

(b) Equation 5.27 is bounded if the inequality is feasible and if $\boldsymbol{Z}$ *is full column rank.*

Proof.

(a) Let's prove that Equation 5.27 defines a convex set $\mathbb{C}$. For any $\boldsymbol{x},\ \boldsymbol{y} \in \mathbb{C}$ and $\lambda \in [0,\ 1]$, let $\boldsymbol{\psi} = \lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}$ then

$$\boldsymbol{Z}\boldsymbol{\psi} = \lambda\boldsymbol{Z}\boldsymbol{x} + (1-\lambda)\boldsymbol{Z}\boldsymbol{y} \preceq \lambda\boldsymbol{\beta} + (1-\lambda)\boldsymbol{\beta} = \boldsymbol{\beta}$$

which proves the convexity property.

(b) Let us prove that Equation 5.27 is bounded when the inequality is feasible and $\boldsymbol{Z}$ has full column rank. From the definition, this inequality has always the following form

$$\boldsymbol{a} \preceq \boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0} \preceq \boldsymbol{b} \tag{5.29}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are finite vectors in $\Re^m$. If the system of inequalities is feasible and $\boldsymbol{N}_{\boldsymbol{T_R}}$ has full column rank, then $\boldsymbol{f}_{m0}$ is upper and lower bounded. Even if the problem is feasible, if $\boldsymbol{N}_{\boldsymbol{T_R}}$ is rank deficient, then $\exists i : f_{m0}^{i} \rightarrow \pm\infty$, thus the feasible space is not bounded. Note, however, that we can always reduce the null space projection to full column rank by eliminating linearly dependent columns in $\boldsymbol{N}_{\boldsymbol{T_R}}$.

□

Traditional solutions of the muscle redundancy problem (Anderson and Pandy, 2001) (e.g., effort minimization) ignore the possibilities of the system and prove to be inadequate in quantifying the decisive factors, whereas Equations 5.27 and 5.28

have an elegant form that outlines the entire set of possible solutions. More importantly, each solution can be characterized (e.g., minimum effort, high stiffness, rigidity, etc.), enabling classification of the different manifolds (strategies) available to the CNS.

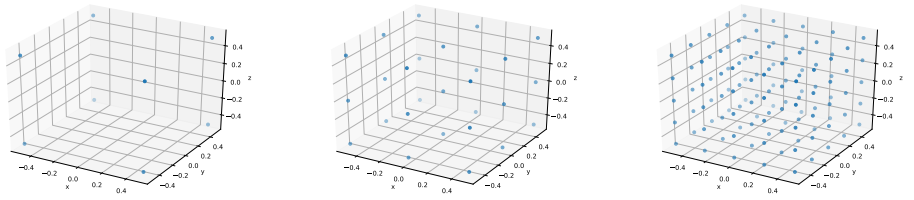
In order to compute the feasible muscle force set $\boldsymbol{f}_m^{\oplus}$, the physiological null space muscle forces $\boldsymbol{f}_{m0}$ must be obtained by sampling the space defined by a set of linear inequalities (Equation 5.27). The linear inequalities define a polytope (a convex polyhedron) as an intersection of a finite number of half-spaces (hyperplane- or H-representation). According to Proposition 5.2, this polytope has a finite number of extreme points (vertex- or V-representation), which we use to sample this polytope sufficiently. The conversion from H-representation to V-representation is called vertex enumeration and can be achieved by using either a deterministic (Avis and Fukuda, 1992) or randomized (Vempala, 2005) approach. From an arrangement of n hyperplanes in $\Re^d$, v vertices are determined in $O(n^2dv)$ time. In our problem $n = 2m,\ d \leq m$ (m being number of muscles), thus the time complexity is $O(vm^3)$. We utilize the `lrs` library (Avis and Fukuda, 1992), which provides a self-contained ANSI C implementation of the reverse search algorithm for vertex enumeration. After obtaining the extreme points of the polytope, additional solutions are generated by interpolating ($\psi = \lambda\boldsymbol{x}+(1-\lambda)\boldsymbol{y}$) between vertices. This process generates samples spanning the entire polytope due to its convexity, as proved earlier. An example of Algorithm 5.1 is presented (Figure 5.1) for the following system

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{A} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \boldsymbol{b} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}. \tag{5.30}$$

Algorithm 5.1 Iterative sampling of a convex bounded polyhedron.

Input: H-representation $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$, d: iteration depth
Output: A representative set $\mathcal{S} = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$

Find the V-representation $\mathcal{V}$ using lrs (or some other method)
for $i = 0$ **to** d **do**
  $\mathcal{T} = \mathcal{V}$
  for all distinct $\{\boldsymbol{x},\ \boldsymbol{y}\} \in \mathcal{T}$ **do**
    Add $\psi = \lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}$ to $\mathcal{V}$ $-\lambda = 0.5''$
  end for
end for
return $\mathcal{V}$



(a) $d = 0$ **(b)** $d = 1$ **(c)** $d = 2$

Figure 5.1: The inequality Equation 5.30 is sampled using Algorithm 5.1 for different levels of iteration depth (d).

5.3 Results

We proceed by demonstrating working examples of the developed models. The subsection 5.3.1 presents the utilization of muscle space projection for solving the posture control problem, where the muscle length and derivatives are the regulating quantities. This experiment highlights the importance of coordinate projection, which essentially transforms the EoMs onto some space (e.g., task or muscle) to enable direct control of the quantity of interest. Subsequently, subsection 5.3.2 continues by evaluating the feasible muscle forces in the context of an action and identifying force variabilities and muscle group correlations. We highlight the main contribution, which is the calculation of the feasible muscle forces that satisfy both the task and the physiological constraints of the muscles. The two experiments follow a MD scheme, where the FD method is used for simulating the model combined with an ID model-based controller (muscle and task space, respectively), representing the underlying system. Last but not least, a joint reaction simulation is performed on a gait movement, appropriately accounting for the null space muscle forces that do not alter the movement, demonstrating the importance of estimating the null space contribution and its impact on the joint reaction loads.

A simplified arm model has been developed in order to demonstrate the utilization of the proposed methods. It has three DoFs and nine muscles (Figure A.1), appropriately constructed to demonstrate both kinematic and dynamic redundancy (e.g., $d < n < m$). The EoMs and the geometric parameters of the model are derived analytically in order to evaluate higher order derivatives of moment arm and task Jacobian matrices (section A.4). The model is able to capture the redundancy effects conveying the importance of this framework and is also an excellent tool for introducing these methods. An anatomically realistic full body gait model (Delp et al., 1990) is used for the joint reaction analysis (`OpenSim` (Delp et al., 2007)), illustrating that the presented methods are universal and can be applied in more practical and realistic scenarios. The model, which has ten DoFs and eighteen muscles (Millard muscle model (Millard et al., 2013)), features lower extremity joint definitions, low back joint and a planar knee model.

5.3.1 Muscle Space Projection and Reflexes

Muscle space projection exhibits several advantages in problems related to the segmental level coordination, mainly because of the different muscle related variables (e.g., length, velocity, muscle stiffness) that are directly/indirectly controlled by the CNS (Stein, 1982; Houk and Rymer, 2011). For example, the proprioceptive receptors, located in the spinal cord, constantly measure the evolution of these variables and issue corrective actions (Loeb et al., 1990). More specifically, the muscle spindle organs (Mileusnic et al., 2006) measure changes in the muscle length and the Golgi tendon organs (Mileusnic and Loeb, 2006) measure the force exerted by their respective muscles. This indicates that the aforementioned variables are used in an internal feedback regulation process (Zhang et al., 2009).

We continue with the design of a posture controller that coarsely encapsulates the characteristics of the internal regulation process performed in the spinal cord and the evaluation of system's response to external disturbances. As the muscle length and its derivative are the regulating variables, muscle space projection is a natural representation in this setting. The following control scheme is adopted for this experiment

$$\ddot{\boldsymbol{l}}_m(t) = k_p(\boldsymbol{l}_m^d - \boldsymbol{l}_m(t - \tau_{so})) - k_d\dot{\boldsymbol{l}}_m(t - \tau_{so}) \tag{5.31}$$

where $\ddot{\boldsymbol{l}}_m(t)$ represents the muscle length acceleration goals, $\boldsymbol{l}_m^d = \boldsymbol{l}_m(t = 0)$ the desired muscle length positions, $\boldsymbol{l}_m(t - \tau_{so})$ and $\dot{\boldsymbol{l}}_m(t - \tau_{so})$ the perceived, delayed ($\tau_{so} = 20$ ms originating from muscle spindle organs (Zhang et al., 2009)) muscle length positions and velocities and k_p, k_d the reflex gains. Note that this control law restores the system to its original posture $\boldsymbol{l}_m(t = 0)$. A force disturbance impulse in an arbitrary direction is applied on the end effector body. The impulse is modeled by a Gaussian function

$$f(t) = ae^{\frac{-(t-t_0)}{2\sigma^2}} \tag{5.32}$$

where a controls the magnitude, t_0 the application time and σ the smoothness of the impulse. For the following experiments we will use $a = 15$, $t_0 = 0.1$, $\sigma = 0.01$ and the perturbation will act in the $-\boldsymbol{x}$ direction. The muscle space EoMs Equation 5.22 can serve as an ID model-based controller for calculating the required muscle forces $\boldsymbol{f}_m$ (Algorithm 5.2), which may then be fed to the FD module. The $\dot{\boldsymbol{l}}_{m0}$ in Algorithm 5.2 can be arbitrarily selected by the modeler to encode various strategies available by the CNS. In our experiments, $\dot{\boldsymbol{l}}_{m0}$ is set zero to encode an unbiased strategy, for the sake of simplicity.

Figure 5.2 depicts collectively the simulation results when the system is perturbed and the two reflex loops are active ($k_p = 10$, $k_d = 10$ and $\tau_{so} = 20$ ms). The impact is absorbed and the system returns to its initial configurations (stable response). Conversely, in Figure 5.3 the "descending" command is disabled ($k_p = 0$), leaving the response of the reflex loop alone, in which the system reaches a new, stable equilibrium. The muscle space projection suits this kind of studies

Algorithm 5.2 Muscle space force computation for a given goal using muscle space projection.

Input: muscle length accelerations $\ddot{\boldsymbol{l}}_m$, null space muscle length velocities $\dot{\boldsymbol{l}}_{m0}$
Output: muscle forces $\boldsymbol{f}_m$
1: Solve Equation 5.22 for $\boldsymbol{f}_m^{\parallel}$ by initially ignoring the null space term $\boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0}$
2: Minimize $\|\boldsymbol{f}_{m0}\|^2$ for $\boldsymbol{f}_{m0}$ subject to Equation 5.27
3: Obtain the required muscle forces as $\boldsymbol{f}_m = \boldsymbol{f}_m^{\parallel} + \boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0}$
4: **return** $\boldsymbol{f}_m$

well as it offers a direct interface with segmental level models. On the contrary, additional transformation of the control variables is required if the problem were expressed in joint space. Different combinations of reflex loop gains, delays and external factors can be evaluated quantitatively, demonstrating the usefulness of this projection.

5.3.2 Characterization of the Feasible Muscle Force Space

In this experiment, the feasible muscle forces that satisfy the task and the physiological muscle constraints are computed. The analysis is restricted on a particular time instance, where we identify the muscle force bounds and correlations between different muscle pairs. The movement of the hand is planned in task space using Equations 5.11 and 5.12 as an ID model-based controller in combination with a PD tracking scheme

$$\ddot{\boldsymbol{x}}_t = \ddot{\boldsymbol{x}}_d + k_p(\boldsymbol{x}_d - \boldsymbol{x}_t) + k_d(\dot{\boldsymbol{x}}_d - \boldsymbol{x}_t) \tag{5.33}$$

where $\boldsymbol{x}_d$, $\dot{\boldsymbol{x}}_d$ and $\ddot{\boldsymbol{x}}_d$ denote the desired position, velocity and acceleration of the end effector and $k_p = 50$, $k_d = 5$ the tracking gains. The planning problem can be encoded naturally in task space as compared to joint and muscle space controllers. Note that this analysis is independent of the underlying ID representation (Equation 5.23). Furthermore, a linear muscle model (Equation 5.25) is assumed.

The desired task goal is derived from a smooth sigmoid function that produces bell-shaped velocity profiles in any direction around the initial position of the end effector

$$\begin{aligned} \boldsymbol{x}_d(t) &= \left[x_t(0) + \frac{a}{2}(\tanh(b(t-t_0)) + 1),\ y_t(0)\right]^T \\ \dot{\boldsymbol{x}}_d(t) &= \frac{d\boldsymbol{x}_d(t)}{dt},\ \ddot{\boldsymbol{x}}_d(t) = \frac{d\dot{\boldsymbol{x}}_d(t)}{dt} \\ \boldsymbol{x}_d^{'} &= \boldsymbol{H}_z(\gamma)\boldsymbol{x}_d,\ \dot{\boldsymbol{x}}_d^{'} = \boldsymbol{H}_z(\gamma)\dot{\boldsymbol{x}}_d,\ \ddot{\boldsymbol{x}}_d^{'} = \boldsymbol{H}_z(\gamma)\ddot{\boldsymbol{x}}_d \end{aligned} \tag{5.34}$$

where x_t, y_t represent the $2D$ components of $\boldsymbol{x}_t$, $a = 0.3$, $b = 4$ and $t_0 = 1$. Different directions of movement are achieved by transforming the goals with $\boldsymbol{H}_z(\gamma)$, which defines a rotation around the $\boldsymbol{z}$-axis of an angle γ. Equations 5.27 and 5.28 enable

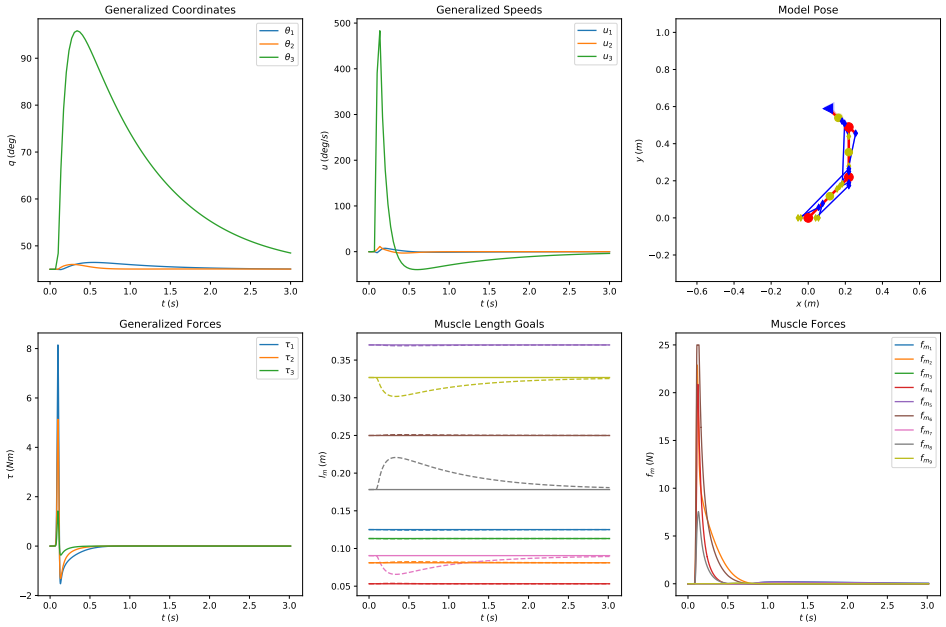


Figure 5.2: Posture analysis of the system in muscle space, including a "descending" command from the CNS. The first row of figures presents the simulated joint space coordinates and speeds. The second row depicts the evolution of the muscle space quantities of the controller. The model is perturbed and the response of the system is observed. The system absorbs the impact and returns to its initial configuration. The controller gains are $k_p = 10$, $k_d = 10$ and the loop delay $\tau_{so} = 20$ ms.

the exploration of the feasible muscle forces. In order to compute the feasible muscle force set $\boldsymbol{f}_m^{\oplus}$ (Equation 5.28), Equation 5.27 must be sampled for $\boldsymbol{f}_{m0}$. In this experiment, we compute the vertices of the polytope using Algorithm 5.1 with $d = 0$. The process is summarized in the following algorithm (Algorithm 5.3).

Figure 5.4 presents the simulation results for the simplified hand movement (movement along the $-\boldsymbol{x}$ direction, $\gamma = \pi$) collectively. We can inspect the evolution of the generalized coordinates and speeds as well as simulated/desired task goals and the corresponding task forces computed by the controller. Two instances of the simulation are isolated in Figure 5.5, where the feasible muscle forces that satisfy the task are further analyzed. The resulting feasible spaces are high-dimensional and various useful conclusions can be drawn. The feasible muscle force space changes as the movement progresses, since it depends on the action, the muscle model and the muscle routing properties.

The box plot shows the force variability of each muscle, which is attributed to the redundant nature of the system. From a control point of view, the variability of each muscle seems to be inversely proportional to the importance of its contribution to the current movement, following the Uncontrolled Manifold Hypothesis

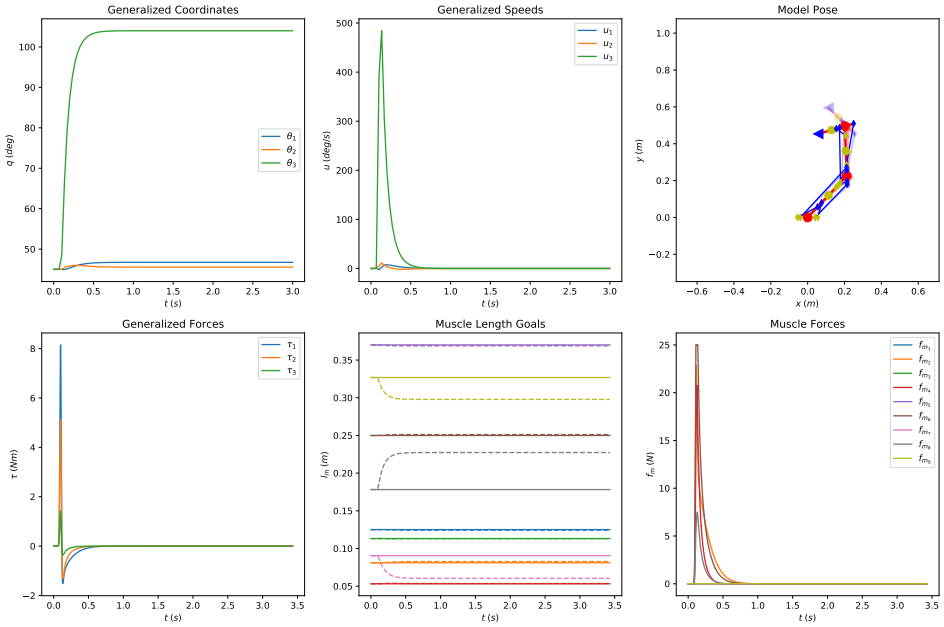


Figure 5.3: Posture analysis of the system in muscle space without "descending" command from the CNS. The first row of figures presents the simulated joint space coordinates and speeds. The second row depicts the evolution of the muscle space quantities of the controller. After the perturbation the system settles to a new, stable equilibrium point. The controller gains are $k_p = 0$, $k_d = 10$ and the loop delay $\tau_{so} = 20$ ms.

(UMH) (Latash et al., 2010). Figure 5.5 presents two instances of the of the simulated movement, one at the onset of the movement ($0.5s$) and one at the peak of the required effort ($0.9s$), permitting a qualitative comparison the feasible muscle forces. As the force requirements are small in the beginning of the movement (Figure 5.5a), the feasible muscle forces span the entire space. On the contrary (Figure 5.5b), as the demand for movement increases the agonists retain higher force profiles, while the antagonists' contribution is attenuated. The provided correlation matrix between different muscle groups reveals (anti-)synergies, which vary as the movement progresses. It can be concluded that there is a significant systematic correlation between different muscle pairs that emerges from the muscle routing properties and the functional constraints of the task.

5.3.3 The Effect of Null Space on the Joint Reaction Loads

The importance of evaluating the feasible muscle forces is demonstrated in the context of joint reaction analysis. An accurate estimation of the muscle forces is essential for the assessment of joint reaction loads. Consequently, the null space contributions can significantly alter the reaction forces without affecting the move-

Algorithm 5.3 Calculation of the feasible muscle forces as affected by the moment arm null space, muscle model and the action.

Input: action $\boldsymbol{\tau}$, muscle model $\mathcal{M}$, null space $\boldsymbol{N_{T_R}}$
Output: feasible muscle forces $\boldsymbol{f}_m^{\oplus}$
1: Find the particular muscle force solution $\boldsymbol{f}_m^{\parallel}$ (Equation 5.23)
2: Sample Equation 5.27 for $\boldsymbol{f}_{m0}$ (Algorithm 5.1)
3: Calculate the feasible muscle force set from Equation 5.28
4: **return** $\boldsymbol{f}_m^{\oplus}$

ment.

A benchmark gait movement, available from the `OpenSim` dataset (Delp et al., 2007) was used for this analysis. In a typical experimental setup the motion and externally applied forces are recorded. Given these recordings, IK and ID are performed in order to assess the model kinematics and kinetics required to track the experimental measurements (Erdemir et al., 2007). Instead of estimating the muscle forces using SO or some other method (Anderson and Pandy, 2001; Thelen and Anderson, 2006), we can recall Equation 5.21 and solve for $\boldsymbol{f}_m$ accounting for the null space muscle forces, which are present only when projecting from low- to high-dimensional space. Different realizations of muscle forces were calculated by appropriately sampling the feasible muscle force space (Equations 5.27 and 5.28). Finally, multiple joint reaction analyses were performed so as to evaluate the effect of the feasible muscle forces on the reaction loads.

Figure 5.6 presents the normalized (with respect to body weight) reaction forces on the hip joint during walking with the heel strike and toe-off events annotated accordingly. We observe that the results obtained from `OpenSim` joint reaction analysis predict low reaction load levels, since the minimum muscle effort criterion is used to compute the muscle forces, ignoring muscle co-contraction. On the contrary, it is possible to calculate the feasible reactions without making any prior assumption, which can limit the scope and extent of the analysis. Last and perhaps most importantly, the large range of possible values confirms that misinterpretation of the results is possible if the null space solutions are ignored.

5.4 Discussion-Conclusions

In this thesis, we approached the musculoskeletal redundancy problem by extending the kinematic and dynamic relationships using the notion of null space. This allowed us to study the problem more systematically and to avoid introducing unnecessary assumptions (e.g., muscle optimization). To achieve this level of modeling three operational spaces (task, joint and muscle space) were introduced (Figure 5.7), whose main purpose is to allow a distinction between the kinematic and dynamic redundancy effects. More specifically, the kinematically redundant DoFs in the context of a task were identified by considering the task and joint space re-

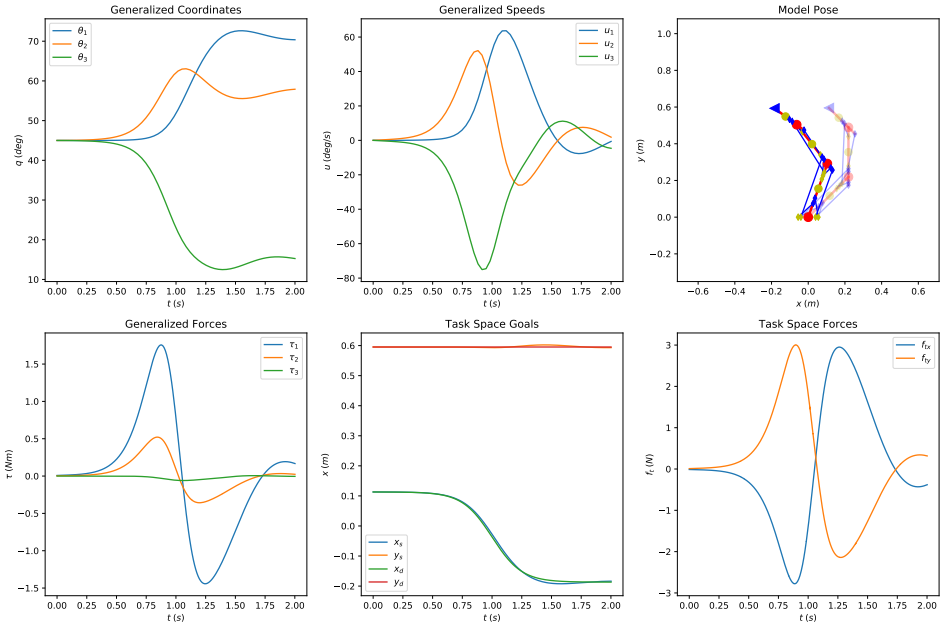
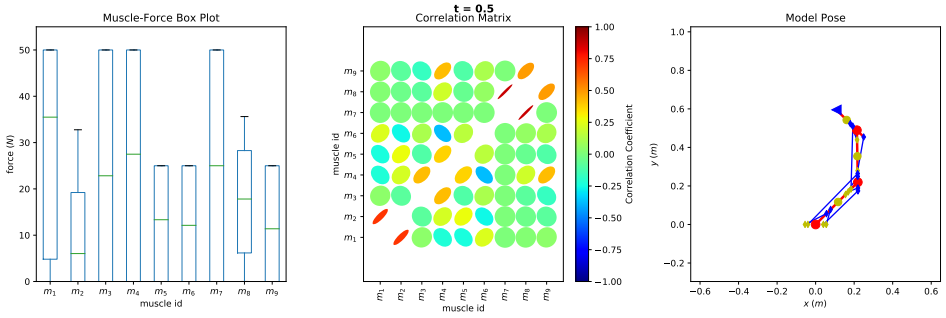


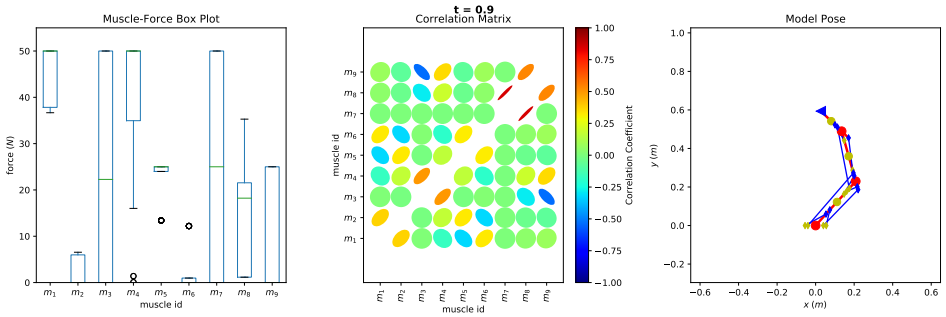
Figure 5.4: Collective simulation results for the simplified hand movement. The evolution of the general coordinates and speeds as well as the simulated movement are presented in the figures of the first row. The second row shows the generalized forces, the simulated and desired task goals and the task space forces computed by the controller. The parameters used in this experiment are $k_p = 50$, $k_d = 5$, $a = 0.3$, $b = 4$, $t_0 = 1$ and $\gamma = \pi$.

lationships. Likewise, the relations between joint and muscle or task and muscle spaces were used to infer the level of muscle co-contraction, which result in the same movement behavior. From a theoretical perspective, the main contribution is the modeling of musculoskeletal redundancy using the notion of null space and the calculation the feasible solution space, enabling various useful post-analyses (e.g., feasibility studies, joint reaction analysis, task/joint stiffness evaluation, etc.). The adoption of this framework can potentially enable the modeling and evaluation of hypotheses related to normal and pathological conditions presented in the coordination process, which can manifest in the null solution space (e.g., slack muscle disorder, increase in joint stiffness and rigidity).

From a control point of view, the purpose of projecting the EoMs onto some space is to enable the direct control of some quantity of interest. In this chapter, we used the various representations of the EoMs as ID model-based controllers in a MD scheme in order to perform FD simulations by controlling the task positions or muscle lengths. When dealing with models that possess a large number of DoFs, their posture is usually controlled by several simultaneous tasks (Sentis, 2007; Stanev and Moustakas, 2018). Task space representation is preferred, since the planning is encoded more naturally compared to joint space (e.g., animation of



(a)



(b)

Figure 5.5: The evolution of the feasible force set. Two instances of the simulated movement along the $-\boldsymbol{x}$ direction are shown to demonstrate the evolution of the feasible muscle space. The box plot depicts the force variability of each muscle as contributed to the redundant nature of the system. The correlation matrix shows the positive or negative correlation between the different muscle pairs.

characters). We introduced the muscle space projection where the controlled quantities are muscle lengths and derivatives. The disadvantage of this representation is that it is very difficult to construct the muscle length goals that will result in a coordinated movement such as walking. Despite this limitation, muscle space projection presents many attractive features. More DoFs permit coordination of different aspects of the movement (e.g., co-contraction) that are uncontrollable from a low-dimensional space. Furthermore, muscle projection provides a convenient representation for interfacing musculoskeletal and segmental level (proprioceptive) models and forms a basis for practical control applications, such as tendon-driven robots (Deshpande et al., 2010).

The various factors affecting kinematic and dynamic redundancy were identified and included in a closed form solution in order to calculate the feasible muscle forces. This not only demonstrated an effective approach for finding the family of feasible solutions, but also exposes the structure of the null space. The main advantage of the proposed approach is that the feasible muscle forces are action-specific, accounting for the dynamic evolution of the motion, while also satisfying

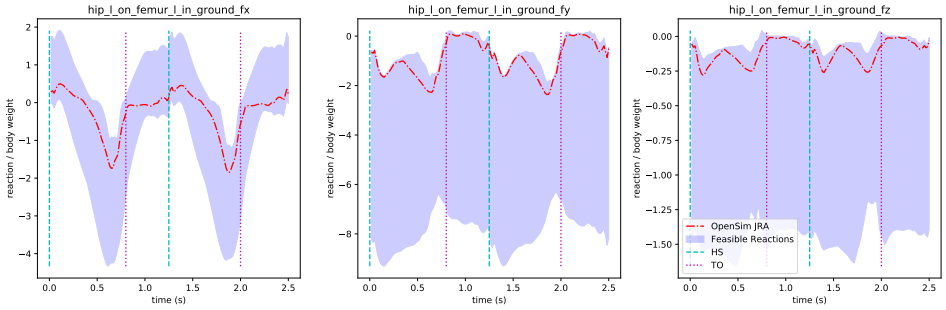


Figure 5.6: The effect of the null space muscle forces on the joint reaction loads. The normalized (with respect to body weight) reaction loads on the hip joint during walking are reported along with the heel strike and toe-off events. The shaded area is the reaction force range as attributed to the null space solutions of muscle forces. The red dotted line are the results obtained from `OpenSim`. The large range in the reaction loads confirms that misinterpretation of the results is possible if the null space solutions are ignored.

the physiological constraints of the muscles, outlining the various factors that affect the solution space. Another important aspect of this formulation is that different muscle models, including nonlinear Hill-type and muscle synergies, can be used in the derivation of the feasible inequality. The bottleneck of this method lies in the time complexity of the vertex enumeration algorithm, used for sampling the feasible space satisfying the constraints presented as linear inequalities. Given that the space defined by the inequality in Equation 5.25 is convex and bounded (Proposition 5.2), the complexity is cubic ($O(m^3)$) with respect to the number of muscles. In highly complicated musculoskeletal models with a large number of muscles, the aforementioned deterministic approach becomes computationally intractable due to the cubic time complexity growth. In such cases, randomized algorithms have to be employed to provide a representative sampling of the high-dimensional polytope (Vempala, 2005).

Two case studies were presented to demonstrate the application of the proposed methods for calculating the feasible muscle forces in the context of an arbitrary task. In the first case study, a simplified arm model was used for simulating a hand movement. The muscle force variabilities and muscle group correlations were calculated. It was shown that: 1) the variability of each muscle is inversely proportional to the importance of its contribution to the current movement and 2) there is a significant systematic correlation between different muscle pairs that emerges from the muscle routing properties and the functional constraints of the task. In the second case study, the feasible joint reaction loads were calculated for the hip joint during walking using a realistic full body model. From the resulting upper and lower bounds, it was concluded that misinterpretation of the results is possible if the null space solutions are ignored.

In this chapter, we discussed several extensions of the kinematic and dynamic equations in order to characterize the redundant nature of musculoskeletal systems. The formal treatment of the problem with well-known techniques stemming from linear algebra and projection operators revealed the means of interaction between the different relations, providing hints towards addressing important scientific questions related to motor coordination. Forming a complete description enabled the development of powerful tools for investigating the feasible outcomes of the system as compared to the local techniques that are suitable for finding only a particular solution. The redundant nature of the musculoskeletal system introduces variability/uncertainty in simulated quantities leading to misinterpretation of the results if ignored. Therefore, this groundwork provides the appropriate formalization to successfully address these issues, facilitating the application of broader types of studies in the realm of motor coordination.

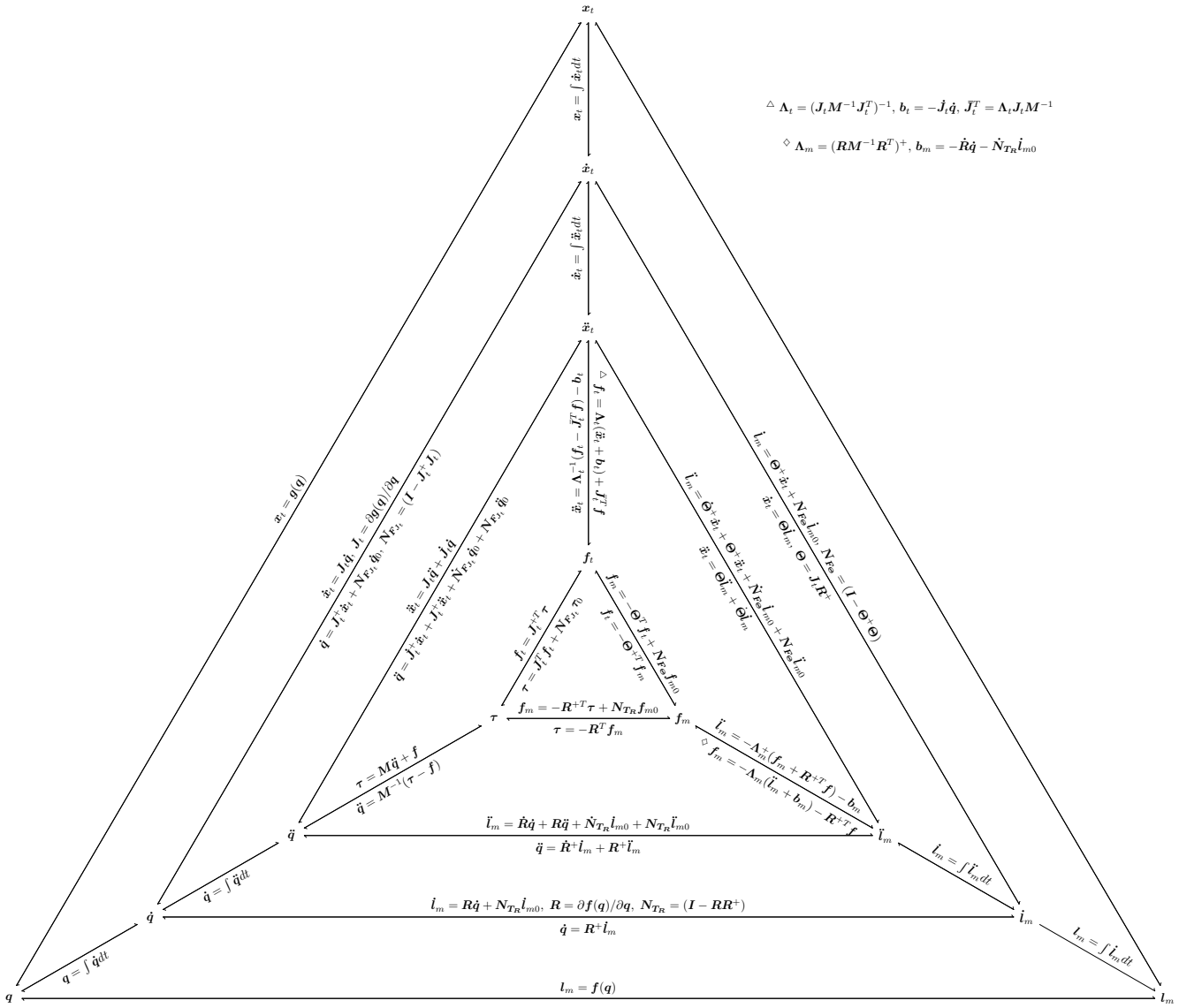


Figure 5.7: Overview of various relationships between task, joint and muscle space.

Chapter 6

Stiffness Modulation of Redundant Musculoskeletal Systems

In the previous chapter (chapter 5) a method for identifying the feasible muscle forces that satisfy the movement and physiological muscle constraints was developed. In this chapter, a model for computing the task and joint space stiffness using inverse methods that account for the musculoskeletal redundancy effects will be presented. This is particularly useful since the CNS does not coordinate the motion of the limbs alone, but also the degrees of muscle co-contraction, thus the stiffness is an important quantity that can provide valuable information during the analysis phase. The calculation of the feasible stiffness can aid in finding patterns of low stiffness capacity in order to improve the design of products and interventions that target these aspects of the movement.

The musculoskeletal task, joint and muscle stiffness are regulated by the CNS towards improving stability and interaction with the environment during movement. Many pathological conditions, such as Parkinson's disease, result in increased rigidity due to elevated muscle tone in antagonist muscle pairs, therefore stiffness quantities convey important information that complements the usual analyses by providing useful insights into levels otherwise inaccessible when only considering movement primitives. Musculoskeletal redundancy poses significant challenges in obtaining accurate stiffness results without introducing critical modeling assumptions. Currently, model-based estimation of stiffness relies on some objective criterion to deal with muscle over-availability (e.g., minimum effort), which, however, cannot be assumed to hold in every context. To alleviate this source of error, our approach explores the entire space of possible solutions that satisfy the action and the physiological muscle constraints. Using the notion of null space, the proposed framework rigorously accounts for the effect of muscle redundancy in the computation of the feasible stiffness characteristics. To confirm this, comprehensive case studies on hand movement and gait are provided, where the fea-

sible endpoint and joint stiffness is evaluated. Notably, this process enables the estimation of stiffness distribution, and aids in further investigation of alternative strategies available to the CNS. Such knowledge can significantly improve modeling by providing a holistic overview of dynamic quantities related to the human musculoskeletal system, despite its inherent redundancy.

6.1 Introduction

The regulation of the limbs' stiffness by the CNS has been subject to many studies over the past decades (Hogan, 1981; Houk and Rymer, 2011; Flash and Hogan, 1985; Flash, 1987; Flash and Mussa-Ivaldi, 1990; Perreault et al., 2001; Babikian et al., 2015). Intuitively, stiffness (or rigidity) is the extent to which the limbs resist movement induced by external forces. In this study, task (e.g., at endpoint), joint and muscle stiffness are distinguished and their relationship is studied. Since muscles are the main actors in musculoskeletal systems, their co-contraction affects the joint and task stiffness (Kutch and Valero-Cuevas, 2011; Inouye and Valero-Cuevas, 2016). Due to the over-availability of actuators, there exist infinitely many solutions for muscle forces giving rise to the same movement, it is thus important to consider not only motion primitives, but also instances and degrees of co-contraction. Minimum effort solutions (e.g., activation squared) tend to underestimate muscle co-contraction, since co-contraction has by construction zero net joint effect, rendering it unsuitable in application where this assumption may not hold (Pedotti et al., 1978; Prilutsky and Zatsiorsky, 2002; Koeppen et al., 2017). Therefore, appropriate methods need to be developed for finding the possible system realizations and further classifying strategies available to the CNS.

Experimental methods for measuring the endpoint stiffness have been proposed (Mussa-Ivaldi et al., 1985; Flash and Mussa-Ivaldi, 1990) and used in conjunction with model-based estimates in order to study the effect of muscle co-contraction on it. Model-based estimation of joint and task stiffness requires the assessment of muscle forces, which can be obtained either from EMG recordings or by minimizing some objective criterion, such as minimum effort. However, both approaches have their drawbacks, namely not all muscles are easily accessible in practice (e.g., deep muscles), transformation from EMG to muscle force is inexact and optimization assumptions do not always hold. The authors in (Hu et al., 2011) compared the experimentally measured endpoint stiffness against the model-based estimates, concluding that deviations from their predictions can be explained by muscle redundancy. To overcome these limitations, a method for identifying all possible solutions of muscle forces that satisfy the task as well as physiological muscle constraints is proposed, in order to evaluate their influence on the task and joint space stiffness.

The main contribution of is the calculation of the feasible task and joint stiffness in the context of an arbitrary action, by modeling musculoskeletal kinematic and dynamic redundancy using the notion of null space. In subsection 6.2.1, the

relationship between muscle, joint and task space stiffness is expressed in a way that highlights the influence of musculoskeletal over-availability on the calculated quantities. A method for describing the feasible muscle forces is proposed in subsection 6.2.2 so as to enable the calculation of the feasible stiffness. The results demonstrate the importance of this analysis and its influence on the estimated stiffness properties. The feasible task stiffness of a simplified arm model performing a hand movement and the joint stiffness of a realistic gait model are studied in Subsections 6.3.1 and 6.3.2, respectively. The source code along with any related material for this study are publicly available (section A.1), providing simple examples so that the readers can reproduce, understand and reuse the presented methods. These case studies highlight the assumption-free nature of the proposed framework, which therefore is able to recover the entire stiffness space for the input models.

6.2 Methods

6.2.1 Muscle, Joint and Task Space Stiffness

In this section we will revisit the relationships between muscle-joint and joint-task space stiffness, so as to establish a link between muscle-task space stiffness. In the next section, the influence of muscle redundancy on the calculated stiffness will be examined. The muscle stiffness is defined as

$$\boldsymbol{K}_m = \frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{l}_m},\ \boldsymbol{K}_m \in \Re^{m \times m} \tag{6.1}$$

where $\boldsymbol{f}_m \in \Re^m$ represents the muscle forces, $\boldsymbol{l}_m \in \Re^m$ the musculotendon lengths and m the number of muscles. The joint stiffness is defined similarly

$$\boldsymbol{K}_j = \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{q}},\ \boldsymbol{K}_j \in \Re^{n \times n} \tag{6.2}$$

where $\boldsymbol{\tau} \in \Re^n$, $\boldsymbol{q} \in \Re^n$ denote the generalized forces and coordinates, respectively and n the DoFs of the system. In order to relate the two definitions above, the kinematic and dynamic relations between muscle and joint space must be considered. The musculotendon lengths are given as a function of the generalized coordinates and depend on the muscle routing parameters (ϕ)

$$\boldsymbol{l}_m = \boldsymbol{f}(\boldsymbol{q}; \phi) \tag{6.3}$$

implying that the generalized coordinates suffice to fully define it, assuming that the muscles are pretensioned. The derivative of Equation 6.3 with respect to time (dot notation) is given by

$$\dot{\boldsymbol{l}}_m = \boldsymbol{R}(\boldsymbol{q})\dot{\boldsymbol{q}},\ \boldsymbol{R}(\boldsymbol{q}) = \frac{\partial \boldsymbol{l}_m}{\partial \boldsymbol{q}} \tag{6.4}$$

where the moment arm matrix ($\boldsymbol{R} \in \Re^{m \times n}$) describes the mapping from joint to muscle space ($\Re^n \rightarrow \Re^m$). Furthermore, this matrix defines a dual relationship between motion and force quantities. In this case, the virtual work principle can be used to establish the link between joint and muscle space forces

$$\begin{aligned} \boldsymbol{\tau}^T \delta \boldsymbol{q} &= -\boldsymbol{f}_m^T \delta \boldsymbol{l}_m \\ \boldsymbol{\tau}^T \delta \boldsymbol{q} &= -\boldsymbol{f}_m^T \boldsymbol{R} \delta \boldsymbol{q} \\ \boldsymbol{\tau} &= -\boldsymbol{R}^T \boldsymbol{f}_m \end{aligned} \tag{6.5}$$

where the negative sign convention is introduced, admitting that a muscle induces positive work during shortening ($\dot{\boldsymbol{l}}_m \prec 0$). Equation 6.5 can be differentiated with respect to $\boldsymbol{q}$

$$\begin{aligned} \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{q}} &= -\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{q}} \bullet_2 \boldsymbol{f}_m - \boldsymbol{R}^T \frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{l}_m} \frac{\partial \boldsymbol{l}_m}{\partial \boldsymbol{q}} \\ \boldsymbol{K}_j &= -\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{q}} \bullet_2 \boldsymbol{f}_m - \boldsymbol{R}^T \boldsymbol{K}_m \boldsymbol{R} \end{aligned} \tag{6.6}$$

where the first term captures the varying effect of the moment arm, while the second term maps the muscle space stiffness to joint space. The notation $\bullet_2$ (section A.3) denotes a product of a rank-3 tensor ($\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{q}} \in \Re^{n \times m \times n}$, a 3D matrix) and a rank-1 tensor ($\boldsymbol{f}_m \in \Re^m$, a vector), where the index 2 specifies that the tensor dimensionality reduction (by summation) is performed across the second dimension, resulting in a reduced rank-2 tensor of dimensions $n \times n$ (Kolda and Bader, 2009).

In order to compute the joint space stiffness using the previous equation, a model for selecting $\boldsymbol{K}_m$ must be defined. The two most adopted approaches are to either use the force-length characteristics of the muscle model or to approximate it using the definition of the short range stiffness, where the latter is shown to explain most of the variance in the experimental measurements (Hu et al., 2011). The short range stiffness is proportional to the force developed by the muscle (f_m)

$$k_s = \gamma \frac{f_m}{l_m^o} \tag{6.7}$$

where the $\gamma = 23.4$ constant has been experimentally determined (Cui et al., 2008) and l_m^o is the optimal muscle length.

Continuing with the relationship between joint and task space stiffness, we start by providing the task stiffness definition

$$\boldsymbol{K}_t = \frac{\partial \boldsymbol{f}_t}{\partial \boldsymbol{x}_t}, \ \boldsymbol{K}_t \in \Re^{d \times d} \tag{6.8}$$

where $\boldsymbol{f}_t \in \Re^d$ denotes the forces, $\boldsymbol{x}_t \in \Re^d$ the positions and d the DoFs of the task. The task position is given as a function of the generalized coordinates

$$\boldsymbol{x}_t = \boldsymbol{g}(\boldsymbol{q}). \tag{6.9}$$

A task has an abstract meaning, which can be interpreted either as a position, an orientation or a spatial primitive, which is a combination of position and orientation (Sentis, 2007). The first derivative of Equation 6.9 with respect to time is given by

$$\dot{\boldsymbol{x}}_t = \boldsymbol{J}_t(\boldsymbol{q})\dot{\boldsymbol{q}},\ \boldsymbol{J}_t(\boldsymbol{q}) = \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{q}} \tag{6.10}$$

where the task Jacobian matrix ($\boldsymbol{J}_t \in \Re^{d\times n}$) describes the mapping from joint to task space ($\Re^n \rightarrow \Re^d$). From the virtual work principle (as in Equation 6.5) we can establish a relationship between task and joint space forces

$$\begin{aligned} \boldsymbol{\tau}^T \delta\boldsymbol{q} &= \boldsymbol{f}_t^T \delta\boldsymbol{x}_t \\ \boldsymbol{\tau}^T \delta\boldsymbol{q} &= \boldsymbol{f}_t^T \boldsymbol{J}_t \delta\boldsymbol{q} \\ \boldsymbol{\tau} &= \boldsymbol{J}_t^T \boldsymbol{f}_t. \end{aligned} \tag{6.11}$$

Equation 6.11 can be differentiated with respect to $\boldsymbol{q}$

$$\begin{aligned} \frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{q}} &= \frac{\partial \boldsymbol{J}_t^T}{\partial \boldsymbol{q}} \bullet_2 \boldsymbol{f}_t + \boldsymbol{J}_t^T \frac{\partial \boldsymbol{f}_t}{\partial \boldsymbol{x}_t}\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{q}} \\ \boldsymbol{K}_j &= \frac{\partial \boldsymbol{J}_t^T}{\partial \boldsymbol{q}} \bullet_2 \boldsymbol{f}_t + \boldsymbol{J}_t^T \boldsymbol{K}_t \boldsymbol{J}_t \end{aligned} \tag{6.12}$$

and by solving for $\boldsymbol{K}_t$ the forward (from high- to low-dimensional space) mapping is obtained

$$\boldsymbol{K}_t = \boldsymbol{J}_t^{+T}\left(\boldsymbol{K}_j - \frac{\partial \boldsymbol{J}_t^T}{\partial \boldsymbol{q}} \bullet_2 \boldsymbol{f}_t\right)\boldsymbol{J}_t^+. \tag{6.13}$$

Combining Equations 6.13 and 6.6, we can arrive at the direct mapping between the task and muscle space stiffness

$$\boldsymbol{K}_t = -\boldsymbol{J}_t^{+T}\left(\frac{\partial \boldsymbol{J}_t^T}{\partial \boldsymbol{q}} \bullet_2 \boldsymbol{f}_t + \frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{q}} \bullet_2 \boldsymbol{f}_m + \boldsymbol{R}^T \boldsymbol{K}_m \boldsymbol{R}\right)\boldsymbol{J}_t^+. \tag{6.14}$$

6.2.2 The Influence of Kinematic and Dynamic Redundancy

In this section, we will make use of the method introduced in chapter 5 for identifying the feasible muscle forces that satisfy both the task and the physiological muscle constraints (Stanev and Moustakas, 2019). This method does not assume minimization of some objective criterion, but relies solely on the task context, the muscle model (e.g., linear/nonlinear muscle model and synergy encoding) and the anatomical properties of the muscle routing (the muscle moment arm and its null space). The family of solutions is guaranteed to preserve the system's kinematic behavior, i.e., produce the same movement, corresponding to different levels of muscle co-activation. We postulate that task and joint stiffness are important at

regulating quantities that characterize each distinct solution (e.g., high and low joint stiffness).

In a typical experimental setup the motion and externally applied forces are recorded. Given these recordings, IK and ID are performed in order to assess the model kinematics and kinetics that satisfy the experimental measurements (Erdemir et al., 2007). Instead of estimating the muscle forces using SO or some other method (Anderson and Pandy, 2001; Thelen and Anderson, 2006), we can recall Equation 6.5 and solve for $\boldsymbol{f}_m$ accounting for the null space muscle forces, which are present only when projecting from low- to high-dimensional space. More formally, the muscle forces $\boldsymbol{f}_m$ can be expressed as the sum of two mutually orthogonal subspaces, spanning the column ($\boldsymbol{f}_m^{\parallel} \in \mathbb{C}(\boldsymbol{R}^{+T})$) and null space ($\boldsymbol{f}_m^{\perp} \in \mathbb{N}(\boldsymbol{R}^{+})$), respectively

$$\boldsymbol{f}_m = \boldsymbol{f}_m^{\parallel} + \boldsymbol{f}_m^{\perp} = -\boldsymbol{R}^{+T}\boldsymbol{\tau} + \boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0},\ \boldsymbol{N}_{\boldsymbol{T_R}} = \boldsymbol{I} - \boldsymbol{R}\boldsymbol{R}^{+} \tag{6.15}$$

where $\boldsymbol{N}_{\boldsymbol{T_R}} \in \Re^{m\times m}$ is the moment arm null space projection operator and $\boldsymbol{f}_{m0} \in \Re^m$ a vector of arbitrarily selected null space muscle forces. Note that this definition spans the entire $\Re^m$ for some arbitrary value of $\boldsymbol{\tau}$, whereas in reality muscle forces are strictly positive (contraction) and bounded (limited force). Therefore, in general, this solution may not be physiologically correct (namely $\exists i : f_m^i > f_{\max}^i$ or $f_m^i < 0$). However, the null space term $\boldsymbol{f}_m^{\perp}$ can provide a suitable correction in order to satisfy the physiological constraints of the muscles. In this study, we will assume a linear muscle model, without loss of generality

$$\boldsymbol{f}_m = \boldsymbol{f}_{\max} \circ \boldsymbol{a}_m,\ \boldsymbol{0} \preceq \boldsymbol{a}_m \preceq \boldsymbol{1} \tag{6.16}$$

where $\boldsymbol{a}_m \in \Re^m$ represents a vector of muscle activations, $\boldsymbol{f}_{\max} \in \Re^m$ a vector specifying the maximum muscle force and $\circ$ the Hadamard (elementwise) product. We can impose bounds on the possible solutions of $\boldsymbol{f}_{m0}$ in the form of linear inequalities, by noting that Equations 6.15 and 6.16 must be equal

$$\begin{gathered}\boldsymbol{f}_m = \boldsymbol{f}_{\max} \circ \boldsymbol{a}_m = \boldsymbol{f}_m^{\parallel} + \boldsymbol{N}_{\boldsymbol{T_R}}\boldsymbol{f}_{m0},\ \boldsymbol{0} \preceq \boldsymbol{a}_m \preceq \boldsymbol{1} \rightarrow \\ \begin{bmatrix} -\boldsymbol{N}_{\boldsymbol{T_R}} \\ \hdashline \boldsymbol{N}_{\boldsymbol{T_R}} \end{bmatrix} \boldsymbol{f}_{m0} \preceq \begin{bmatrix} \boldsymbol{f}_m^{\parallel} \\ \hdashline \boldsymbol{f}_{\max} - \boldsymbol{f}_m^{\parallel} \end{bmatrix} \\ \boldsymbol{Z}\boldsymbol{f}_{m0} \preceq \boldsymbol{\beta}\end{gathered} \tag{6.17}$$

where the upper part ensures that $\boldsymbol{f}_m$ is strictly positive and the lower part that $\boldsymbol{f}_m$ has a finite upper bound. An important advantage of this formulation is that different muscle models can be included in the derivation of the feasible inequality. For example, we can incorporate muscle synergies (Steele et al., 2015) by substituting $\boldsymbol{a}_m = \boldsymbol{W}\boldsymbol{c}$ in Equation 6.17. Term $\boldsymbol{W} \in \Re^{m\times s}$ represents the muscle synergy matrix and $\boldsymbol{c} \in \Re^s$ the synergy activations. Moreover, Equation 6.17 defines a closed, convex space for $\boldsymbol{f}_{m0}$ (Proposition 5.2), which can therefore be sampled using vector enumeration techniques (Avis and Fukuda, 1992; Vempala, 2005). Finally, the feasible muscle force set is calculated by

$$\boldsymbol{f}_m^{\oplus} = \{\boldsymbol{f}_m^{\parallel} + \boldsymbol{N}_{\boldsymbol{T}_R}\boldsymbol{f}_{m0}^{i},\ \forall i\}. \tag{6.18}$$

6.3 Results

We will present two case studies, where the influence of musculoskeletal redundancy on the feasible task and joint stiffness will be evaluated. In the first case, a simplified planar arm model (section A.4 and Figure A.1), having three DoFs and nine muscles, is appropriately constructed to demonstrate both kinematic and dynamic redundancy (i.e., $d < n < m$). The movement will be planned in task space by controlling the position of the hand using task space projection (Stanev and Moustakas, 2018). In the second experiment, the feasible joint stiffness of the hip, knee and ankle joints during walking will be assessed using the `OpenSim` (Delp et al., 2007) gait model that has ten DoFs and eighteen muscles (Millard et al., 2013). These models were constructed with an emphasis on clarity, simultaneously preserving a degree of anatomic realism.

6.3.1 Variability in the Arm Endpoint Stiffness

In this experiment, the feasible task and joint stiffness of a simplified arm model will be evaluated. More specifically, Equations 6.6 and 6.14 will be used to compute the stiffness at each time step of the simulation. Since stiffness depends on the muscle forces and the model is redundant, there are infinitely many solutions that satisfy the reaching task. To account for this, the space defined by Equation 6.17 will be sampled in order to compute the feasible muscle forces (Equation 6.18), assuming a linear muscle model. Consequently, these solutions will be used to obtain the feasible stiffness values.

Since task forces $\boldsymbol{f}_t$ are required in the calculation of Equation 6.17, the planning can be encoded using task space projection (Khatib et al., 2009; Stanev and Moustakas, 2018), by tracking the desired position of the end effector. Essentially, a mixed dynamics scheme is adopted, where the task space ID model-based controller accepts the desired task goal and returns the generalized forces $\boldsymbol{\tau}$ (Equation 6.11). These forces are applied to the model in a FD manner and the resulting movement is thus simulated. Furthermore, the generalized forces are used to evaluate the particular solution $\boldsymbol{f}_m^{\parallel}$ (Equation 5.21), required in the calculation of the feasible muscle forces. A PD tracking scheme is adopted

$$\ddot{\boldsymbol{x}}_t = \ddot{\boldsymbol{x}}_d + k_p(\boldsymbol{x}_d - \boldsymbol{x}_t) + k_d(\dot{\boldsymbol{x}}_d - \boldsymbol{x}_t) \tag{6.19}$$

where $\boldsymbol{x}_d,\ \dot{\boldsymbol{x}}_d,\ \ddot{\boldsymbol{x}}_d$ are the desired position, velocity and acceleration of the task and $k_p = 50,\ k_d = 5$ the tracking gains. The desired task goal is derived from a smooth sigmoid function that produces bell-shaped velocity profiles in any direction around the initial position of the end effector

$$\begin{aligned} \boldsymbol{x}_d(t) &= \left[x_t(0) + \frac{a}{2}(\tanh(b(t-t_0)) + 1),\ y_t(0)\right]^T \\ \dot{\boldsymbol{x}}_d(t) &= \frac{d\boldsymbol{x}_d(t)}{dt},\ \ddot{\boldsymbol{x}}_d(t) = \frac{d\dot{\boldsymbol{x}}_d(t)}{dt} \\ \boldsymbol{x}_d^{'} &= \boldsymbol{H}_z(\gamma)\boldsymbol{x}_d,\ \dot{\boldsymbol{x}}_d^{'} = \boldsymbol{H}_z(\gamma)\dot{\boldsymbol{x}}_d,\ \ddot{\boldsymbol{x}}_d^{'} = \boldsymbol{H}_z(\gamma)\ddot{\boldsymbol{x}}_d \end{aligned} \tag{6.20}$$

where x_t, y_t represent the $2D$ components of $\boldsymbol{x}_t$, $a = 0.3$, $b = 4$ and $t_0 = 1$. Different directions of movement are achieved by transforming the goals with $\boldsymbol{H}_z(\gamma)$, which defines a rotation around the $\boldsymbol{z}$-axis of an angle γ.

Figure 6.1 collects three instances of the simulated movement (along the $-\boldsymbol{x}$ direction, $\gamma = \pi$). The left diagram shows the feasible endpoint stiffness using scaled ellipses (scaling = 0.0006), whose major and minor axes are the eigenvectors of the 2D stiffness matrix. The ellipse is a common way to visualize the task stiffness (Flash and Mussa-Ivaldi, 1990; Cui et al., 2008), where the major axis (red) of the ellipse is oriented along the maximum stiffness and the area is proportional to the determinant of $\boldsymbol{K}_t$, conveying the stiffness amplitude. The stiffness capacity (area) is increased in the last pose, since the arm has already reached its final position and muscle forces aren't needed for it to execute any further motion. The second diagram (middle) depicts the distribution of ellipse parameters (area and orientation ϕ). Finally, the rightmost box plot shows the feasible joint stiffness distribution at three distinct time instants. Experimental measurements (Perreault et al., 2001) have showed that the orientation of stiffness ellipses varies in a range of about 30°. While our simulation results confirm this, they also reveal a tendency of fixation towards specific directions for higher stiffness amplitudes. The large variation of feasible stiffness verifies that this type of analysis conveys important findings that complement experimental observations.

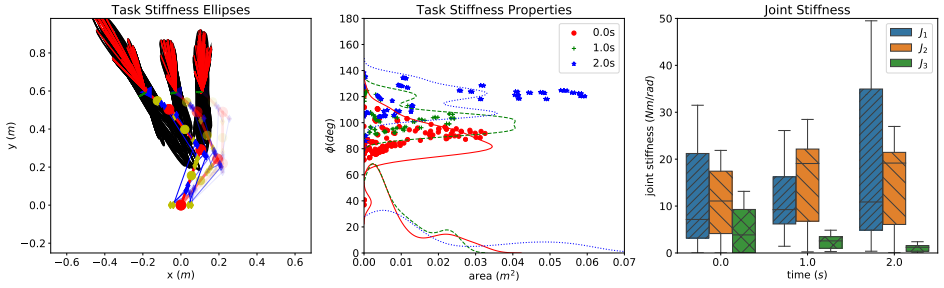


Figure 6.1: Collected results of the simulated arm movement (along the $-\boldsymbol{x}$ direction, $\gamma = \pi$) at three time instants. Left diagram shows the feasible endpoint stiffness using scaled ellipses (scaling = 0.0006), where the red and green lines denote the major and minor axes, respectively. The middle plot illustrates the ellipse parameter distribution (area and orientation ϕ). The box plot on the right depicts the feasible joint stiffness distribution.

6.3.2 Calculation of the Feasible Joint Stiffness During Walking

This experiment involves a model available in the `OpenSim` repository, used to estimate the joint stiffness during walking. The model has ten DoFs and a reduced set of eighteen Hill-type muscles (Millard et al., 2013) that are important for gait. The model was first scaled using a static pose in order to match the subject-specific anthropometrics. Afterwards, IK and ID were performed to obtain the kinematics and kinetics required to track the experimental marker trajectories and match the ground reaction forces.

Recall that in the definition of joint stiffness (Equation 6.6) the muscle moment arm must be differentiated with respect to the generalized coordinates. Unfortunately, `OpenSim` lacks the analytical means for evaluating higher order derivatives, as it computes the muscle moment arm numerically from the input joint configuration. In order to derive a symbolic representation, multivariate polynomial fitting (van den Bogert et al., 2011) was performed on samples of the muscle moment arm at different configurations. To reduce the complexity and improve the robustness of the fit, we determined the coordinates affecting each element in the moment arm matrix, by identifying the DoFs spanned by each muscle. Figure 6.2 compares the sampled and symbolically obtained moment arm of the vastus intermedius (a mono-articular muscle) at the knee joint as a function of the knee flexion angle and the moment arm of the hamstring muscle at the knee joint as a function of the hip and knee flexion angles.

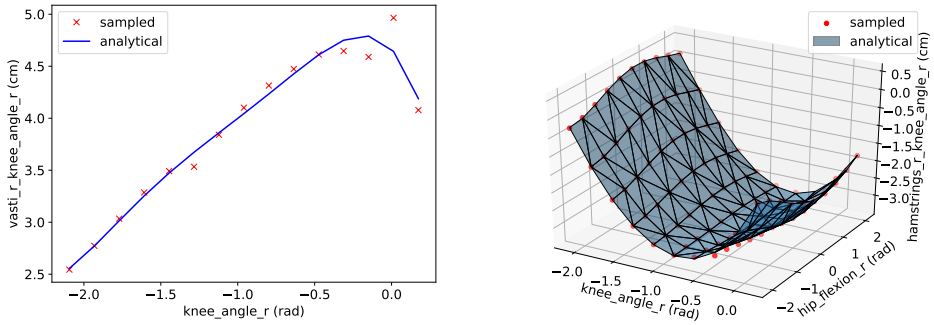


Figure 6.2: Comparison between the sampled and symbolically obtained moment arm of the vastus intermedius muscle at the knee joint as a function of the knee flexion angle (left). Likewise, the moment arm of the hamstring muscle at the knee joint as a function of the hip and knee flexion angles (right).

The feasible muscle force set was computed at each time step of the analysis, using Equations 5.25 and 5.28, assuming a linear muscle model. The short range stiffness (Equation 6.7) was used to form the muscle stiffness diagonal matrix $\boldsymbol{K}_m$, which does not include inter-muscle coupling, since it is difficult to model and measure in practice.

Figure 6.3 depicts the feasible joint stiffness of the hip, knee and ankle joints during walking with the heel strike and toe-off events annotated accordingly. These results confirm experimental measurements (Shamaei et al., 2015) and furthermore present similarities in the outline of the minimum stiffness predicted by our method. Notably, the hip stiffness range is gradually decreasing between heel strike and toe-off, because the flexor muscles are preparing for the swing phase and the capacity to increase the joint stiffness riches its lowest value before the toe-off event. A similar pattern is observed at the knee joint, which undergoes a flexion and a subsequent extension during the swing phase. We observe that the capacity of the muscles to modulate the ankle stiffness is not decreased and the range is gradually shifted upwards in the region between the heel strike and toe-off events. The increase in the minimum possible values of the ankle stiffness is attributed to the counterbalance of the ground reaction forces by the ankle plantar flexion muscles. As muscle effort is spent by these muscles, one would expect a lower maximum bound, which is not the case here. This could be contributed to the fact that the musculoskeletal system is asymmetric, i.e., the plantar flexion muscles can induce larger magnitudes of moment at the ankle joint in comparison to the dorsiflexion muscles. We can conclude that the contribution of the ground reaction forces results in an increase of the ankle stiffness.

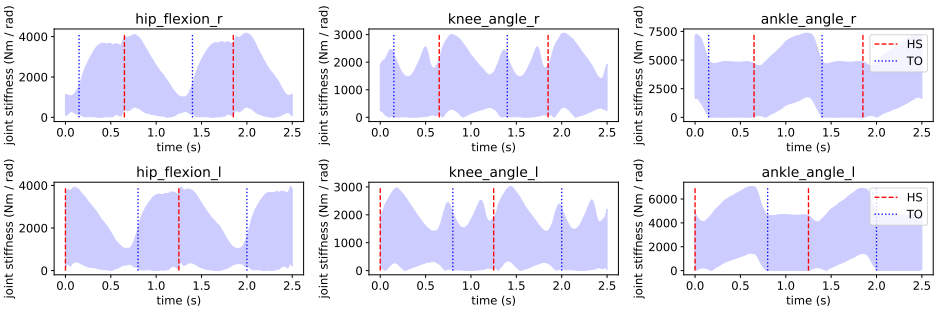


Figure 6.3: The feasible joint stiffness of the hip, knee and ankle joints during walking with the heel strike (HS) and toe-off (TO) events annotated accordingly.

6.4 Discussion-Conclusions

The dynamic evolution of task and joint stiffness is important because the CNS does not coordinate the motion of the limbs alone, but also regulates the overall stability, impedance and admittance of the musculoskeletal system. As musculoskeletal systems are intrinsically redundant, it is very difficult to interpret the hierarchical muscle activation patterns, especially when different muscle groups are co-activated. Considering these facts, we show that global stiffness evaluation is possible, hence this metric can be used to characterize various actions and complement traditional analyses. The calculation of the feasible stiffness can aid in

finding patterns of low stiffness capacity in order to improve the design of products and interventions that target these aspects of movement (e.g., exoskeleton design, interaction with the environment, ergonomics, etc.). Moreover, this kind of analysis can be used in combination with the single solution methods in order to evaluate the uncertainty in the estimated quantities, due to redundancy.

While task and joint stiffness can be measured experimentally using specialized equipment, one cannot estimate the muscle stiffness from those measurements alone, because the mapping from low- (task or joint) to high-dimensional muscle stiffness is not unique. Therefore, accurate estimation of muscle stiffness from task or joint stiffness is in general not possible, whereas validation would require direct measurement. Consequently, identification of the feasible solution space, as outlined in this work, can help to properly interpret results obtained from the redundant musculoskeletal systems and to quantify the uncertainty in the estimated quantities due to musculoskeletal redundancy.

The presented studies demonstrate the application of the proposed methodology in the context of real-world tasks. The first used a simplified model to study the arm endpoint stiffness, while the second used a more realistic `OpenSim` gait model to examine the joint stiffness during walking. The main weakness of the proposed method is that the accuracy in the predicted quantities depends on the quality of the musculoskeletal model that is being used, thus it is important to examine the sensitivity of the feasible stiffness with respect to the uncertainties in the model parameters. Nevertheless, the obtained results confirm previous findings and outline the feasible solution space emphasizing that misinterpretation due to large variability is possible if the null space solutions are ignored. The analysis provides useful conclusions and insights, such as the balance between stiffness capacity and task requirements, indicating that higher task requirements reduce the ability for stiffness modulation.

In this chapter a method to determine the feasible task and joint stiffness of the musculoskeletal system for any movement was presented. Undoubtedly, this is crucial for understanding the muscle coordination mechanisms and the various strategies available to the CNS, aiding in the development of effective evaluation and treatment of disorders such as Parkinson's disease. Practical and experimental limitations severely hinder the in vivo measurement of stiffness, while model-based estimation suffers from the musculoskeletal redundancy. To overcome the latter limitation, we choose to model the muscle over-availability using the notion of null space and identify the possible solutions that satisfy the task and muscle constraints. Results show that the musculoskeletal system is capable of achieving a highly variable stiffness using muscle co-contraction, highlighting the importance of performing feasibility studies.

Chapter 7

Concluding Remarks

The principles of computer modeling and simulation have helped engineers solve complex problems that have literally transformed our everyday life. Recent advances in different fields have laid the foundations to apply modeling and simulation in the diagnosis, treatment, prevention of diseases and development of products. Biological models are hard to model and analyze mainly due to their inherit hidden complexity and our inability to inspect different parts of the system with low uncertainty in the measured quantities. Perhaps the biggest challenge is to show that the model predictions are robust and accurate, convening important information. The extraordinary complexity of the human body and the rise of the digital age calls for the development of novel methods to address the current challenges.

In chapter 3, a case study that aims to create a decision support system for the reconstruction of the ACL after injury based on personalized musculoskeletal models and simulations, was presented. It was further shown that computational musculoskeletal models are a valuable tool for predicting the outcome of surgical decision, demonstrating that such analyses may lead to improved clinical decision-making and ultimately a better health outcome or lower cost of treatment. However, we also highlight, and address latter, two major problems that can render the application of musculoskeletal modeling and simulation frequently inappropriate for clinical practice. These limitations are not a drawback of the proposed approach, but are rather inherit issues that arise from the nature of musculoskeletal models and the current methods that are used in simulation and analysis.

There are mainly two major approaches for analyzing musculoskeletal systems, namely either in an inverse or forward dynamics manner. Inverse methods rely on experimentally measured kinematics and externally applied forces, and are used to investigate situations that are based on these measurements alone. On the other hand, forward methods depend on the knowledge of the muscle activity that generates a coordinate movement and are used for prediction. We would like to use musculoskeletal models to predict things that have not happened, yet given the current technological advances, it is more practical to record a movement and perform inverse simulation. It seems like we are faced with a “catch-22” paradoxical

situation and things are even worse if we consider the issue of musculoskeletal redundancy. More specifically, musculoskeletal systems are redundant by nature, because there are more DoFs than those required to perform certain kinematic tasks and each DoF is actuated by multiple muscles. This over-availability poses numerous challenges in the process of modeling and simulation that can negatively affect the validity of the models and the obtained results. It turns out that coordinate projection methods are a valuable tool for addressing these issues.

The possibility of planning virtual simulations without necessarily relying on experimental measured kinematics and muscle activity was examined in this dissertation. The traditional analysis of musculoskeletal systems relies mostly on experimental measured kinematics and externally applied forces. This limits the scope and extend as these methods as they are not used for prediction, but rather for analyzing situations that have already happened. Musculoskeletal modeling and simulation should be used to predict whether a change in the surgical procedure or the introduction of an assistive device will alter the kinematics and dynamics of the system. Some of the challenges are to create both accurate models that capture the physical and physiological properties of the system and scenarios that are anatomically feasible and plausible. Task-based, MD and optimization-based methods are indispensable tools that can pave the way towards achieving this goal.

Forward and inverse methods can be combined into a MD scheme in order to perform virtual simulations of constrained musculoskeletal systems that do not necessarily relying on experimental measured kinematics and muscle activity. Task space projection was used to solve the planning problem, as a complex movement can be decomposed into simple interrelating task goals. Constraint projection was adopted in the design of the ID model-based controllers in order to account for realistic anatomical joints and complex interaction with the environment that require constraint modeling. Two methods that operate in task space, namely a dynamics-based method for solving the IK problem and a method for performing muscle driven dynamics simulations that closely reproduce experimental measurements of kinematics and GRF, were presented. Perhaps the biggest limitation of the MD scheme is that it is a dynamics-based method that uses numerical integration which is typically unsuitable for real-time application, using current computing machines.

The biggest challenge in musculoskeletal analysis is to demonstrate that model predictions are robust and accurate. Given that musculoskeletal systems are inherently redundant, meaning that there are infinitely many possible solutions of muscle forces that give rise to the same movement, we can never be sure whether quantities (e.g., joint reaction loads, task and joint stiffness, etc.) that rely on these estimates can be computed reliably. There are different methods that can select one of the many solutions by "supposingly" modeling the strategy adopted by CNS. However, it is extremely difficult to model this process, especially in pathological conditions such as Parkinson's disease. Having this in mind, the main contribution of this dissertation is the modeling of musculoskeletal redundancy using the notion of null space and the calculation the feasible solution space, enabling useful post-analyses (e.g., feasibility studies, joint reaction analysis, task/joint stiffness evalu-

ation, etc.). The main advantage of this approach is that the feasible muscle forces are action-specific, accounting for the dynamic evolution of the motion, while also satisfying the physiological constraints of the muscles, outlining the factors that affect the solution space. The bottleneck of this method lies in the time complexity of the vertex enumeration algorithm, used for sampling the feasible space satisfying the constraints presented as linear inequalities. The redundant nature of the musculoskeletal system introduces variability/uncertainty in simulated quantities leading to misinterpretation of the results if ignored. Therefore, this framework provides the appropriate formalization to successfully address these issues, facilitating the application of broader types of studies in the realm of motor coordination.

The method for calculating the feasible muscles forces does not assume minimization of some objective criterion, but relies solely on the task context, the muscle model and the anatomical properties of the muscle routing. In the current derivation, muscles are treated individually, without assuming any degree of muscle synergies. An advantage of the proposed formulation is that different muscle models, including nonlinear Hill-type and muscle synergies, can be used in the calculation of the feasible muscle forces. Therefore, reduction of the feasible solution space can be further achieved by imposing different degrees of correlation between individual muscles originating from well-established muscle synergies, leading to a lower uncertainty in the estimated quantities.

While obtaining the feasible solution space is useful as it is, not all solutions of muscle forces have the same probability of occurrence. It is thus important to develop new methods that can help to explore and characterize the solution space in order to understand which region corresponds to (non-)pathological conditions. Null space solutions, although typically ignored in musculoskeletal system simulations, offer deep insights and provide a much broader framework for modeling a wide range of phenomena arising in clinical practice, such as rigidity and slack muscle disorder. This is of great importance, because if we decode and model these diseases using consistent tools originating from linear algebra, then we will be able to understand, predict and possibly negate their effect.

The CNS does not coordinate the motion of the limbs alone, but also regulates the overall stability, impedance and admittance of the musculoskeletal system through muscle co-contraction. It is thus important to examine whether the limbs' stiffness quantities can characterize the infinitely many solutions of muscle forces that give rise to the same movement. Interestingly, this kind of analysis provides useful conclusions and insights, such as the balance between stiffness capacity and task requirements, where it is evident that higher task requirements reduce the ability for stiffness modulation. Results show that the musculoskeletal system is capable of achieving a highly variable stiffness using muscle co-contraction, highlighting the importance of performing feasibility studies.

The usefulness of coordinate projection methods was explored in this dissertation, with emphasis on the modeling, simulation and analysis of redundant musculoskeletal models. Four useful representations, namely constraint, task, joint and muscle space as well as their relationships were established. This has led to the

development of innovative methods for solving a wide range of problems in the field of musculoskeletal modeling and simulation, including constraint modeling, control, planning of virtual simulations and modeling musculoskeletal redundancy.

Appendix A

Appendix

A.1 Implementation Details

The following `Python` packages were utilized `numpy`, `sympy`, `matplotlib`, `pandas` and `seaborn`; and `multipolyfit` for linear algebra manipulation, symbolic derivation, plotting, statistical analysis and polynomial fitting, respectively. The open source `OpenSim` framework was used for modeling and simulation of musculoskeletal systems. The source code along with any related material for this dissertation are publicly available (Table A.1), providing simple examples so that the readers can reproduce, understand and reuse the presented methods.

Table A.1: A summary of publicly developed materials related to the dissertation.

Description	URL
An open-source biomechanical eye model	`https://simtk.org/projects/eye`
ACL reconstruction DSS	`https://simtk.org/home/multikneedss`
Constrained task space projection	`https://simtk.org/projects/task-space`
Modeling kinematic and dynamic redundancy	`https://simtk.org/projects/redundancy`
Stiffness modulation of redundant systems	`https://simtk.org/projects/stiffness`

A.2 Fundamental Theorems

In this section, we provide some basic linear algebra background for the sake of completeness. These results are constantly referenced in the text, and they are the basis for proving some of the main results. Let $\boldsymbol{A}$ be an $m \times n$ matrix. $\boldsymbol{A}$ is a fat matrix if $m < n$, a tall matrix if $m > n$ and a square matrix if $m = n$.

Theorem A.1. *[The Fundamental Theorem]*

(a) For any system of equations, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, only one of the following three possibilities hold:

(i) The system has no solutions.

(ii) The system has a unique solution.

(iii) The system has infinite solutions.

(b) For a homogeneous system of equations ($\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$), only one of the following two possibilities hold:

(i) The system has a unique solution.

(ii) The system has infinite solutions.

When a system of equations has at least one solution, we say that the system is *consistent*, otherwise *inconsistent*. The following relate the rank of the matrix to its dimensions.

Proposition A.1. *[Matrix Rank]*
Let $\boldsymbol{A}$ be an $m \times n$ matrix. Then rank$(\boldsymbol{A}) \leq \min(m, n)$.

Definition A.1. *[Full Rank]*
Let $\boldsymbol{A}$ be an $m \times n$ matrix.

(a) $\boldsymbol{A}$ has full row rank if rank$(\boldsymbol{A}) = m$.

(b) $\boldsymbol{A}$ has full column rank if rank$(\boldsymbol{A}) = n$

The following propositions define the relationship between the existence of a solution and the full row rank property, as well as the uniqueness of a solution and the full column rank property.

Proposition A.2. *[Existence of Solutions-Full Row Rank]*
Let $\boldsymbol{A}$ be an $m \times n$ matrix. Then the system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is consistent for every $\boldsymbol{b} \in \Re^m$ iff $\boldsymbol{A}$ has full row rank (rank$(\boldsymbol{A}) = m$).

Proposition A.3. *[Uniqueness of Solutions-Full Column Rank]*
Let $\boldsymbol{A}$ be an $m \times n$ matrix. Then the system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has at most one solution for every $\boldsymbol{b} \in \Re^m$ iff $\boldsymbol{A}$ has full column rank (rank$(\boldsymbol{A}) = n$).

Full row rank implies existence and full column rank uniqueness. We can present a summary of properties based on the type of matrix.

Proposition A.4. *[Fat Matrix]*
Let $\boldsymbol{A}$ be an $m \times n$ fat matrix ($m < n$). Then the following assertions hold:

(a) rank$(\boldsymbol{A}) \leq m$.

(b) The homogeneous system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ has infinite solutions.

(c) The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is either inconsistent or has infinite solutions (i.e. it never has a unique solution).

(d) The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *is consistent* $\forall \boldsymbol{b} \in \Re^m$ *iff* $rank(\boldsymbol{A}) = m$.

Proposition A.5. *[Tall Matrix]*
Let $\boldsymbol{A}$ *be an* $m \times n$ *tall matrix* $(m > n)$. *Then the following assertions hold:*

(a) $rank(\boldsymbol{A}) \leq n$.

(b) The homogeneous system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ *has a unique (trivial) solution iff* $rank(\boldsymbol{A}) = n$ *(i.e. it has infinite solutions iff* $rank(\boldsymbol{A}) < n$*).*

(c) The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *is inconsistent for some* $\boldsymbol{b} \in \Re^m$ *(i.e. the system is never consistent* $\forall \boldsymbol{b} \in \Re^m$*).*

(d) The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *has at most one solution* $\forall \boldsymbol{b} \in \Re^m$ *iff* $rank(\boldsymbol{A}) = n$.

Proposition A.6. *[Square Matrix]*
Let $\boldsymbol{A}$ *be an* $n \times n$ *square matrix. Then the following assertions hold:*

(a) $rank(\boldsymbol{A}) \leq n$.

(b) The following are equivalent:

(i) $rank(\boldsymbol{A}) = n$.

(ii) $\boldsymbol{A}$ *is invertible.*

(iii) The homogeneous system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ *has a unique (trivial) solution.*

(iv) The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *has a unique solution* $\forall \boldsymbol{b} \in \Re^n$.

(v) The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *is consistent* $\forall \boldsymbol{b} \in \Re^n$.

(vi) $\det \boldsymbol{A} \neq 0$.

(c) According to part (b) the following are equivalent.

(i) $rank(\boldsymbol{A}) < n$.

(ii) $\boldsymbol{A}$ *is not invertible.*

(iii) The homogeneous system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ *has infinite solutions.*

(iv) The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *has infinite solutions for some* $\boldsymbol{b} \in \Re^n$.

(v) The system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *is inconsistent for some* $\boldsymbol{b} \in \Re^n$.

(vi) $\det \boldsymbol{A} = 0$.

Finally, Definition A.2 summarizes some useful properties of the MPP matrix.

Definition A.2. *[Moore-Penrose Pseudoinverse]*
For a matrix $\boldsymbol{A} \in \Re^{m \times n}$, *the MPP of* $\boldsymbol{A}$ *is defined as the matrix* $\boldsymbol{A}^+ \in \Re^{n \times m}$, *satisfying all the following:*

(a) $\boldsymbol{A}\boldsymbol{A}^{+}\boldsymbol{A} = \boldsymbol{A}$

(b) $\boldsymbol{A}^{+}\boldsymbol{A}\boldsymbol{A}^{+} = \boldsymbol{A}^{+}$

(c) $(\boldsymbol{A}\boldsymbol{A}^{+})^{T} = \boldsymbol{A}\boldsymbol{A}^{+}$

(d) $(\boldsymbol{A}^{+}\boldsymbol{A})^{T} = \boldsymbol{A}^{+}\boldsymbol{A}$

(e) If $\boldsymbol{A}$ *has full column rank, then* $\boldsymbol{A}^{+} = (\boldsymbol{A}^{T}\boldsymbol{A})^{-1}\boldsymbol{A}^{T}$ *(left inverse) and* $\boldsymbol{A}^{+}\boldsymbol{A} = \boldsymbol{I}$.

(f) If $\boldsymbol{A}$ *has full row rank, then* $\boldsymbol{A}^{+} = \boldsymbol{A}^{T}(\boldsymbol{A}\boldsymbol{A}^{T})^{-1}$ *(right inverse) and* $\boldsymbol{A}\boldsymbol{A}^{+} = \boldsymbol{I}$.

(g) The MPP solves the "least squares" problem such that

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \geq \left\|\boldsymbol{A}\boldsymbol{x}_p - \boldsymbol{b}\right\|_2, \ \forall \boldsymbol{x} \in \Re^n, \ \boldsymbol{x}_p = \boldsymbol{A}^{+}\boldsymbol{b}.$$

A.3 Tensor-Vector Product

A derivative of a matrix with respect to a vector yields a rank-3 tensor (essentially a 3D matrix). As these products emerge in the derivations, it is important to follow a consistent notation.

Definition A.3. *[n-model product]*

The n-model product $\bullet_n$ *of a tensor (Kolda and Bader, 2009)* $\boldsymbol{\mathcal{T}} \in \Re^{I_1 \times I_2 \times \cdots \times I_N}$ *and a vector* $\boldsymbol{v} \in \Re^{I_n}$ *produces a tensor of order* $N-1$ *with size* $I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N$ *such as*

$$(\boldsymbol{\mathcal{T}} \bullet_n \boldsymbol{v})_{i_1,\ldots,i_{n-1},i_{n+1},\ldots i_N} = \sum_{i_n}^{I_n} \boldsymbol{\mathcal{T}}_{i_1,i_2,\ldots,i_N} \boldsymbol{v}_{i_n}. \tag{A.1}$$

For the sake of completeness lets present an example

$$\boldsymbol{T}_{:,:,0} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \boldsymbol{T}_{:,:,1} = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix} \boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$$\boldsymbol{\mathcal{T}} \bullet_2 \boldsymbol{v} = \begin{bmatrix} 70 & 190 \\ 80 & 200 \\ 90 & 210 \end{bmatrix}$$

A.4 Simple Arm Model

The simplified arm model (Figure A.1) has three DoFs and nine muscles, some of them being bi-articular. The analytical expressions of the EoMs is given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau}_g(\boldsymbol{q}) = \boldsymbol{\tau} \tag{A.2}$$

where $\boldsymbol{M} \in \Re^{n \times n}$ represents the inertia mass matrix, n the DoFs of the model, $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \Re^n$ the generalized coordinates and their derivatives, $\boldsymbol{C} \in \Re^{n \times n}$ the Coriolis and centrifugal matrix, $\boldsymbol{\tau}_g \in \Re^n$ the gravity contribution and $\boldsymbol{\tau}$ the specified generalized forces.

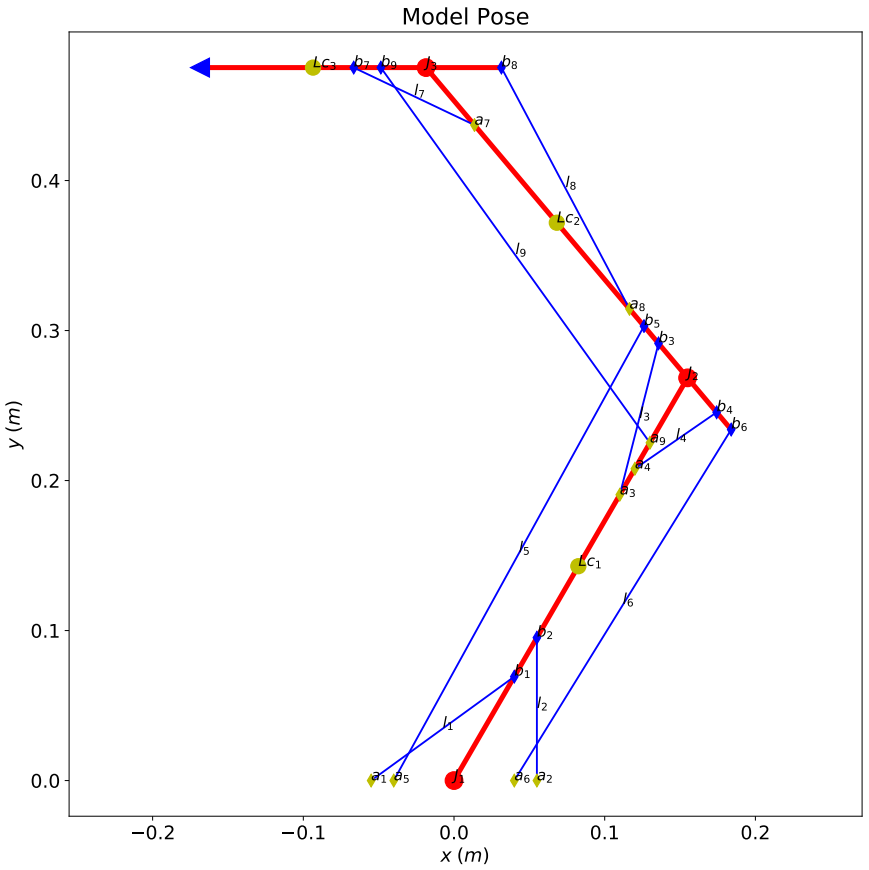


Figure A.1: Diagram of the simplified arm model, which has three DoFs and nine muscles, some of them being bi-articular. The muscle origins are labeled as a_i and the muscle insertions as b_i. l_i stands for muscle length, Lc_i for center of mass and J_i for joint center.

As the model is an open kinematic chain a simple procedure to derive the EoMs can be followed. Assuming that the spatial velocity (translational, rotational) of each body segment is given by $\boldsymbol{v}_b = [\boldsymbol{u}, \boldsymbol{\omega}]^T \in \Re^{6 \times 1}$, the Kinetic Energy (KE) of

the system in body local coordinates is defined as

$$K = \frac{1}{2}\sum_{i=1}^{n_b}(m_i \boldsymbol{u}_i^2 + \boldsymbol{I}_i \boldsymbol{\omega}_i^2) = \frac{1}{2}\sum_{i=1}^{n_b} \boldsymbol{v}_i^T \boldsymbol{M}_i \boldsymbol{v}_i \tag{A.3}$$

where $\boldsymbol{M}_i = diag(m_i, m_i, m_i, [\boldsymbol{I}_i]_{3\times3}) \in \Re^{6\times6}$ denotes the spatial inertia mass matrix, m_i the mass and $\boldsymbol{I}_i \in \Re^{3\times3}$ the inertia matrix of body i. The spatial quantities are related to the generalized coordinates by the body Jacobian $\boldsymbol{v}_b = \boldsymbol{J}_b\dot{\boldsymbol{q}},\ \boldsymbol{J}_b \in \Re^{6\times n}$. The total KE is coordinate invariant, thus it can be expressed in a different coordinate system

$$K = \frac{1}{2}\sum_{i=1}^{n_b} \boldsymbol{q}^T \boldsymbol{J}_i^T \boldsymbol{M}_i \boldsymbol{J}_i \boldsymbol{q}. \tag{A.4}$$

Following the above definition, the inertia mass matrix of the system can be written as

$$\boldsymbol{M}(\boldsymbol{q}) = \sum_{i=1}^{n_b} \boldsymbol{J}_i^T \boldsymbol{M}_i \boldsymbol{J}_i. \tag{A.5}$$

Furthermore, the Coriolis and centrifugal forces $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ can be determined directly from the inertia mass matrix

$$\boldsymbol{C}_{ij}(\boldsymbol{q},\ \dot{\boldsymbol{q}}) = \sum_{k=1}^{n} \boldsymbol{\Gamma}_{ijk}\ \dot{q}_k,\ i,\ j \in [1, \dots n],\ \boldsymbol{\Gamma}_{ijk} = \frac{1}{2}\left(\frac{\partial \boldsymbol{M}_{ij}(\boldsymbol{q})}{\partial q_k} + \frac{\partial \boldsymbol{M}_{ik}(\boldsymbol{q})}{\partial q_j} - \frac{\partial \boldsymbol{M}_{kj}(\boldsymbol{q})}{\partial q_i}\right) \tag{A.6}$$

where the functions $\boldsymbol{\Gamma}_{ijk}$ are called the Christoffel symbols. The gravity contribution can be determined from the Potential Energy (PE) function

$$\boldsymbol{\tau}_g(\boldsymbol{q}) = \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}},\ V(\boldsymbol{q}) = \sum_{i=1}^{n_b} m_i g h_i(\boldsymbol{q}) \tag{A.7}$$

where $h_i(\boldsymbol{q})$ denotes the vertical displacement of body i with respect to the ground. In this derivation we chose to collect all forces that act on the system in the term $\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}})$.

The muscle forces $\boldsymbol{f}_m$ are transformed into joint space generalized forces ($\boldsymbol{\tau}$) by the moment arm matrix ($\boldsymbol{\tau} = -\boldsymbol{R}^T \boldsymbol{f}_m$). For a n-lateral polygon it can be shown that the derivative of the side length with respect to the opposite angle is the moment arm component. As a consequence, when expressing the muscle length as a function of the generalized coordinates of the model, the moment arm matrix is evaluated by $\boldsymbol{R} = \partial \boldsymbol{l}_m(\boldsymbol{q})/\partial \boldsymbol{q}$. The analytical expressions of the EoMs following our convention are provided below (Equation A.8). The mass properties of the model are summarized in Table A.2 and the muscle parameters in Table A.3 ((Tahara et al., 2006)).

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \boldsymbol{\tau} \\ \boldsymbol{\tau} &= -\boldsymbol{R}^T(\boldsymbol{q})\boldsymbol{f}_m \end{aligned} \tag{A.8}$$

$$\begin{aligned} M_{1,1} = {} & Iz_1 + Iz_2 + Iz_3 + L_1^2 m_2 + L_1^2 m_3 + 2L_1 L_2 m_3 \cos(q_2(t)) \\ & + 2L_1 Lc_2 m_2 \cos(q_2(t)) + 2L_1 Lc_3 m_3 \cos(q_2(t) + q_3(t)) \\ & + L_2^2 m_3 + 2L_2 Lc_3 m_3 \cos(q_3(t)) + Lc_1^2 m_1 + Lc_2^2 m_2 + Lc_3^2 m_3 \end{aligned} \tag{A.9}$$

$$\begin{aligned} M_{1,2} = {} & Iz_2 + Iz_3 + L_1 L_2 m_3 \cos(q_2(t)) + L_1 Lc_2 m_2 \cos(q_2(t)) \\ & + L_1 Lc_3 m_3 \cos(q_2(t) + q_3(t)) + L_2^2 m_3 \\ & + 2L_2 Lc_3 m_3 \cos(q_3(t)) + Lc_2^2 m_2 + Lc_3^2 m_3 \end{aligned} \tag{A.10}$$

$$M_{1,3} = Iz_3 + L_1 Lc_3 m_3 \cos(q_2(t) + q_3(t)) + L_2 Lc_3 m_3 \cos(q_3(t)) + Lc_3^2 m_3 \tag{A.11}$$

$$\begin{aligned} M_{2,1} = {} & Iz_2 + Iz_3 + L_1 L_2 m_3 \cos(q_2(t)) + L_1 Lc_2 m_2 \cos(q_2(t)) \\ & + L_1 Lc_3 m_3 \cos(q_2(t) + q_3(t)) + L_2^2 m_3 \\ & + 2L_2 Lc_3 m_3 \cos(q_3(t)) + Lc_2^2 m_2 + Lc_3^2 m_3 \end{aligned} \tag{A.12}$$

$$M_{2,2} = Iz_2 + Iz_3 + L_2^2 m_3 + 2L_2 Lc_3 m_3 \cos(q_3(t)) + Lc_2^2 m_2 + Lc_3^2 m_3 \tag{A.13}$$

$$M_{2,3} = Iz_3 + L_2 Lc_3 m_3 \cos(q_3(t)) + Lc_3^2 m_3 \tag{A.14}$$

$$M_{3,1} = Iz_3 + L_1 Lc_3 m_3 \cos(q_2(t) + q_3(t)) + L_2 Lc_3 m_3 \cos(q_3(t)) + Lc_3^2 m_3 \tag{A.15}$$

$$M_{3,2} = Iz_3 + L_2 Lc_3 m_3 \cos(q_3(t)) + Lc_3^2 m_3 \tag{A.16}$$

$$M_{3,3} = Iz_3 + Lc_3^2 m_3 \tag{A.17}$$

$$\begin{aligned} f_1 = {} & Lc_1 g m_1 \cos(q_1(t)) + g m_2 (L_1 \cos(q_1(t)) + Lc_2 \cos(q_1(t) + q_2(t))) \\ & + g m_3 (L_1 \cos(q_1(t)) + L_2 \cos(q_1(t) + q_2(t)) + Lc_3 \cos(q_1(t) + q_2(t) + q_3(t))) \\ & - 2L_1 L_2 m_3 \sin(q_2(t)) \frac{d}{dt} q_1(t) \frac{d}{dt} q_2(t) + L_1 L_2 m_3 \sin(q_2(t)) \left(\frac{d}{dt} q_2(t) \right)^2 \\ & + 2L_1 Lc_2 m_2 \sin(q_2(t)) \frac{d}{dt} q_1(t) \frac{d}{dt} q_2(t) + L_1 Lc_2 m_2 \sin(q_2(t)) \left(\frac{d}{dt} q_2(t) \right)^2 \\ & + 2L_1 Lc_3 m_3 \sin(q_2(t) + q_3(t)) \frac{d}{dt} q_1(t) \frac{d}{dt} q_2(t) \\ & + 2L_1 Lc_3 m_3 \sin(q_2(t) + q_3(t)) \frac{d}{dt} q_1(t) \frac{d}{dt} q_3(t) \\ & + L_1 Lc_3 m_3 \sin(q_2(t) + q_3(t)) \left(\frac{d}{dt} q_2(t) \right)^2 \\ & + 2L_1 Lc_3 m_3 \sin(q_2(t) + q_3(t)) \frac{d}{dt} q_2(t) \frac{d}{dt} q_3(t) \\ & + L_1 Lc_3 m_3 \sin(q_2(t) + q_3(t)) \left(\frac{d}{dt} q_3(t) \right)^2 + 2L_2 Lc_3 m_3 \sin(q_3(t)) \frac{d}{dt} q_1(t) \frac{d}{dt} q_3(t) \\ & + 2L_2 Lc_3 m_3 \sin(q_3(t)) \frac{d}{dt} q_2(t) \frac{d}{dt} q_3(t) + L_2 Lc_3 m_3 \sin(q_3(t)) \left(\frac{d}{dt} q_3(t) \right)^2 \end{aligned} \tag{A.18}$$

$$
\begin{aligned}
f_2 = & -L_2 L c_3 m_3 \left(\frac{d}{dt}q_1(t) + \frac{d}{dt}q_2(t) + \frac{d}{dt}q_3(t)\right)\sin\left(q_3(t)\right)\frac{d}{dt}q_3(t) \\
& - L_2 L c_3 m_3 \sin\left(q_3(t)\right)\frac{d}{dt}q_2(t)\frac{d}{dt}q_3(t) + L c_2 g m_2 \cos\left(q_1(t) + q_2(t)\right) \\
& + g m_3 \left(L_2 \cos\left(q_1(t) + q_2(t)\right) + L c_3 \cos\left(q_1(t) + q_2(t) + q_3(t)\right)\right) \\
& + \left(L_1 \left(L_2 m_3 \sin\left(q_2(t)\right) + L c_2 m_2 \sin\left(q_2(t)\right) + L c_3 m_3 \sin\left(q_2(t) + q_3(t)\right)\right)\frac{d}{dt}q_1(t)\right. \\
& \left. - L_2 L c_3 m_3 \sin\left(q_3(t)\right)\frac{d}{dt}q_3(t)\right)\frac{d}{dt}q_1(t)
\end{aligned} \tag{A.19}
$$

$$
\begin{aligned}
f_3 = & L c_3 g m_3 \cos\left(q_1(t) + q_2(t) + q_3(t)\right) \\
& + L c_3 m_3 \left(L_1 \sin\left(q_2(t) + q_3(t)\right)\left(\frac{d}{dt}q_1(t)\right)^2 + L_2 \sin\left(q_3(t)\right)\left(\frac{d}{dt}q_1(t)\right)^2\right. \\
& \left. + 2 L_2 \sin\left(q_3(t)\right)\frac{d}{dt}q_1(t)\frac{d}{dt}q_2(t) + L_2 \sin\left(q_3(t)\right)\left(\frac{d}{dt}q_2(t)\right)^2\right)
\end{aligned} \tag{A.20}
$$

$$
R_{1,1} = -\frac{a_1 b_1 \sin\left(q_1(t)\right)}{\sqrt{a_1^2 + 2 a_1 b_1 \cos\left(q_1(t)\right) + b_1^2}} \tag{A.21}
$$

$$
R_{1,2} = 0 \tag{A.22}
$$

$$
R_{1,3} = 0 \tag{A.23}
$$

$$
R_{2,1} = \frac{a_2 b_2 \sin\left(q_1(t)\right)}{\sqrt{a_2^2 - 2 a_2 b_2 \cos\left(q_1(t)\right) + b_2^2}} \tag{A.24}
$$

$$
R_{2,2} = 0 \tag{A.25}
$$

$$
R_{2,3} = 0 \tag{A.26}
$$

$$
R_{3,1} = 0 \tag{A.27}
$$

$$
R_{3,2} = -\frac{b_3 \left(2 L_1 - 2 a_3\right)\sin\left(q_2(t)\right)}{2\sqrt{b_3^2 + b_3 \left(2 L_1 - 2 a_3\right)\cos\left(q_2(t)\right) + \left(L_1 - a_3\right)^2}} \tag{A.28}
$$

$$
R_{3,3} = 0 \tag{A.29}
$$

$$
R_{4,1} = 0 \tag{A.30}
$$

$$
R_{4,2} = \frac{b_4 \left(2 L_1 - 2 a_4\right)\sin\left(q_2(t)\right)}{2\sqrt{b_4^2 - b_4 \left(2 L_1 - 2 a_4\right)\cos\left(q_2(t)\right) + \left(L_1 - a_4\right)^2}} \tag{A.31}
$$

$$
R_{4,3} = 0 \tag{A.32}
$$

$$
R_{5,1} = -\frac{a_5 \left(L_1 \sin\left(q_1(t)\right) + b_5 \sin\left(q_1(t) + q_2(t)\right)\right)}{\sqrt{L_1^2 + 2 L_1 a_5 \cos\left(q_1(t)\right) + 2 L_1 b_5 \cos\left(q_2(t)\right) + a_5^2 + 2 a_5 b_5 \cos\left(q_1(t) + q_2(t)\right) + b_5^2}} \tag{A.33}
$$

$$
R_{5,2} = -\frac{b_5 \left(L_1 \sin\left(q_2(t)\right) + a_5 \sin\left(q_1(t) + q_2(t)\right)\right)}{\sqrt{L_1^2 + 2 L_1 a_5 \cos\left(q_1(t)\right) + 2 L_1 b_5 \cos\left(q_2(t)\right) + a_5^2 + 2 a_5 b_5 \cos\left(q_1(t) + q_2(t)\right) + b_5^2}} \tag{A.34}
$$

$$
R_{5,3} = 0 \tag{A.35}
$$

$$R_{6,1} = \frac{a_6 \left(L_1 \sin\left(q_1(t)\right) - b_6 \sin\left(q_1(t) + q_2(t)\right)\right)}{\sqrt{L_1^2 - 2L_1 a_6 \cos\left(q_1(t)\right) - 2L_1 b_6 \cos\left(q_2(t)\right) + a_6^2 + 2a_6 b_6 \cos\left(q_1(t) + q_2(t)\right) + b_6^2}} \tag{A.36}$$

$$R_{6,2} = \frac{b_6 \left(L_1 \sin\left(q_2(t)\right) - a_6 \sin\left(q_1(t) + q_2(t)\right)\right)}{\sqrt{L_1^2 - 2L_1 a_6 \cos\left(q_1(t)\right) - 2L_1 b_6 \cos\left(q_2(t)\right) + a_6^2 + 2a_6 b_6 \cos\left(q_1(t) + q_2(t)\right) + b_6^2}} \tag{A.37}$$

$$R_{6,3} = 0 \tag{A.38}$$

$$R_{7,1} = 0 \tag{A.39}$$

$$R_{7,2} = 0 \tag{A.40}$$

$$R_{7,3} = -\frac{b_7 \left(2L_2 - 2a_7\right) \sin\left(q_3(t)\right)}{2\sqrt{b_7^2 + b_7 \left(2L_2 - 2a_7\right) \cos\left(q_3(t)\right) + \left(L_2 - a_7\right)^2}} \tag{A.41}$$

$$R_{8,1} = 0 \tag{A.42}$$

$$R_{8,2} = 0 \tag{A.43}$$

$$R_{8,3} = \frac{b_8 \left(2L_2 - 2a_8\right) \sin\left(q_3(t)\right)}{2\sqrt{b_8^2 - b_8 \left(2L_2 - 2a_8\right) \cos\left(q_3(t)\right) + \left(L_2 - a_8\right)^2}} \tag{A.44}$$

$$R_{9,1} = 0 \tag{A.45}$$

$$R_{9,2} = \frac{-\frac{L_2}{2} \left(2L_1 - 2a_9\right) \sin\left(q_2(t)\right) - \frac{b_9}{2} \left(2L_1 - 2a_9\right) \sin\left(q_2(t) + q_3(t)\right)}{\sqrt{L_2^2 + 2L_2 b_9 \cos\left(q_3(t)\right) + L_2 \left(2L_1 - 2a_9\right) \cos\left(q_2(t)\right) + b_9^2 + b_9 \left(2L_1 - 2a_9\right) \cos\left(q_2(t) + q_3(t)\right) + \left(L_1 - a_9\right)^2}} \tag{A.46}$$

$$R_{9,3} = \frac{-L_2 b_9 \sin\left(q_3(t)\right) - \frac{b_9}{2} \left(2L_1 - 2a_9\right) \sin\left(q_2(t) + q_3(t)\right)}{\sqrt{L_2^2 + 2L_2 b_9 \cos\left(q_3(t)\right) + L_2 \left(2L_1 - 2a_9\right) \cos\left(q_2(t)\right) + b_9^2 + b_9 \left(2L_1 - 2a_9\right) \cos\left(q_2(t) + q_3(t)\right) + \left(L_1 - a_9\right)^2}} \tag{A.47}$$

Table A.2: Body parameters: mass, inertia, length and center of mass (CoM).

Bodies	$m\ (kg)$	$I_z\ (kg \cdot m^2)$	$L\ (m)$	$CoM\ (m)$
arm	1.93	0.0141	0.31	0.165
forearm	1.32	0.0120	0.27	0.135
hand	0.35	0.0010	0.15	0.075

Table A.3: Muscle parameters: muscle origin, insertion and maximum force.

Muscle	a (m)	b (m)	$f_{\max}$ (N)
m_1	0.055	0.080	50
m_2	0.055	0.110	50
m_3	0.220	0.030	50
m_4	0.240	0.030	50
m_5	0.040	0.045	25
m_6	0.040	0.045	25
m_7	0.220	0.048	50
m_8	0.060	0.050	50
m_9	0.260	0.030	25

A.5 List of Publications

Journals

J1 **D. Stanev**, K. Moustakas, J. Gliatis and C. Koutsojannis, ACL Reconstruction Decision Support Through Personalized Simulation of the Lachman Test and Custom Activities, **Methods of Information in Medicine**, vol. 55 no. 1, pp. 98–105, Jan. 2016, DOI: 10.3414/ME14–02–0022

J2 **D. Stanev** and K. Moustakas, Simulation of Constrained Musculoskeletal Systems in Task Space, **IEEE Transactions on Biomedical Engineering**, vol.65, no. 2, pp. 307–318, Feb. 2018, DOI: 10.1109/TBME.2017.2764630

J3 **D. Stanev** and K. Moustakas, Modeling musculoskeletal kinematic and dynamic redundancy using null space projection, **PLoS ONE**, 14(1): e0209171, Jan. 2019, DOI: https://doi.org/10.1371/journal.pone.0209171

J4 **D. Stanev** and K. Moustakas, Stiffness Modulation of Redundant Musculoskeletal Systems, **Journal of Biomechanics**, Jan. 2019, DOI: https://doi.org/10.1016/j.jbiomech.2019.01.017

Conferences

C1 **D. Stanev** and K. Moustakas, Virtual human behavioural profile extraction using Kinect based motion tracking, **International Conference on Cyberworlds**, Santander, Spain, pp. 411–414, Oct. 2014

C2 **D. Stanev** and K. Moustakas, Proprioceptive modeling of the peripheral nervous system as an extension to the biomechanics musculoskeletal models, **Greek Society for Biomedical Engineering ELEVIT**, Greece, May 2015

C3 **D. Stanev**, P. Moschonas, K. Votis, D. Tzovaras and K. Moustakas, Simulation and Visual Analysis of Neuromusculoskeletal Models and Data, **Artificial Intelligence Applications and Innovations Conference AIAI**, France, Sep. 2015, DOI: 10.1007/978–3–319–23868–5˙29

C4 **D. Stanev**, J. Gliatis, K. Moustakas and K. Constantinos, ACL reconstruction through patient specific simulation of the Lachman test, **Surgery, Knee Traumatology, Sports**, vol. 24, no. 1, pp. S423, May 2016

C5 **D. Stanev**, A. Blenkinsop, K. Gurney and K. Moustakas, Neuromusculoskeletal Inertial Filtering of Centrally Generated Beta Oscillations in Parkinson's Disease, **Virtual Physiological Human (VPH)**, Amsterdam, Sep. 2016

C6 **D. Stanev** and K. Moustakas, The Effect of Kinematic and Dynamic Redundancy on the Assessment of Joint Reaction Loads, **Virtual Physiological Human (VPH)**, Zaragoza, Sep. 2018

C7 **D. Stanev**, C. Koutsojannis and K. Moustakas, Exploring Musculoskeletal Redundancy Using Null Space Projection, **Movement and Cognition**, Tel-Aviv, Jul. 2019

Generic

G1 K. Filip, **D. Stanev** and K. Moustakas, An Open-Source OpenSim Oculomotor Model for Kinematics and Dynamics Simulation, **arXiv:1807.07332 [physics.bio-ph]**, Jul. 2018

Bibliography

Ackermann, M. and van den Bogert, A. J. (2010). Optimality principles for model-based prediction of human gait. *J. Biomech.*, 43(6):1055–1060.

Aghili, F. (2005). A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: Applications to control and simulation. *IEEE Trans. Robot.*, 21(5):834–849.

Anderson, F. C. and Pandy, M. G. (1999). A Dynamic Optimization Solution for Vertical Jumping in Three Dimensions. *Comput. Methods Biomech. Biomed. Engin.*, 2(3):201–231.

Anderson, F. C. and Pandy, M. G. (2001). Static and dynamic optimization solutions for gait are practically equivalent. *J. Biomech.*, 34(2):153–161.

Arnold, A. S., Asakawa, D. J., and Delp, S. L. (2000). Do the hamstrings and adductors contribute to excessive internal rotation of the hip in persons with cerebral palsy? *Gait Posture*, 11(3):181–190.

Avis, D. and Fukuda, K. (1992). A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra. *Discret. Comput. Geom.*, 8(1):295–313.

Babikian, S., Valero-cuevas, F. J., and Kanso, E. (2015). Bio-inspired Actuation of Slow Movements. *J. Nonlinear Sci.*, pages 1–12.

Baerlocher, P. (2001). *Inverse kinematics techniques for the interactive posture control of articulated figures*, volume 2383. EPFL.

Balasch, H., Schiller, M., Friebel, H., and Hoffmann, F. (1999). Evaluation of anterior knee joint instability with the Rolimeter. *Knee Surgery, Sport. Traumatol. Arthrosc.*, 7(4):204–208.

Baumgarte, J. (1972). Stabilization of constraints and integrals of motion in dynamical systems. *Comput. Methods Appl. Mech. Eng.*, 1(1):1–16.

Benjaminse, A., Gokeler, A., and van der Schans, C. P. (2006). Clinical diagnosis of an anterior cruciate ligament rupture: a meta-analysis. *J. Orthop. Sports Phys. Ther.*, 36(5):267–88.

Bernshtein, N. A. (1967). *The co-ordination and regulation of movements*. Oxford: Pergamon Press.

Besier, T. F., Gold, G. E., Beaupre, G. S., and Delp, S. L. (2005). A modeling framework to estimate patellofemoral joint cartilage stress in vivo. *Med. Sci. Sports Exerc.*, 37(11):1924–1930.

Blankevoort, L. and Huiskes, R. (1996). Validation of a three-dimensional model of the knee. *J. Biomech. Eng.*, 29(7):955–961.

Blankevoort, L., Kuiper, J. H., Huiskes, R., and Grootenboer, H. J. (1991). Articular contact in a three-dimensional model of the knee. *J. Biomech.*, 24(11):1019–1031.

Bloemker, K. H., Guess, T. M., Maletsky, L., and Dodd, K. (2012). Computational Knee Ligament Modeling Using Experimentally Determined Zero-Load Lengths. *J. Biomech. Eng.*, 6:33–41.

Buchanan, T. S., Lloyd, D. G., Manal, K., and Besier, T. F. (2006). Neuromusculoskeletal Modeling: Estimation of Muscle Forces and Joint Moments and Movements From Measurements of Neural Command. *J. Appl. Biomech.*, 20(4):367–395.

Campos, A., Guenther, R., and Martins, D. (2005). Differential kinematics of serial manipulators using virtual chains. *J. Brazilian Soc. Mech. Sci. Eng.*, 27(4):345–356.

Carbone, V., Fluit, R., Pellikaan, P., van der Krogt, M. M., Janssen, D., Damsgaard, M., Vigneron, L., Feilkas, T., Koopman, H., and Verdonschot, N. (2015). Tlem 2.0–A Comprehensive Musculoskeletal Geometry Dataset For Subject-Specific Modeling Of Lower Extremity. *J. Biomech.*, 48(5):1–8.

Chadwick, E. K., Blana, D., Kirsch, R. F., and van den Bogert, A. J. (2014). Real-time simulation of three-dimensional shoulder girdle and arm dynamics. *IEEE Trans. Biomed. Eng.*, 61(7):1947–1956.

Cohen, J. Y., Heitz, R. P., Woodman, G. F., and Schall, J. D. (2009). Neural basis of the set-size effect in frontal eye field: timing of attention during visual search. *J. Neurophysiol.*, 101(4):1699–1704.

Collins, C. C., Carlson, M. R., Scott, A. B., and Jampolsky, a. (1981). Extraocular muscle forces in normal human subjects. *Invest. Ophthalmol. Vis. Sci.*, 20(5):652–664.

Cui, L., Perreault, E. J., Mass, H., and Sandercock, T. (2008). Modeling Short-Range Stiffness of Feline Lower Hindlimb Muscles. *J. Biomech.*, 41(9):1945–1952.

Davoudabadi Farahani, S., Svinin, M., Andersen, M. S., de Zee, M., and Rasmussen, J. (2016). Prediction of closed-chain human arm dynamics in a crank-rotation task. *J. Biomech.*, 49(13):2684–2693.

De Groote, F., De Laet, T., Jonkers, I., De Schutter, J., Groote, F. D., Laet, T. D., Jonkers, I., and Schutter, J. D. (2008). Kalman smoothing improves the estimation of joint kinematics and kinetics in marker-based human gait analysis. *J. Biomech.*, 41(16):3390–3398.

De Groote, F., Kinney, A. L., Rao, A. V., and Fregly, B. J. (2016). Evaluation of Direct Collocation Optimal Control Problem Formulations for Solving the Muscle Redundancy Problem. *Ann. Biomed. Eng.*, 44(10):2922–2936.

De Groote, F., Pipeleers, G., Jonkers, I., Demeulenaere, B., Patten, C., Swevers, J., and De Schutter, J. (2009). A physiology based inverse dynamic analysis of humpan gait: potential and perspectives. *Comput. Methods Biomech. Biomed. Eng.*, 12(March 2015):563–574.

De Sapio, V., Khatib, O., and Delp, S. L. (2006). Task-level approaches for the control of constrained multibody systems. *Multibody Syst. Dyn.*, 16(1):73–102.

De Sapio, V. and Park, J. (2010). Multitask Constrained Motion Control Using a Mass-Weighted Orthogonal Decomposition. *J. Appl. Mech.*, 77(4):1–9.

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P. L., 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 Trans. Biomed. Eng.*, 54(11):1940–1950.

Delp, S. L. and Loan, P. J. (2000). A Computational Framework for Simulating and Analyzing Human and Animal Movement. In *Comput. Sci. Eng.*, pages 46–55. IEEE.

Delp, S. L., Loan, P. J., Hoy, M. G., Zajac, F. E., Topp, E. L., and Rosen, J. M. (1990). An interactive graphics-based model of the lower extremity to study orthopedic surgical procedures. *IEEE Trans. Biomed. Eng.*, 37(8):757–767.

Delp, S. L. and Loan, P. L. (1995). A graphics-based software system to develop and analyze models of musculoskeletal structures. *Comput. Biol. Med.*, 25(1):21–34.

Dempster, W. T. (1955). The Anthropometry of Body Action. *Ann. N. Y. Acad. Sci.*, 63(4):559–585.

Deshpande, A. D., Balasubramanian, R., Ko, J., and Matsuoka, Y. (2010). Acquiring variable moment arms for index finger using a robotic testbed. *IEEE Trans. Biomed. Eng.*, 57(8):2034–2044.

DeWolf, T. and Eliasmith, C. (2011). The neural optimal control hierarchy for motor control. *J. Neural Eng.*, 8(6):1–21.

Dideriksen, J. L., Enoka, R. M., Farina, D., and Member, S. (2011). A Model of the Surface Electromyogram in Pathological Tremor. *IEEE Trans. Biomed. Eng.*, 58(8):2178–2185.

Drillis, R., Contini, R., and Bluestein, M. (1964). Body Segment Parameters; A Survey of Mmeasurement Techniques. *Artif. Limbs*, 8(2):44–66.

Durandau, G., Farina, D., and Sartori, M. (2017). Real-time musculoskeletal modeling driven by electromyograms. *IEEE Trans. Biomed. Eng.*, PP(99):1–8.

Dziewiecki, K., Blajer, W., Mazur, Z., and Czaplicki, A. (2014). Modeling and computational issues in the inverse dynamics simulation of triple jump. *Multibody Syst. Dyn.*, 32(3):299–316.

Eich, E. (1993). Convergence Results for a Coordinate Projection Method Applied to Mechanical Systems with Algebraic Constraints. *SIAM J. Numer. Anal.*, 30(5):1467–1482.

Erdemir, A., Guess, T. M., Halloran, J. P., Modenese, L., Reinbolt, J. A., Thelen, D. G., and Umberger, B. R. (2016). Commentary on the Integration of Model Sharing and Reproducibility Analysis to Scholarly Publishing Workflow in Computational Biomechanics. *IEEE Trans. Biomed. Eng.*, 63(10):2080–2085.

Erdemir, A., McLean, S. G., Herzog, W., and Van Den Bogert, A. J. (2007). Model-based estimation of muscle forces exerted during movements. *Clin. Biomech.*, 22(2):131–154.

Fardoun, H. M., Mashat, A. S., and Lange, B. (2015). New Methodologies for Patients Rehabilitation. *Methods Inf Med*, 54(02):111–113.

Featherstone, R. (2007). *Rigid Body Dynamics Algorithms*. Springer.

Feldman, A. G. (1986). Once More on the Equilibrium-Point Hypothesis (λ Model) for Motor Control. *J. Mot. Behav.*, 18(1):17–54.

Ferber, R., Osis, S. T., Hicks, J. L., and Delp, S. L. (2016). Gait biomechanics in the era of data science. *J. Biomech.*, 49(16):3759–3761.

Filip, K., Stanev, D., and Moustakas, K. (2018). An Open-Source OpenSim Oculomotor Model for Kinematics and Dynamics Simulation.

Fisher, W. and Mujtaba, S. (1991). Hybrid Position/Force Control: A Correct Formulation. *Int. J. Rob. Res.*, 11(4):299–311.

Flash, T. (1987). The control of hand equilibrium trajectories in multi-joint arm movements. *Biol. Cybern.*, 57(4-5):257–274.

Flash, T. and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *J. Neurosci.*, 5(7):1688–1703.

Flash, T. and Mussa-Ivaldi, F. A. (1990). Human arm stiffness characteristics during the maintenance of posture. *Exp. Brain Res.*, 82(2):315–26.

Fregly, B. J., Besier, T. F., Lloyd, D. G., Delp, S. L., Banks, S. a., Pandy, M. G., and D'Lima, D. D. (2011). Grand challenge competition to predict in vivo knee loads. *J. Orthop. Res.*, 30(4):503–513.

Ganko, A., Engebretsen, L., and Ozer, H. (2000). The Rolimeter: a new arthrometer compared with the KT-1000. *Knee Surgery, Sport. Traumatol. Arthrosc.*, 8(1):36–39.

Hamner, S. R., Seth, A., and Delp, S. L. (2010). Muscle contributions to propulsion and support during running. *J. Biomech.*, 43(14):2709–16.

Hatcher, J., Hatcher, A., Arbuthnot, J., and McNickolas, M. (2005). An investigation to examine the inter-tester and intra-tester reliability of the Rolimeter knee tester, and its sensitivity in identifying knee joint laxity. *J. Orthop. Res.*, 23(6):1399–1403.

Hicks, J. L., Delp, S. L., and Schwartz, M. H. (2011). Can biomechanical variables predict improvement in crouch gait? *Gait Posture*, 34(2):197–201.

Hicks, J. L., Uchida, T. K., Seth, A., Rajagopal, A., and Delp, S. L. (2014). Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of human movement. *J. Biomech. Eng.*, 137(February):1–14.

Hogan, N. (1981). The Mechanics of Multi-Joint Posture and Movement Control. *Biol. Cybern.*, 52:315–331.

Hogan, N. (1984). Adaptive Control of Mechanical Impedance by Coactivation of Antagonist Muscles. *IEEE Trans. Automat. Contr.*, 29(8):681–690.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642.

Houk, J. C. and Rymer, W. Z. (2011). Neural control of muscle length and tension. In *Handb. Physiol. Nerv. Syst. Mot. Control*, pages 257–323. John Wiley & Sons, Inc.

Hu, X., Murray, W. M., and Perreault, E. J. (2011). Muscle short-range stiffness can be used to estimate the endpoint stiffness of the human arm. *J. Neurophysiol.*, 105(4):1633–1641.

Huxley, A. F. (1974). Muscular contraction. *J. Physiol.*, 243(1):1–43.

Inouye, J. M., Kutch, J. J., and Valero-Cuevas, F. J. (2012). A Novel Synthesis of Computational Approaches Enables Optimization of Grasp Quality of Tendon-Driven Hands. *IEEE Trans. Robot.*, 28(4):958–966.

Inouye, J. M. and Valero-Cuevas, F. J. (2016). Muscle Synergies Heavily Influence the Neural Control of Arm Endpoint Stiffness and Energy Consumption. *PLoS Comput. Biol.*, 12(2):1–24.

Iskander, J., Hossny, M., Nahavandi, S., and del Porto, L. (2018). An ocular biomechanic model for dynamic simulation of different eye movements. *J. Biomech.*

Jain, A. (2011). *Robot and Multibody Dynamics*. Springer.

Kar, J. and Quesada, P. M. (2013). A musculoskeletal modeling approach for estimating anterior cruciate ligament strains and knee anterior-posterior shear forces in stop-jumps performed by young recreational female athletes. *Ann. Biomed. Eng.*, 41(2):338–348.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Curr. Opin. Neurobiol.*, 9(6):718–727.

Khatib, O. (1995). Inertial Properties in Robotic Manipulation: An Object-Level Framework. *Int. J. Rob. Res.*, 14(1):19–36.

Khatib, O., Demircan, E., De Sapio, V., Sentis, L., Besier, T. F., and Delp, S. L. (2009). Robotics-based synthesis of human motion. *J. Physiol.*, 103(3-5):211–219.

Khatib, O., Sentis, L., Park, J., and Warren, J. (2004). Whole-Body Dynamic Behavior and Control of Human-Like Robots. *Int. J. Humanoid Robot.*, 1(1):29–43.

Koeppen, R., Sternad, D., Huber, M. E., and Hogan, N. (2017). Controlling Physical Interactions : Humans do not Minimize Muscle Effort. In *Proc. ASME 2017 Dyn. Syst. Control Conf.*, pages 1–8.

Kolda, T. G. and Bader, B. W. (2009). Tensor Decompositions and Applications. *SIAM Rev.*, 51(3):455–500.

Kutch, J. J. and Valero-Cuevas, F. J. (2011). Muscle redundancy does not imply robustness to muscle dysfunction. *J. Biomech.*, 44(7):1264–1270.

Kutch, J. J. and Valero-Cuevas, F. J. (2012). Challenges and new approaches to proving the existence of muscle synergies of neural origin. *PLoS Comput. Biol.*, 8(5):1–11.

Latash, M. L., Levin, M. F., Scholz, J. P., and Schoner, G. (2010). Motor Control Theories and Their Applications. *Med.*, 46(6):382–92.

Latash, M. L., Scholz, J. P., and Schoner, G. (2002). Motor Control Strategies Revealed in the Structure of Motor Variability. *Exerc. Sport Sci. Rev.*, 30(1):26–31.

Laz, P. J. and Browne, M. (2010). A review of probabilistic analysis in orthopaedic biomechanics. *Proc. Inst. Mech. Eng. H.*, 224(8):927–943.

Le Cavorzin, P., Carrault, G., Chagneau, F., Rochcongar, P., and Allain, H. (2003). A computer model of rigidity and related motor dysfunction in Parkinson's disease. *Mov. Disord.*, 18(11):1257–65.

Lee, L.-F. and Umberger, B. R. (2016). Generating optimal control simulations of musculoskeletal movement using OpenSim and MATLAB. *PeerJ*, 4:e1638.

Levin, S., de Solorzano, S. L., Scarr, G., de Solorzano, S. L., and Scarr, G. (2017). The significance of closed kinematic chains to biological movement and dynamic stability. *J. Bodyw. Mov. Ther.*, 21(July):664–672.

Liu, F., Yue, B., Gadikota, H. R., Kozanek, M., Liu, W., Gill, T. J., Rubash, H. E., and Li, G. (2010). Morphology of the medial collateral ligament of the knee. *J. Orthop. Surg. Res.*, 5(1):69.

Loeb, G. E. (2001). Learning from the spinal cord. *J. Physiol.*, 533(1):111–117.

Loeb, G. E., Levine, W. S., and He, B. J. (1990). Understanding sensorimotor feedback through optimal control. *Cold Spring Harb. Symp. Quant. Biol.*, 55(1):791–803.

Mileusnic, M. P., Brown, I. E., Lan, N., and Loeb, G. E. (2006). Mathematical models of proprioceptors. I. Control and transduction in the muscle spindle. *J. Neurophysiol.*, 96(4):1772–88.

Mileusnic, M. P. and Loeb, G. E. (2006). Mathematical models of proprioceptors. II. Structure and function of the Golgi tendon organ. *J. Neurophysiol.*, 96(4):1789–802.

Millard, M., Uchida, T., Seth, A., and Delp, S. L. (2013). Flexing computational muscle: modeling and simulation of musculotendon dynamics. *J. Biomech. Eng.*, 135(2):1–12.

Mistry, M. and Righetti, L. (2011). Operational Space Control of Constrained and Underactuated Systems. In *Proc. Robot. Sci. Syst.*, pages 225 – 232.

Mootanah, R., Imhauser, C. W., Reisse, F., Carpanen, D., Walker, R. W., Koff, M. F., Lenhoff, M. W., Rozbruch, S. R., Fragomen, A. T., Dewan, Z., Kirane, Y. M., Cheah, K., Dowell, J. K., and Hillstrom, H. J. (2014). Development and validation of a computational model of the knee joint for the evaluation of surgical treatments for osteoarthritis. *Comput. Methods Biomech. Biomed. Engin.*, 17(13):1502–17.

Muellner, T., Bugge, W., Johansen, S., Holtan, C., and Engebretsen, L. (2001). Inter- and intratester comparison of the Rolimeter knee tester: effect of tester's experience and the examination technique. *Knee Surgery, Sport. Traumatol. Arthrosc.*, 9(5):302–306.

Mussa-Ivaldi, F. A., Hogan, N., and Bizzi, E. (1985). Neural, mechanical, and geometrical factors subserving arm posture in humans. *J. Neurosci.*, 5(10):2732–2743.

Myers, C. A., Shelburne, K. B., Laz, P. J., and Davidson, B. S. (2015). A Probabilistic Approach to Quantify the Impact of Uncertainty Propagation in Musculoskeletal Simulations. *Ann. Biomed. Eng.*, 43(5):1098–1111.

Nielson, G. M. (2004). v-Quaternion Splines for the Smooth Interpolation of Orientations. *IEEE Trans. Vis. Comput. Graph.*, 10(2):224–229.

Ong, C., Hicks, J., and Delp, S. L. (2016). Simulation-Based Design for Wearable Robotic Systems: An Optimization Framework for Enhancing a Standing Long Jump. *IEEE Trans. Biomed. Eng.*, 63(5):894–903.

Pandy, M. G. (2001). Computer Modeling and Simulation of Human Movement. *Ann. Biomed. Eng.*, 3:245–73.

Pedotti, A., Krishnan, V. V., and Stark, L. (1978). Optimization of muscle-force sequencing in human locomotion. *Math. Biosci.*, 38(1-2):57–76.

Perreault, E. J., Kirsch, R. F., and Crago, P. E. (2001). Effects of voluntary force generation on the elastic components of endpoint stiffness. *Exp. Brain Res.*, 141(3):312–323.

Piazza, S. J. and Delp, S. L. (2001). Three-Dimensional Dynamic Simulation of Total Knee Replacement Motion During a Step-Up Task. *J. Biomech. Eng.*, 123(6):599–606.

Pizzolato, C., Lloyd, D. G., Sartori, M., Ceseracciu, E., Besier, T. F., Fregly, B. J., and Reggiani, M. (2015). CEINMS: A toolbox to investigate the influence of different neural control solutions on the prediction of muscle excitation and joint moments during dynamic motor tasks. *J. Biomech.*, 48(14):3929–3936.

Prilutsky, B. I. and Zatsiorsky, V. M. (2002). Optimization-based models of muscle coordination. *Exerc. Sport Sci. Rev.*, 30(1):32–38.

Purushotham, A. and Anjeneyulu, J. (2013). Kane ' s Method for Robotic Arm Dynamics : a Novel Approach. *IOSR J. Mech. Civ. Eng.*, 6(4):7–13.

Raibert, M. H. and Craig, J. J. (1981). Hybrid Position / Force Control of Manipulators. *J. Dyn. Syst. Meas. Control*, 103(2):126–133.

Rajagopal, A., Dembia, C., DeMers, M. S., Delp, D., Hicks, J., and Delp, S. L. (2016). Full body musculoskeletal model for muscle-driven simulation of human gait. *IEEE Trans. Biomed. Eng.*, 9294(c):1–1.

Rasmussen, J., de Zee, M., Damsgaard, M., Christensen, S. T., Marek, C., and Siebertz, K. (2005). A General Method for Scaling Musculo-Skeletal Models. In *2005 Int. Symp. Comput. Simul. Biomech.*, pages 1–2.

Razavian, R. S., Mehrabi, N., and McPhee, J. (2015). A model-based approach to predict muscle synergies using optimization: application to feedback control. *Front. Comput. Neurosci.*, 9(121):1–13.

Reinbolt, J. A., Fox, M. D., Schwartz, M. H., and Delp, S. L. (2009). Predicting outcomes of rectus femoris transfer surgery. *Gait Posture*, 30(1):100–105.

Reinbolt, J. a., Schutte, J. F., Fregly, B. J., Koh, B. I., Haftka, R. T., George, A. D., and Mitchell, K. H. (2005). Determination of patient-specific multi-joint kinematic models through two-level optimization. *J. Biomech.*, 38(3):621–626.

Righetti, L., Buchli, J., Mistry, M., and Schaal, S. (2011). Inverse Dynamics Control of Floating-Base Robots With External Contraints: an Unified View. In *2011 IEEE Int. Conf. Robot. Autom.*, pages 1085–1090.

Robinson, D. A., O'Meara, D. M., Scott, A. B., and Collins, C. C. (1969). Mechanical components of human eye movements. *J. Appl. Physiol.*, 26(5):548–553.

Saul, K. R., Hu, X., Goehler, C. M., Vidt, M. E., Daly, M., Velisar, A., and Murray, W. M. (2014). Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Comput. Methods Biomech. Biomed. Engin.*, 18(December):1–14.

Schutte, L. M., Rodgers, M. M., Zajac, F. E., and Glaser, R. M. (1993). Improving the Efficacy of Electrical Stimulation- Induced Leg Cycle Ergometry: An Analysis Based on a Dynamic Musculoskeletal Model. *IEEE Trans. Rehabil. Eng.*, 1(2):109–125.

Sentis, L. (2007). *Synthesis and Control of Whole-Body Behaviors in Humanoid Systems*. Stanford University.

Seth, A., Sherman, M. A., Eastman, P., and Delp, S. L. (2010). Minimal formulation of joint motion for biomechanisms. *Nonlinear Dyn.*, 62(1-2):291–303.

Seth, A., Sherman, M. A., Reinbolt, J. a., and Delp, S. L. (2011). OpenSim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange. *IUTAM Symp. Hum. Body Dyn.*, 2:212–232.

Shamaei, K., Cenciarini, M., Adams, A. A., Gregorczyk, K. N., Schiffman, J. M., and Dollar, A. M. (2015). Biomechanical effects of stiffness in parallel with the knee joint during walking. *IEEE Trans. Biomed. Eng.*, 62(10):2389–2401.

Shelburne, K. B., Pandy, M. G., and Torry, M. R. (2004). Comparison of shear forces and ligament loading in the healthy and ACL-deficient knee during gait. *J. Biomech.*, 37(3):313–319.

Sherman, M. A. (2013). *Theory Manual*. SimTK.

Sherman, M. A., Seth, A., and Delp, S. L. (2011). Simbody: Multibody dynamics for biomedical research. *Procedia IUTAM*, 2:241–261.

Sherman, M. A., Seth, A., and Delp, S. L. (2013). What is a moment arm? Calculating muscle effectiveness in biomechanical models using generalized coordinates. In *Proceeding ASME*, pages 1–9.

Shourijeh, M. S., Smale, K. B., Potvin, B. M., and Benoit, D. L. (2016). A forward-muscular inverse-skeletal dynamics framework for human musculoskeletal simulations. *J. Biomech.*, 49(9):1718–1723.

Stanev, D. and Moustakas, K. (2014). Virtual human behavioural profile extraction using Kinect based motion tracking. In *Int. Conf. Cyberworlds*, pages 411–414.

Stanev, D. and Moustakas, K. (2018). Simulation of Constrained Musculoskeletal Systems in Task Space. *IEEE Trans. Biomed. Eng.*, 65(2):307–318.

Stanev, D. and Moustakas, K. (2019). Modeling musculoskeletal kinematic and dynamic redundancy using null space projection. *PLoS One*, 14(1):1–26.

Stanev, D., Moustakas, K., Gliatis, J., and Koutsojannis, C. (2015). ACL reconstruction decision support: Personalized simulation of the lachman test and custom activities. *Methods Inf. Med.*, 55(1):98–105.

Steele, K. M., Tresch, M. C., and Perreault, E. J. (2015). Consequences of biomechanically constrained tasks in the design and interpretation of synergy analyses. *J. Neurophysiol.*, 113(7):2102–13.

Stein, R. B. (1982). *What muscle variable(s) does the nervous system control in limb movements?*, volume 5. Cambridge University Press.

Sussman, J. G. and Wisdom, J. (2001). *Structure and Interpretation of Classical Mechanics*. MIT Press.

Tahara, K., Luo, Z.-w. W., and Arimoto, S. (2006). On Control Mechanism of Human-Like Reaching Movements with Musculo-Skeletal Redundancy. In *Int. Conf. Intell. Robot. Syst.*, pages 1402–1409.

Thelen, D. G. (2003). Adjustment of Muscle Mechanics Model Parameters to Simulate Dynamic Contractions in Older Adults. *J. Biomech. Eng.*, 125(1):70–77.

Thelen, D. G. and Anderson, F. C. (2006). Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *J. Biomech.*, 39(6):1107–15.

Thelen, D. G., Anderson, F. C., and Delp, S. L. (2003). Generating dynamic simulations of movement using computed muscle control. *J. Biomech.*, 36(3):321–328.

Uchida, T. K., Seth, A., Pouya, S., Dembia, C. L., Hicks, J. L., and Delp, S. L. (2016). Simulating Ideal Assistive Devices to Reduce the Metabolic Cost of Running. *PLoS One*, 11(9):e0163417.

Valente, G., Pitto, L., Testi, D., Seth, A., Delp, S. L., Stagni, R., Viceconti, M., and Taddei, F. (2014). Are subject-specific musculoskeletal models robust to the uncertainties in parameter identification? *PLoS One*, 9(11).

Valero-Cuevas, F. J. (2009a). A Mathematical Approach to the Mechanical Capabilities of Limbs and Fingers. *Adv Exo Med Biol*, 629:619–633.

Valero-Cuevas, F. J. (2009b). *Fundamentals of Neuromechanics*. Springer.

Valero-Cuevas, F. J., Hoffmann, H., Kurse, M. U., Kutch, J. J., and Theodorou, E. A. (2009). Computational Models for Neuromuscular Function. *IEEE Rev Biomed Eng*, 2:110–135.

Valero-Cuevas, F. J., Johanson, M. E., and Towles, J. D. (2003). Towards a realistic biomechanical model of the thumb: The choice of kinematic description may be more critical than the solution method or the variability/uncertainty of musculoskeletal parameters. *J. Biomech.*, 36(7):1019–1030.

van den Bogert, A. J., Blana, D., and Heinrich, D. (2011). Implicit methods for efficient musculoskeletal simulation and optimal control. *Procedia IUTAM*, 2(2011):297–316.

van Eck, C. F., van den Bekerom, M. P. J., Fu, F. H., Poolman, R. W., and Kerkhoffs, G. M. M. J. (2013). Methods to diagnose acute anterior cruciate ligament rupture: a meta-analysis of physical examinations with and without anaesthesia. *Knee Surgery, Sport. Traumatol. Arthrosc.*, 21(8):1895–1903.

Vempala, S. S. (2005). Geometric Random Walks: A Survey. *MSRI Comb. Comput. Geom.*, 52:573–612.

Wachter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1):25–57.

Weiss, J. A. and Gardiner, J. C. (2001). Computational Modeling of Ligament Mechanics. *Crit. Rev. Biomed. Eng.*, 29(3):303–371.

Weiss, J. A., Gardiner, J. C., Ellis, B. J., Lujan, T. J., and Phatak, N. S. (2005). Three-dimensional finite element modeling of ligaments: Technical aspects. *Med. Eng. Phys.*, 27(10):845–61.

Wesseling, M., Derikx, L. C., de Groote, F., Bartels, W., Meyer, C., Verdonschot, N., and Jonkers, I. (2015). Muscle optimization techniques impact the magnitude of calculated hip joint contact forces. *J. Orthop. Res.*, 33(3):430–438.

Wilson, A. M., McGuigan, M. P., Su, A., and Van den Bogert, A. J. (2001). Horses damp the spring in their step. *Nature*, 414(6866):895–899.

Winter, D. A. (1990). *Biomechanics and motor control of human movements*. Wiley.

Wolpert, D. M. (1997). Computational approaches to motor control. *Nature*, 1(6):209–216.

Xu, H., Bloswick, D., and Merryweather, A. (2014). An improved OpenSim gait model with multiple degrees of freedom knee joint and knee ligaments. *Comput. Methods Biomech. Biomed. Engin.*, pages 37–41.

Yamaguchi, G. T. and Zajac, F. E. (1989). A planar model of the knee joint to characterize the knee extensor mechanism. *J. Biomech.*, 22(1):1–10.

Yang, N., Canavan, P., Nayeb-Hashemi, H., Najafi, B., and Vaziri, A. (2010). Protocol for constructing subject-specific biomechanical models of knee joint. *Comput. Methods Biomech. Biomed. Engin.*, 13(5):589–603.

Zajac, F. E. (1989). Muscle and Tendon: Properties, Models, Scaling and Application to Biomechanics and Motor Control. *Crit. Rev. Biomed. Eng.*, 17(4):359–411.

Zajac, F. E. (1993). Muscle coordination of movement: a prespective. *J. Biomech.*, 26:109–124.

Zhang, D., Poignet, P., Bo, A. P. L., and Ang, W. T. (2009). Exploring peripheral mechanism of tremor on neuromusculoskeletal model: a general simulation study. *IEEE Trans. Biomed. Eng.*, 56(10):2359–69.