Data-driven Models of Human Body Inertia

by

Guanxiong Chen

B.A.Sc. in Computer Engineering, The University of British Columbia, 2021

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL
STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

April 2024

© Guanxiong Chen, 2024
The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

**Data-driven Models of Human Body Inertia**

submitted by Guanxiong Chen in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

**Examinining Committee:**

Dinesh K. Pai, Professor, Department of Computer Science, The University of British Columbia

*Supervisor*

Helge Rhodin, Professor, Faculty of Technology, Bielefeld University

*Supervisory Committee Member*
Abstract

Accurate estimation of mass properties of the human musculoskeletal system is of great interest to many tasks, from gait analysis in biomechanics to motion tracking and control in computer animation. Previous work typically simplified the human musculoskeletal structure as a chain of rigid capsules, with muscle mass lumped with body segments. Such simplifications lead to errors in the system’s inertia matrix, and the error propagates to torque and pose estimates. In this study, we show that we can estimate the generalized joint-space inertia matrix of a human in motion, using a deep neural network or with a simple statistical model. The models do not make any assumptions other than that effective inertia matrices must be symmetric and positive definite. The models are trained and tested with real-world human data that includes synchronized motion and ground reaction forces. We show that a joint-space inertia matrix estimated from data can be physically plausible by revealing inertial coupling which a rigid, lumped inertia matrix fails to entail, and that effective inertia estimates are motion-type dependent. Moreover, we show that our neural inertia model SPDNet can predict inertia matrices parametrized by pose, body mass and height, that its predicted matrices are physically plausible, and that it generalizes well to unseen poses and mass distributions when used to reconstruct motion.
Lay Summary

Accurate estimation of inertial properties of a human body is an important task that many biomechanists and computer animators have sought to solve. In this work, we propose methods of estimating the joint-space inertia matrix of human bodies from motion capture and force plate data. The estimated inertia matrix can be useful for many downstream tasks: letting biomechanists solve joint moments, or letting vision researchers generate new motions for a humanoid character. We show that our data-driven methods work better than prior methods that analytically construct inertia matrices, and that our method works well when injected into a simulator for reconstructing motions.
Preface

The project was initially conceived by Dr. Bastian Wandt and Prof. Helge Rhodin as a follow-up work to the ECCV 2020 paper by Zell et al, “Weakly-supervised Learning of Human Motion Dynamics”. Prof. Dinesh Pai then formulated the research question as an inertia estimation problem, and developed several high-level objectives for the goal of identifying inertia matrix from publicly available motion capture and force plate data. Following the objectives put forth by Prof. Pai, I designed experiments, implemented the code for running the experiments and for visualizing experimental results, executed the experiments, analyzed their results and wrote up most of the thesis. I also made use of the publicly available mocap and force plate dataset, as well as the data preprocessing code from Prof. Bodo Rosenhahn’s group at Leibniz University Hannover for preprocessing our data. Throughout the course of the project, Prof. Pai has helped me with designing experiments and understanding technical details related to the publicly available human motion dataset by Zell et al. Also, he offered me suggestions regarding implementation of the simulation framework, as well as his analysis of results from biomechanical perspectives. Prof. Rhodin and Prof. Wandt have offered me suggestions on training and testing of neural networks. Prof. Pai, Prof. Rhodin and Prof. Wandt have all been involved in refining the thesis write-up.
# Table of Contents

Abstract ................................................................. iii

Lay Summary .............................................................. iv

Preface ................................................................. v

Table of Contents ........................................................ vi

List of Tables ............................................................. viii

List of Figures ............................................................. ix

Acknowledgments ......................................................... xiii

Dedication ............................................................... xv

1 Introduction ............................................................. 1

2 Related Work ........................................................... 4
  2.1 Inertia Parameter Estimation ...................................... 4
  2.2 Body Segment Models for Downstream Tasks .................. 6
  2.3 Human Motion Datasets ........................................... 9

3 Methods ................................................................. 10
  3.1 Effective Inertia ................................................... 11
  3.2 Dataset ............................................................ 12
  3.3 Constant Inertia Models ........................................... 13
List of Tables

Table 4.1  Test set MAE and MPJPE of different models for inertia representation. We evaluated each model on six folds; for each fold we use gait cycles from one subject for testing. (a) Walking Set results. (b) Running Set results. . . . . . . . . . . . . . . . . 29

Table 4.2  Mean test set MAE (↓) and MPJPE (↓) of walking gait inertia estimation from different SPDLayer configurations. We also list the MAE and MPJPE from SymmNet, UnconNet, Constant Inertia Model and the ARB Inertia Model for comparison. . . 30
## List of Figures

| Figure 3.1 | Method overview. (a) Curation of mocap and force plate data and the dataset partitioning scheme. (b) Selection of data to use for inertia estimation. An inertia model, be it a neural network or a constant inertia model, takes data from the first 100 ms of the initial single support phase of a gait cycle, and predicts a joint-space inertia matrix corresponding to the initial pose of the cycle. (c) Kinematic model of the dataset. We focus on analyzing inertial properties and reconstructing motion of lower body segments and the torso. | 11 |
| Figure 3.2 | Architecture of SPDNet. First, we feed body mass and height into the MLP-based feature extractor to predict an 8-D person vector $p$. Next, we feed $p$ along with joint pose $q$ into the MLP-based inertia predictor to produce $\frac{d(d+1)}{2}$ entries. Finally, we feed these entries to the SPD Layer [Jekel et al., 2022] to make the final prediction SPD. | 15 |
Figure 4.1  Distribution of effective inertia estimated by the constant inertia model vs computed by the ARB inertia model over all subjects. Each subject is (a) walking with left foot on ground; (b) walking with right foot on ground; (c) running with left foot on ground; (d) running with right foot on ground. The magnitudes are computed by diagonally lumping inertia matrices first, then aggregating over angles (displacements) associated with a joint (direction). We plot the distribution of magnitudes on log scale (left) and linear scale (right).

Figure 4.2  Distribution of walking vs running effective inertia with the constant inertia model and the ARB inertia model. (a) Constant inertia model’s predictions for left-foot contact gaits and (b) right-foot contact gaits. (c) ARB inertia model’s constructions for left-foot gaits and (d) right-foot gaits. The magnitudes are also computed by diagonal lumping then aggregated. We plot the distribution across all subjects on log scale (left) and linear scale (right).

Figure 4.3  Distribution of effective inertia estimated by SPDNet vs computed by the ARB inertia model over all subjects. Each subject is (a) walking with left foot on ground; (b) walking with right foot on ground; (c) running with left foot on ground; (d) running with right foot on ground. The magnitudes are also computed by diagonal lumping then aggregating, then plotted on log scale (left) and linear scale (right).

Figure 4.4  Distribution of walking vs running effective inertia with SPDNet and the ARB inertia model. (a) SPDNet’s predictions for left-foot contact gaits and (b) right-foot contact gaits. (c) The ARB inertia model’s constructions for left-foot gaits and (d) right-foot gaits. The magnitudes are also computed by diagonal lumping then aggregated. We plot the distribution across all subjects on log scale (left) and linear scale (right).
Figure 4.5  Box-and-whisker plots of metrics from reconstructing test set motion clips of the six folds by different inertia models: (a) Walking Set MAE and (b) MPJPE; (c) Running Set MAE and (d) MPJPE. Left: log scale, right: linear scale.

Figure 4.6  Joint angles’ mean ± one standard deviation vs time from motion clips reconstructed by different models. The joint angles are grouped by anatomical planes they are situated in: e.g. the hip Sagittal angle is the hip-x axis angle shown in Fig. 3.1 (c), and the knee Horizontal angle is the knee-y axis angle shown in Fig. 3.1 (c). An empty plot in the grid means that angle was not measured in the dataset provided by [Zell et al., 2020].

Figure 4.7  Qualitative comparison of reconstructed motion between neural networks, the constant inertia model and the ARB inertia model. The ground truth motion is measured on Subject No. 17 while running, during a stance phase with right foot staying in contact with the ground. We compare at three time instances: 10 ms, 50 ms, 90 ms.

Figure 4.8  Eigenvalues of inertia matrices, either constructed or predicted on Subject No. 17 from a running clip with right foot contacting ground. We can tell SPDNet guarantees eigenvalues of predicted matrix being positive. Note that we set the minimum acceptable eigenvalue to be $1^{-10}$ but that value is too small to be printed out in the figure.

Figure 4.9  An inertia matrix with negative eigenvalues can make the simulated human produce physically implausible behaviors. Here to the simulated human we apply an eigenforce, i.e. an impulse force in the same direction as an eigenvector corresponding to a negative eigenvalue of the system’s predicted inertia matrix. (a) The generalized unitary eigenforce applied to the system. (a) The system’s responding acceleration $\ddot{q}$. We see the applied force and the induced acceleration are in opposite directions, and that is not physically plausible.
Figure A.1  **Synthetic dataset construction and partitioning.** (a) We create two leg models to generate synthetic data from: an unactuated double pendulum and a tendon-driven double pendulum. With the unactuated pendulum, we generate data at different initial poses to obtain different effective inertia matrices. For the tendon-actuated model, we apply different muscle activations so the initial pose would be different. We also set the pendulum’s plane of motion at different tilt angles w.r.t. ground in order to diversify our data. (b) Musculotendious actuators that map to real-world leg muscles.

Figure A.2  **$R^2$ scores of effective inertia fits, on synthetic data from the tendon-actuated double pendulum.** (a) - (b): Fitting constant inertia models on the Pose, Mass Set. (c) - (d): Fitting Neural Network on the Pose, Mass Set.
Acknowledgments

First and foremost, I would like to offer my sincerest appreciation to Prof. Pai, who guided me throughout the entire project, offering his invaluable suggestions on nearly every single aspect of it, from the design of experiments to analysis of results. Without his guidance I may never have pulled off this project. But in fact, to me Prof. Pai’s role is more than my thesis supervisor who oversees just a project; he is a true mentor who taught me how to conduct research and specifically, research in physics-based simulation. During our meetings he presented to me many textbooks, papers (especially those that were authored by himself and his former students) that are highly relevant to the project, and it was just mind blowing to me seeing how he was able to draw connections between trendy, current research topics to studies he did decades ago with his deep insights into physics-based simulation. Moreover, from his reading group discussions I was able to pick up many invaluable lessons, from literature review strategies to his and Dr. Shu’s invaluable opinions related to most cutting edge research work in the field. On top of that, Prof. Pai is an amazing, supportive supervisor who is willing to offer his encouragement and help when I was either stuck and lacking progress in research, or when I bumped into issues in other aspects of my studies. So I believe Prof. Pai is not only an expert in the field and a true scholar, but also a supervisor of good character. I truely appreciate all the amazing opportunities he has offered to me, and I consider him as a role model, a moral compass, a Jedi Master in the sense that if I have students to mentor in the future I should be guiding the same way that he guides me.

Next, I want to thank Prof. Rhodin and Prof. Wandt for their genuine support over the course of the project. They are real experts in machine learning and vision,
and only with their help in designing and tuning of neural architectures I was able to build a neural network that outperforms baseline. They are always willing to listen to my newbie questions and offer their opinions with great patience. Also, I want to thank Prof. Wandt for connecting me to Prof. Bodo Rosenhahn’s group so I could ask them questions related to how their data was collected and preprocessed.

And I hereby extend my appreciation to Natural Sciences and Engineering Research Council of Canada (NSERC) for its funding of the project.

Moreover, I want to thank Felix Winkler for helping me with organizing the data preprocessing scripts used for preparing the publicly human motion dataset from Prof. Rosenhahn’s group, and answering many of my questions related to the dataset. He is an expert in his project, a wonderful human being, and I wish him the best in his studies.

Other than people directly involved in the project, I would like to thank my schoolmates and friends including Pearson, Megha, Egor, Jennie, Yibo, Haomiao, Bole, Ming for their help and company; especially to Yuhao, Tianxin, Dingqing, Soheil for their valuable advice on my studies. Hearing what they think about research and telling them what I think about my experiences make me feel that I am not alone in my academic endeavor, that my experiences are more shared than individual, and that I am connected to my peers in many ways. I also want to thank my friend Chun for offering me a cozy space to write up the thesis during my trip to China. The place was quiet, lovely, with amazing food options, and it really helped me focus on the writing. Without your help my time during the trip would have been a lot less productive.

Last but not least, I want to thank my mother, my grandmother and my father for offering me unconditional love and every kind of support I can ever think of. Knowing that they will always love me, protect me, believe in me and support me on pursuing graduate studies is an essential part of what drives my forward in the realm of academia. Furthermore, I want to thank grandfather Tailin and uncle Rui for their great advice on how to work with others in academia and on conducting research. Their wise words offered me new perspectives on many subjects that matter to graduate students, and these words played rather crucial roles in my making of several important decisions. I cannot thank them more for the generous support they are willing to offer me.
Dedication

To my grandmother Jingwen and my late grandfather Chenglin.
Chapter 1

Introduction

Identifying inertial parameters of human subjects has been a pivotal and long-standing problem in biomechanics. The identified inertia can be useful for a multitude of applications: from solving inverse dynamics [Dao, 2019, Lam and Vujaklija, 2021, Shourijeh et al., 2017, Xiang et al., 2011, Zell et al., 2020] and simulating human motions [Clancy et al., 2023, Ren et al., 2007, Sharif Razavian et al., 2019, Song et al., 2021] within the domain of biomechanics, to simulating anatomically realistic animated characters [Jiang et al., 2019, Lee et al., 2019a, Nakada et al., 2018, Park et al., 2022] or improving accuracy and physical realism of motion tracking in computer vision [Andrews et al., 2016, Gärtner et al., 2022, Shimada et al., 2020].

Pioneering work that tackled the problem estimated body segment inertial parameters (BSIPs) of live subjects from photometric data and anthropometric measurements [Dumas et al., 2007, McConville et al., 1980]; the data though generalizes to the average human population, is not specific to individuals outside the group of measured subjects. Recent work measured BSIPs through in-vivo studies on live subjects using motion capture and force plate data [Chen et al., 2011, Damavandi et al., 2009, Hansen et al., 2014, Jovic et al., 2016, Pataky et al., 2003, Venture et al., 2008]. Regardless of their data sampling approach or estimation method, these approaches were all built upon the articulated rigid body (ARB) assumptions: body segments were approximated with rigid capsules or cylinders, chained up by joints; the mass of a muscle is lumped to the skeletal mass of its near-
est segment and uniformly distributed along the segment [Pai, 2010]. In fact, these assumptions were employed not only by prior work of estimating inertia from real-world data, but also by software that simulate biomechanical systems [Rajagopal et al., 2016, Scholz et al., 2016, Seth et al., 2018]. But the two assumptions can fail to fully characterize the complexity of human musculoskeletal systems: (i) body segments are non-rigid with complex geometries; (ii) a muscle can be attached to multiple bones, can elongate or contract in motion. Since muscles contribute a significant portion of human body mass and thus significantly to the inertia of the full body, overlooking these aspects in musculoskeletal modelling can lead to 7.6% of error in estimates of the system’s effective ankle inertia [Pai, 2010] and propagate the error to downstream applications, e.g. gait prediction or joint torque estimation [Verheul et al., 2023]. To mitigate errors due to the ARB assumptions, Wang et al. [2022b] built a differentiable simulator that unlumps muscle mass from skeletal mass, by approximating muscles as threads of “mass points” attached to bones, then derived the joint space inertia matrix of a leg system analytically from the approximated model. Although this approach does account for the non-rigidness of body segments and inertia can scale to individual subjects, the mass points are nevertheless an approximation, so are the computed inertia.

In this study, we aim to develop data-driven representations of the joint-space inertia matrix of human subjects in motion. To this end, first we developed the constant inertia model, an inertia matrix representation defined per subject, per pose from kinematic and dynamic data collected via a motion capture system and force plates [Zell et al., 2020]. The model assumes nothing other than that the matrix should be symmetric and positive definite (SPD). We show that after fitting the model to data with constrained least squares, the yielded matrix is biomechanically plausible, and it reveals inertial effects that the articulated rigid body (ARB) inertia model fail to capture, including larger inertia on distal joints and the motion type dependence of inertia. On the other hand, we recognize that least-squares fitting one model for each subject, each pose can be computationally expensive, and the model can fail to generalize to unseen subjects and poses when used for downstream tasks such as motion reconstruction. Upon witnessing the incredible prowess of deep neural networks [Goodfellow et al., 2016] on a multitude of regression tasks in biomechanics, from motion tracking [Pearl et al., 2023] to joint torque
prediction [Lam and Vujaklija, 2021], we developed a neural inertia model named SPDNet as a representation of joint-space inertia matrix parametrized by pose and subject-specific physical properties including body mass and height. We show that a trained SPDNet can predict physically plausible inertia; and that when trained on datasets of different motions, the predicted inertia can also reveal motion-type dependence of inertia. Further, we show that SPDNet’s predicted matrices yield lower motion errors than ARB inertia when we applied them to motion reconstruction of gaits from subjects not seen in training. To the best of our knowledge, we are the first group ever attempted to use neural networks for estimating the inertia matrix or BSIPs from human kinematic and force plate data in a biomechanical context.

But this study still has several major limitations. First, while our study shows that data-driven models can reveal that estimated inertia can be different for walking and running motion, explaining the gap between walking and running inertia is beyond the scope of this study. Second, our estimation and motion reconstruction only works on a specific phase in each gait cycle lasting roughly 100 ms, during which velocity change is large and musculocontrol stays constant. Third, our methods require both kinematic data and dynamic data, making large, publicly available datasets such as Mahmood et al. [2019] that contain only kinematic data not suitable for estimation. And thus far we have only evaluated the methods on walking and running motion, so how they will perform on other motions such as jumping, kicking are yet to be known. Lastly, we have yet to interpret our estimated inertia from the standpoint of mass distribution and segment geometry. We believe future studies can be carried out to resolve these limitations, so our method can be applied in a broader scope.
Chapter 2

Related Work

Inertia estimation of human body is not only an area of interest in biomechanics, but also fundamentally connected to the problem of human motion tracking and character control that are of great interest to the computer vision and graphics community: in a typical force-based tracking or control pipeline, the character’s actuation forces are fed into a forward dynamics solver, and the solver computes velocity and position in the next time step based on the character’s mass matrix as well as force inputs. In this Chapter, we offer a systematic review of prior work in the following domains:

- Estimation of human body segments’ inertial parameters (BSIPs) or the joint-space mass matrix from biomechanics and computer vision venues;

- Body segment models built for analytically constructing the joint-space mass matrix. We also review various downstream tasks in biomechanics, vision and graphics that make use of those models and constructed matrices: solving inverse dynamics, motion tracking and character control.

- Datasets available for data-driven inertia estimation.

2.1 Inertia Parameter Estimation

Biomechanists have sought to accurately identify the inertial parameters of human body segments for centuries [McConville et al., 1980]. While people in the dis-
tant past have conceived and executed ingenious methods to measure BSIPs, we constrain our discussion to methods from recent decades that utilize modern measurement and computational techniques. Pioneering work by Dumas et al. [2007], McConville et al. [1980], Young et al. [1983], Zatsiorsky [1985] attempted to estimate BSIPs such as the mass, center of mass and principle moment of inertia of live human subjects from photometric data and anthropometric measurements. These measurements are taken when subjects were at static poses but not when they were in motion, and in these early work the geometry of each body segment was not thoroughly examined. Later work derived BSIPs of human subjects from force plate data [Damavandi et al., 2009, Jovic et al., 2016, Pataky et al., 2003], or kinematic mocap data [Venture et al., 2008] or both [Ayusawa et al., 2011, Chen et al., 2011, Hansen et al., 2014]. These methods allow measurements to take place while subjects are performing actions and thus, detect differences in inertial parameters between when a subject is in motion and when the subject is static. Yet two problems persist in these later methods: (1) No canonical standards exist to evaluate the accuracy of obtained BSIPs [Mungiole and Martin, 1990, Pataky et al., 2003]. Typically, inertial parameters were verified by comparing them with values obtained from prior studies. But since prior studies often did not employ same measurement and estimation method, one may wonder the extent to which such comparisons are meaningful. (2) They typically employ the articulated rigid body (ARB) assumption: body segments were approximated with rigid capsules or cylinders of uniform mass distribution, chained up by joints. While these are reasonable assumptions to make when identifying inertial parameters of robots [Bonnet et al., 2016, Lee et al., 2019b, Wensing et al., 2017], a human body segment contains bones, tendons, fat and muscles which are deformable, so parameters such as the distribution of mass within a segment can change during human movement.

Another line of work in biomechanics seeks to use MRI data for identifying BSIPs [Cheng et al., 2000, Martin et al., 1989, Mungiole and Martin, 1990], or identifying BSIPs along with constructing detailed geometric representation of each segment [Sreenivasa et al., 2016]. While BSIPs estimated from MRI scans are comparable to those identified from photometric data and anthropometric measurements, MRI-based images also suffer from problems that plagued the prior methods, such as overestimating segment volumes. Moreover, MRI scans can only
be captured when a subject is in a static pose, and procuring these scans can be expensive.

Other than literature in biomechanics, a few recent work from computer vision and graphics have also attempted to estimate inertial parameters of humans in motion. Lv et al. [2016] estimated the mass, center of mass and inertia of each body segment of human subjects when performing multiple kinds of motions, including walking, running and jumping, by using mocap data, force plate data and pressure insoles’ data as inputs to their method. They acknowledge that their approximation of body segments as rigid bodies can lead to errors, but offered no validation of the estimated parameters. Yang et al. [2023] attempted to estimate per-segment mass of humans in motion from monocular videos containing only kinematic information, for their end goal of reconstructing human motion. They confirmed that estimating per-segment mass indeed improves the quality of their reconstruction. The most recent, relevant work by Zhang et al. [2024] estimated the entire joint-space mass matrix of simulated human subjects rather than their individual body segment masses using a neural network. The network takes past joint angles as input, transforms it through multiple fully-connected layers with activation and finally produces the mass matrix. Although they showed that their motion reconstruction quality improves upon introducing a physics-based branch that estimates mass and other physical forces, since inertia estimation is not the focus of the paper, they did not either verify the physical plausibility of estimated mass matrices, or confirm estimating the matrices alone can improve reconstruction quality.

2.2 Body Segment Models for Downstream Tasks

In the majority of biomechanics, vision and graphics literature, researchers construct joint-space mass matrix for a humanoid character to solve its dynamics. The matrix is constructed analytically from body segment models built with geometric primitives, and the level of detail of such models varies across literature. For solving inverse dynamics, e.g. estimating ground reaction force (GRF) or joint torque (JT), one would use the mass matrix and the character’s kinematic state to solve for forces [Brubaker et al., 2009, Clancy et al., 2023, Park et al., 2022, Ren et al., 2007, Shimada et al., 2021, Venture et al., 2008, Verheul et al., 2022, Yi et al., 2022, Zell
and Rosenhahn, 2015, 2020, Zell et al., 2020]; to solve motion tracking or character control problem, an integrator would take the constructed mass matrix as well as external forces applied to the character, compute the character’s joint-space acceleration and finally, advance the character to its next state [Andrews et al., 2016, Brubaker et al., 2010, Gärtnert et al., 2022, Rajagopal et al., 2016, Yang et al., 2023].

The simplest yet most commonly employed is the class of articulated rigid body (ARB) models: each body segment is modelled as a rigid object, often in cylindrical or capsular shape, with mass uniformly distributed in the segment [Bergamin et al., 2019, Gärtnert et al., 2022, Liu et al., 2021, Lv et al., 2016, Shimada et al., 2021, Venture et al., 2008, Yi et al., 2022, Zell et al., 2020]. The mass of each segment either follows population average [Gärtnert et al., 2022, Yi et al., 2022, Zell et al., 2020] or is personalized per subject via optimization [Lv et al., 2016, Venture et al., 2008]. These models are easy to build and to actuate for control tasks [Bergamin et al., 2019, Shimada et al., 2021].

Another class of body segment model is musculoskeletal models that uses different geometric primitives to represent tissues that constitute body segments, e.g. bones, muscles and tendons. In these models, tendons and muscles are represented by elastic strings attached to bones [Park et al., 2022, Rajagopal et al., 2016, Scholz et al., 2016, Seth et al., 2018, Shourijeh et al., 2017, Zhang et al., 2019]. This kind of models is better in terms of anatomical realism, therefore suitable for clinical analysis in biomechanics [Rajagopal et al., 2016, Scholz et al., 2016, Zhang et al., 2019] and has been used by professional biomechanical modelling software such as AnyBody [Damsgaard et al., 2006] and OpenSim [Seth et al., 2018]. Recent work in graphics have also become interested in this class of models: while such models are muscle-driven and therefore inherently difficult to control, due to significant advances in neural network-based controllers in recent years we are seeing more neural network-controlled, musculoskeletal animated characters [Lee et al., 2019a, Park et al., 2022]. Mainstream robotics simulation suites such as MuJoCo also support simulating this kind of models [Todorov et al., 2012], and open-source part models have been built by the graphics and robotics community [Wang et al., 2022a]. But the major difference between this class of models and ARB models is in control: while most ARB models come with PD (joint angle) or torque
controllers, musculoskeletal models are often muscle-actuated, i.e. a controller produces muscle activation levels, and the activation level controls the length of tendons (muscles), so bones rotate around joints [Lee et al., 2019a, Park et al., 2022, Seth et al., 2018, Thelen and Anderson, 2006]. But in terms of geometric construction tendons are not distinguished from muscles [Lee et al., 2019a, Park et al., 2022, Seth et al., 2018]; moreover, when looking at inertial parameters or the mass matrix derived from those parameters, such models are fundamentally identical to ARB models: the mass of muscles and tendons are lumped to their nearest bones, such that each segment is effectively treated as a single rigid body in inertia evaluation [Scholz et al., 2016].

Biomechanists have noticed the issue with mass lumping, and developed solutions to mitigate it. Pai [2010] pointed out that the lumping strategy is problematic because (i) the mass of a body segment comes from muscles, not bones; (ii) lumping ignores the inertial coupling of muscles to distal joints and thus can lead to inertia of distal joints being incorrectly estimated. They developed a skeletal model consisting of two bones connected by a revolute joint, and created an “unlumped” musculoskeletal model in which a muscle is an elastic line segment with uniformly distributed mass connecting two bones. Compared with the inertia obtained from a model with all muscle mass lumped to the bones, the unlumped system’s inertia is significantly different. Later work attempted to resolve this issue by leveling up simulated details: Zhang et al. [2019] models a muscle using multiple Cosserat rods, with each rod having its own elastic properties and mass. But this model suffers from scalability issues due to extra degrees of freedom introduced [Zhang et al., 2019]. Building upon Pai [2010], Wang et al. [2022b] further discretized muscle as a thread of “mass points” connected via a muscle path that is attached to bones. Although the “mass points on a thread” model mitigates the issues due to muscle lumping to some extent, it is nevertheless an approximation to real muscles, and in some cases the mass matrix analytically constructed from the model still underestimates muscle inertia [Wang et al., 2022b].
2.3 Human Motion Datasets

Datasets that can be used to estimate joint-space mass matrices of humans in motion can be broadly divided into two categories: dynamic datasets and kinematic datasets.

Dynamic datasets contain dynamic information such as ground reaction force, and often kinematic information such as joint positions, and the physical dimension, mass of each subject as well. Prior work in biomechanics generally use dynamic datasets for identifying BSIPs [Ayusawa et al., 2011, Chen et al., 2011, Damavandi et al., 2009, Hansen et al., 2014, Jovic et al., 2016, Pataky et al., 2003]. Datasets such as the ones used in Ayusawa et al. [2011], Chen et al. [2011], Hansen et al. [2014] consist of motion clips collected with mocap systems that measure indoor human motion up to high accuracy, and force plate data that contain ground reaction forces. Although these datasets contain abundant amount of information from included human movements, their are small in scale because only the motion of a few subjects and a few types of in-door motions were measured since mocap systems can only be installed in laboratory settings. Also, these datasets are not publicly available. But in recent years we see more dynamic datasets becoming publicly available: Zell et al. [2020] built a dataset consisting of mocap clips and force plate measurements from 22 subjects; Mourot et al. [2022] built a dataset including mocap clips as well as pressure insole data measured on 18 categories of motions.

On the other hand, many large-scale kinematic datasets are publicly available for training visual models to understand human motion [Kuehne et al., 2011, Mahmood et al., 2019]. The AMASS dataset for example contains 40 hours of motion data collected from hundreds of subjects covering over ten thousand kinds of motions [Mahmood et al., 2019]. Recent work from computer vision have attempted to estimate inertial parameters from kinematic data [Yang et al., 2023, Zhang et al., 2024] using deep neural networks. Since dynamic information such as ground reaction force is not available, they performed the estimation by plugging the resulting mass matrix into a differentiable simulation pipeline, and using a loss function that penalizes differences between ground-truth and predicted trajectory to train the neural networks.
Chapter 3

Methods

In this Chapter we discuss in detail our methods of estimating the effective inertia of a human subject in motion, using kinematic and dynamic data. Section 3.1 defines effective inertia (matrix) which we intend to estimate. Section 3.2 describes the specification of the dynamic dataset that we use. In Section 3.3 we introduce the constant inertia model, a non-parameterized representation of the effective inertia matrix defined on a per-subject, per-pose basis, and discuss how we optimized the model via constrained optimization. In Section 3.4 we introduce neural inertia models that define effective inertia matrices with respect to poses and body proportions. These are parametrized representations of the effective inertia matrix, and we discuss how we optimized them using gradient descent in the same section. Section 3.5 gives a brief overview of the articulated rigid body (ARB) inertia model that we use as a baseline to compare with the former two models. The ARB inertia model is a parametric representation of the inertia matrix body mass, body height and pose; yet unlike the other two models, it is defined by analytical construction rather than estimation from data. In Section 3.6 we discuss how we plug an inertia matrix, be it constructed or estimated into a differentiable simulation pipeline to reconstruct the motion of human subjects in stance phase, and then move onto Section 3.7 to discuss the metrics we used to evaluate the quality of reconstruction and hence, the quality of inertia estimation. We also refer readers to Appendix A for details about how we constructed a synthetic dataset and prototyped our estimation methods on it.
3.1 Effective Inertia

Figure 3.1: Method overview. (a) Curation of mocap and force plate data and the dataset partitioning scheme. (b) Selection of data to use for inertia estimation. An inertia model, be it a neural network or a constant inertia model, takes data from the first 100 ms of the initial single support phase of a gait cycle, and predicts a joint-space inertia matrix corresponding to the initial pose of the cycle. (c) Kinematic model of the dataset. We focus on analyzing inertial properties and reconstructing motion of lower body segments and the torso.

Prior work in human motion reconstruction and analysis [Zell et al., 2020, Zhang et al., 2024] typically formulate the dynamic equation of motion of a hu-
the human kinematic model in generalized coordinates as

\[ M(m, h, q)\ddot{q} = G(m, q) + C(m, \dot{q}) + \tau_{\text{contact}}, \]  

(3.1)

where \( m \) is the body mass of a human subject, \( h \) is the body height of the subject \( M(q) \) is the \( d \times d \) (where \( d \) is the degree of freedom of the kinematic model) subject and pose-dependent joint-space inertia matrix of the subject, \( , q \) is pose, \( G(m, q) \) is the pose and body mass-dependent gravitational force (GRF), \( C(m, \dot{q}) \) is the Coriolis force, \( \tau_{\text{contact}} \) is ground reaction force and moment. And the inertia matrix \( M(q) \) is often habitually constructed from the articulated body (ARB) assumptions—we showcase the construction from Zell et al. [2020] in Sec. 3.5 and used the inertia as the baseline. But an analytically constructed ARB inertia cannot capture the complex geometry and mass distribution of body segments; moreover, due to lumping of muscle mass to skeletal mass, the inertia of a joint would only be influenced by its distal segments, and the mass of a muscle will not contribute to the inertia of its distal joints, even though it should be inertially coupled to those joints [Pai, 2010]. To address the two problems, Verheul et al. [2022], Wang et al. [2022b] took an analytical approach by approximating a muscle as a thread of mass points, with mass uniformly distributed among the points. We instead take a data-driven approach: given measured body mass \( m \), pose \( q \) and GRF, we directly estimate the inertia matrix \( M(q) \); we refer to an estimated inertia that takes account of the complexity of geometry, mass distribution and inertial coupling as an effective inertia.

### 3.2 Dataset

We used the publicly-available human motion dataset from Zell et al. [2020]. It contains 318 walking clips and 132 running clips collected from 22 subjects, with different gender and BMI’s; each clip consists of joint poses sampled via a Vicon Mocap System and contact forces sampled via AMTI force plates [Zell et al., 2020] at 10 ms intervals, from the first half of a gait cycle. Since we are interested in knowing if inertia is motion-type dependent, we divided the dataset into the Walking Set and the Running Set, and estimated inertia on the two sets separately. Since the inertia matrix also depends on body mass proportion and pose, we further partitioned each of the two sets on a per-subject, per-pose basis: split into subsets,
each made up of gait cycles from one subject, then divided the subset into left-foot and right-foot contact set. See Fig. 3.1 (a). Moreover, we exclusively use data from the initial single support phase $t_0 \leq t < t_L$ in a gait cycle for estimation: see Fig. 3.1 (b) [Bonnefoy-Mazure and Armand, 2015]. This is a period of time during which the ipsilateral foot is in contact with the ground and the contralateral foot has just lifted off, and it lasts for roughly 13% of a gait cycle [Bonnefoy-Mazure and Armand, 2015]. So we pick a uniform window of $t_L = 100$ms for all samples in both the Walking and the Running Set. We pick this particular window because during this time joint velocity change is sufficiently large such that ignoring velocity dependent forces (e.g. the Coriolis force) and musculo control will not lead to significant errors. The dataset has been preprocessed by Zell et al. [2020] such that the kinematic model consists of 24 degrees of freedom, including 6 degrees representing root translation and rotation, and 18 lower-body joint angles: see Fig. 3.1 (c).

### 3.3 Constant Inertia Models

We can estimate a single effective inertia matrix at a particular pose for a particular subject in a 100 ms-long estimation window, because we can safely assume that the matrix stays constant in the window since pose changes are minuscule. Starting from discretizing the motion of a $d$-DoF system at an arbitrary time $t = t_i$ in the window during $t_0 \leq t_i \leq t_{L-1}$ on the velocity level:

$$M_{\text{effective}}(\dot{q}_{t_i+1} - \dot{q}_{t_i}) = hG(q_{t_i}) + h\tau_{\text{contact}}(t_i), \quad (3.2)$$

where $M_{\text{effective}}$ is a $d \times d$ effective inertia matrix that depends on motion type, $G$ is gravitational torque and $h$ is the length of a timestep. We drop the Coriolis term since it is small relative to other terms. We expect $M_{\text{effective}}$ to be symmetric and positive definite, since a non-SPD effective inertia implies a non-positive eigenvalue, which means if we apply a generalized force that is parallel to its corresponding eigenvector, the responding acceleration would be in the opposite direction, which is physically implausible. So when pose and subject are fixed, we can construct a simple yet effective statistical model to estimate the effective inertia, by minimizing the difference between predicted and ground-truth per-step squared
impulse \( f(M_{\text{effective}}) \), using constrained least squares:

\[
\begin{align*}
\text{subject to } & M_{\text{effective}} \succ 0,
\end{align*}
\]

(3.3)

where \( X \in \mathbb{R}^{d \times L} \) is the velocity difference matrix:

\[
X = [x_0, x_1, \ldots, x_{L-1}] \quad \text{where} \quad x_i = q_{t_{i+1}} - q_{t_i},
\]

(3.4)

and \( \tau \in \mathbb{R}^{d \times L} \) is the external force vector:

\[
\tau = [\tau_0, \tau_1, \ldots, \tau_{L-1}] \quad \text{where} \quad \tau_i = [G(q_{t_i}) + \tau_{\text{contact}}(t_i)].
\]

(3.5)

The effective inertia matrix \( M_{\text{effective}}^* \) that yields \( f_{\text{min}} \) is the constant inertia matrix we seek for.

To evaluate the physical plausibility of the constant inertia model, we fit to the Running Set and the Walking Set respectively, and for each of the two sets we fit one model per subject, per ipsilateral foot. This guarantees that gait cycles used to fit each model share the same body proportions and a roughly consistent pose. But to evaluate the constant inertia model’s ability to generalize to unseen body proportions and poses and for comparison with neural inertia models, we fit a single constant inertia model respectively to the Walking Set and the Running Set, such that the fitted effective inertia is averaged across different subjects and poses. In terms of implementation, we use optimization package CVXPY [Diamond and Boyd, 2016] along with CVXPYLayer [Agrawal et al., 2019].

We demonstrate that the estimated effective inertia is physically plausible, and that the effective inertia matrices are both drastically different from ARB inertia and dependent of the type of motion a gait cycle belongs to. These properties can be conveniently overlooked by the aforementioned ARB assumptions, but not so when we estimate inertial properties from dynamic data.
### 3.4 Neural Inertia Models

A problem with the constant inertia model as a non-parametrized representation of effective inertia matrix is that it does not generalize to different mass distributions or poses, so an estimated inertia matrix is only physically plausible if fitted to a particular subject at a specific pose. This makes constant inertia models computationally expensive to estimate, if we ever want to estimate multiple such models in order to reconstruct the motion of multiple subjects in different poses. Therefore, we aim to develop neural networks that predict a joint-space inertia matrix parameterized by joint poses and body mass distributions.

![Figure 3.2: Architecture of SPDNet. First, we feed body mass and height into the MLP-based feature extractor to predict an 8-D person vector \( p \). Next, we feed \( p \) along with joint pose \( q \) into the MLP-based inertia predictor to produce \( d(d+1)/2 \) entries. Finally, we feed these entries to the SPD Layer [Jekel et al., 2022] to make the final prediction SPD.](image)

Fig. 3.2 shows the architecture of SPDNet which we developed for inertial prediction. The network predicts a \( d \times d \) SPD effective inertia matrix from joint pose, body mass \( m \) and body height \( h \):

\[
M_{\text{effective}} = f_{SPD}(q, m, h),
\]  

where \( q \in \mathbb{R}^d \) are joint angles. It consists of three components: (i) a person vector extractor. This is a simple multilayer perceptron (MLP) network [Goodfellow et al., 2016] with five fully-connected (FC) layers and leaky ReLU (LReLU) ac-
tivation [Nair and Hinton, 2010, Xu et al., 2015]. It takes a subject’s body mass and height as inputs and predicts an 8-D latent vector $p$ which encodes the inertial and geometric properties of the subject. (ii) an inertia predictor. It is also an MLP network, but with eight FC layers and more neurons in each layer. For the last FC layer we include $\frac{d(d+1)}{2}$ neurons, so the network predicts the set of independent entries required for constructing a symmetric inertia matrix. (iii) An SPD Layer [Jekel et al., 2022] which enforces inertia matrix predictions to be SPD. To fit the network on real-world data with $d = 24$, we include $\frac{24 \times 25}{2} = 300$ neurons in the last FC layer in the inertia predictor. For both MLP networks we included a descending order of neurons from the first to the last FC layer, following the convention of MLP network design for regression tasks [Ozates et al., 2023].

We train the network using per-step impulse loss which is similar to Eq. 3.3:

$$
\frac{1}{N} \frac{1}{L} \sum_{j=1}^{N} \sum_{i=0}^{L-1} \left\| M_{\text{effective}}(\dot{q}_{j,t_{i+1}} - \dot{q}_{j,t_{i}}) - h(G(q_{j,t_{i}}) + \tau_{\text{contact}}(t_{i})) \right\|^2,
$$

(3.7)

where $G(q_{j,t_{i}})$ and $\tau_{\text{contact}}(t_{i})$ are ground-truth generalized forces from gait cycle $j$, time $t_{i}$. In terms of the optimization method, we use mini-batch gradient descent, with the ADAM optimizer [Kingma and Ba, 2014] at the learning rate of $5e^{-4}$, and batch size of 32 gait cycles for 500 iterations.

To confirm that our trained models can generalize to unseen poses as well as subjects of unseen mass distribution and heights, we perform leave-one-out cross validation: For each of the Walking Set and the Running Set, we create a fold by randomly pulling one subject out for testing, two subjects for validation and we leave the rest of subjects for training, and we train and test the network over six folds. We show, using the ARB inertia model as a baseline, that if we simulate a subject’s motion with inertia matrices predicted from SPDNet, we can attain lower motion errors than using the ARB inertia model.

Next, we present an ablation study to explore the impact of architectural decisions. We developed SymmNet $f_{\text{Symm}}$ which takes the same inputs as SPDNet, contains the same MLP networks, and constructs a symmetric effective inertia matrix from the output of the MLP network just like SPDNet. But it does not pass the matrix to the SPDLayer [Jekel et al., 2022] to make the matrix positive definite. We
also developed UnconNet, an “unconstrained” network whose inertia matrix predictions are not subject to any hard constraints. Its MLP network directly predicts \(d \times d\) entries of an inertia matrix instead of its unique entries, and it does not contain the SPD Layer following the MLP network. For training and testing SymmNet and UnconNet, we use the same hyperparameters used for SPDNet. In Sec. 4.2, we show that enforcing inertia matrix to be SPD via the SPD layer [Jekel et al., 2022] significantly lowers motion errors, while guaranteeing physical plausibility, and that SPDNet outperforms its two counterparts. Then we look into hyperparameter decisions on the SPD Layer—specifically our selection of the Eigen Layer and the ReLU function for positivity enforcement, and discuss why we believe the combination of the two yields the best performance.

The neural networks were implemented in PyTorch [Paszke et al., 2019]. We completed all of our experiments on a desktop with i7-11700 CPU, 64 GB of RAM and NVIDIA RTX 3090 GPU with 24GB of RAM.

### 3.5 The Articulated Rigid Body (ARB) Inertia Model

While both the constant inertia model and neural networks need to be trained with data, we can define an ARB inertia model by analytically construction, when the body mass, height and the pose of a subject is given, following the approach in [Zell et al., 2020], without relying on state trajectory or dynamic information. First we compute the rotational inertia of each body segment \(I_{\text{torso}}\), \(I_{\text{pelvis}}\), \(I_{\text{thigh}}\), \(I_{\text{shank}}\), \(I_{\text{foot}}\) in its local frame under the following assumptions: (i) the pelvis and torso are elliptical cylinders; (ii) thighs and shanks are cylindrical cylinders; (iii) feet are semi-ellipsoids. This gives us the half-body rotational inertia \(I_{\text{rot}} \in \mathbb{R}^{3N_s \times 3N_s}\), where \(N_s\) is the number of segments in our ARB skeletal model. Then we put translational and rotational inertia together into the spatial inertia matrix:

\[
M_{\text{spatial}}(q,m,h) = \begin{bmatrix}
I_{\text{trans}}(m) & 0 \\
0 & R_{\text{segs}}(q,h)I_{\text{rot}}(m,h)R_{\text{segs}}(q,h)^T
\end{bmatrix},
\]
where $I_{\text{trans}} \in \mathbb{R}^{3N_s \times 3N_s}$ is the diagonal matrix of per-segment masses:

$$I_{\text{trans}}(m) = \begin{bmatrix} m_{\text{torso}} & 0 & \ldots & 0 \\ 0 & m_{\text{pelvis}} & \ldots & 0 \\ 0 & \ldots & m_{\text{left thigh}} & 0 \\ 0 & 0 & \ldots & m_{\text{right foot}} \end{bmatrix}, \quad (3.9)$$

and $R_{\text{segs}}(q)$ is the $R^{3N_s \times 3N_s}$ segment rotation matrix that depends on configuration $q$. The mass of each body segment is computed by multiplying total body mass $m$ with Dempster’s body parameters [Winter, 2009], which is estimated from general human population. We can then compute a joint-space, pose and body proportion dependent ARB inertia matrix $M_{\text{ARB}}(q, m, h)$ for any pose $q$:

$$M_{\text{ARB}}(q, m, h) = J_{\text{segs}}(q)^T M_{\text{spatial}}(q, m, h) J_{\text{segs}}(q), \quad (3.10)$$

where $J_{\text{segs}}(q)$ is the $R^{6N_s \times d}$ segment Jacobian matrix.

### 3.6 Simulation

To reconstruct a subject’s motion in the estimation window, we inject the estimated inertia matrix $M(q_{t=0})$ to a forward dynamics pipeline, and obtain pose $q$ over $L$ frames in the estimation window. Specifically, at each frame $t$ we solve for generalized acceleration first, using Tikhonov-regularized least squares since the system can be highly ill-conditioned:

$$\ddot{q}_t = \arg \min_{\dot{q}} \| M_{\text{effective}}(\dot{q}_{t=0}) \ddot{q}_t - (G(q_t) + \tau_{\text{contact},t}) \|^2 + \lambda \| D \ddot{q}_t \|^2, \quad (3.11)$$

where $\lambda = 1e-5$ is the regularization parameter and $D$ is the $d \times d$ identity matrix that penalizes large joint acceleration along any particular dimension. We then apply Semi-Implicit Euler integration to update the generalized state:

$$\begin{bmatrix} \dot{q}_{t+1} \\ \ddot{q}_{t+1} \end{bmatrix} = \begin{bmatrix} \dot{q}_t \\ \ddot{q}_t \end{bmatrix} + h \cdot \begin{bmatrix} \ddot{q}_t \\ \ddot{q}_{t+1} \end{bmatrix}. \quad (3.12)$$
3.7 Metrics

Since no ground truth inertia matrices that capture non-rigidity and inertial coupling are available, to evaluate our trained inertia estimators, we inject the estimated inertia into a motion simulation pipeline and measure reconstruction errors, including mean angle error (MAE) and mean per-joint position error (MPJPE).

MAE is the predominant metric for evaluating the quality of human motion prediction in the last decade [Lyu et al., 2022]. For a batch of \( N \) gait cycles, we compute MAE as

\[
\text{MAE} = \frac{1}{N L K} \sum_{n=1}^{N} \sum_{i=1}^{L-1} \sum_{k=1}^{K} ||q_{n,i,k} - \hat{q}_{n,i,k}||,
\]  

(3.13)

where \( i \) is the frame index and \( k \) is the joint index. Since the first six degrees of freedom correspond to root translation and orientation, we extract the later 18 degrees of freedom for measuring MAE.

Mean per-joint position error (MPJPE) Lyu et al. [2022] measures the Euclidean distance between ground truth and simulated joint positions:

\[
\text{MPJPE} = \frac{1}{N L K} \sum_{n=1}^{N} \sum_{i=1}^{L-1} \sum_{k=1}^{K} ||x_{n,i,k} - \hat{x}_{n,i,k}||^2,
\]  

(3.14)

where \( x_{n,i,k} \in \mathbb{R}^3 \) is the global position of joint \( k \) at time \( t_i \) in gait sample \( n \), obtained from forward kinematics on the predicted pose \( q_{n,i,k} \).
Chapter 4

Results

4.1 Inertia Matrices Estimated with Constant Inertia Models

Fig. 4.1 shows effective inertia estimates from constant inertia models alongside with inertia computed by the ARB inertia model, visualized as box plots. To visualize a $d \times d$ inertia matrix we first accumulate entries along each row (i.e. lump to diagonal entries), then for each joint aggregate inertia of angles that belong to the joint with 2-norm. We can make several observations from the figure and see that inertia estimated by constant inertia models are physically plausible.

First, we can see from Fig. 4.1 (a), (b) that our estimated walking inertia is bilaterally symmetric. Furthermore when the ipsilateral foot is the left foot, inertia of joints on the left leg are larger than the inertia of their counterparts on the right leg; and when the ipsilateral foot is the right foot, right leg inertia is larger than their counterparts on the left leg. This pattern is also apparent in running samples, see Fig. 4.1 (c), (d). We hypothesis that this is due to the role of neural activation on inertia coupling. During walking, muscles of the stance leg before foot fall may be actuated to a greater amount than the muscles of the contralateral leg, increasing the effective inertia of the ipsilateral joints. Similarly, we expect muscles to be activated to a greater extent during running. This is consistent with the observation that the estimated joint inertia (Constant Inertia model) during running is consistently greater during running than during walking. More research to test this
hypothesis is still needed.

Second, if we look at effective inertia of distal joints, i.e. knee and ankle, we see that estimated inertia are in general larger than their counterparts constructed under the ARB assumptions. Most notably, ankle inertia is more apparent when estimated from data. This is consistent with results from Pai [2010]: when muscle mass is unlumped from bone mass the muscle increases inertia of joints distal to it. So the additional estimated inertia is due to shank muscles’ coupling to the ankle joint.

What the ARB inertia model fails to capture, and the constant inertia model is able to capture however is the dependence of effective inertia on motion type. From Fig. 4.2 we see that constant inertia’s magnitudes become larger once we switch motion from walking to running. Yet the ARB inertia model fails to capture such effects, as we can tell from the figure that inertia difference is minuscule and only exists due to small pose changes from walking to running. Although we can easily deduce that ARB inertia model cannot capture motion type dependence because it is not fitted on data and it is not parametrized by motion type, we cannot tell why the estimated running inertia are larger in magnitude, and understanding that requires further study.

4.2 Inertia Matrices Estimated with Neural Inertia Models

Fig. 4.3 compares effective inertia matrices estimated by SPDNet versus those constructed via the ARB inertia model. Again, we see patterns that imply physical plausibility: (i) larger estimated inertia than constructed ARB inertia on ankle; (ii) bilateral symmetry in magnitude of inertia, i.e. when a leg is in contact with ground, inertia on its joints are larger than inertia on their counterparts on the other leg. The symmetry is obvious for walking gaits, but in the case of running gaits, inertia on left-leg joints are larger even if the stance leg is the right leg; see Fig. 4.3 (d). The breaking of bilateral symmetry can be due to that we fitted a single SPDNet to both left and right foot-contact gait cycles. Since stance leg is not explicitly used as an input to the network in training, the predicted inertia can be averaged out by a bit. Comparing walking inertia with running inertia of
SPDNet in Fig. 4.4, we notice again that the gap between walking and running inertia would be larger if the inertia is estimated from data, than constructed under the ARB assumptions. But contrary to what we have seen in Fig. 4.2, SPDNet-predicted running inertia are lower in general. Knowing why we see this however would require further analysis.

Tab. 4.1 shows the magnitudes of motion errors from the constant inertia model, neural inertia models and the ARB inertia model in testing. We see that our SPDNet can generalize to walking gaits of unseen poses and mass distributions, and that SPDNet outperforms the ARB inertia model, the constant inertia model and other neural inertia models with lower motion reconstruction errors, on both the Walking Set and the Running Set. Also, our neural networks outperform the constant inertia model on test sets, as we expected, because a key advantage of the neural networks over the constant inertia model is that they are parmeterized by body mass, body height and poses, and thus can generalize to unseen body proportions and poses at test time. Fig. 4.5 visualizes the distribution of MAE and MPJPE of each model over six test subjects selected from the Walking Set and the Running Set. We can tell from the figure that SPDNet does not outperform other models simply because it does better on a few outlier clips. We believe the constant inertia model’s particularly bad performance is due to (i) the non-parametrized model’s inability to generalize to unseen body proportions and poses; (ii) the averaging effect. Note that the model is fitted to an entire training set consisting of gait samples from different subjects at different poses; see Sec. 3.3. So the estimated inertia can be averaged out such that it does not work well if used to simulate a particular subject’s motion with a specific initial pose. In Fig. 4.6 we visualize 150 ms (including 100 ms in the estimation window and 50 ms after the window) of joint angular trajectory reconstructed by SPDNet, the ARB inertia model alongside with ground truth trajectories. Notice that the ARB inertia model produces unstable motions when tested on either the Walking Set or the Running Set. On the other hand, SPDNet’s predicted angular trajectories are a lot closer to the ground truth than ARB inertia’s predicted trajectories on both the Walking Set and the Running Set, and its reconstructed motion remains stable until after the estimation window. So SPDNet’s predicted matrices can generalize well to reconstructing unseen motion for a short period of time outside window.
Turning to qualitative evaluation: Fig. 4.7 visualizes a motion clip reconstructed by different models. We see that skeletal motion reconstructed with SPDNet is the closest to the ground truth. But SymmNet and UnconNet fail to produce visually better results than the ARB inertia model. Also, the motion reconstructed with the constant inertia model is no longer stable after 50 ms, and in terms of visual closeness to ground-truth motion it is also worse than others.

The observation that SPDNet but not SymmNet or UnconNet beats the ARB baseline motivates us to look into the impact of architectural decisions on our neural networks’ performance. From Tab. 4.1 we notice that enforcing the predicted inertia matrix to be SPD yields much lower motion reconstruction errors than enforcing the matrix to be simply symmetric, or not enforcing constraints at all, either on the Walking Set or the Running Set. Our qualitative evaluation in Fig. 4.7 can also confirm this. On the other hand, from Tab. 4.1 we notice that enforcing symmetry alone does not grant much performance boost over UnconNet on the Walking Set and in fact, makes performance worse than UnconNet on the Running Set. We believe a part of the reason that contributes to the performance difference is that inertia matrices predicted by SymmNet or UnconNet have negative eigenvalues which lead to physically implausible behaviors. Looking at Fig. 4.8, we see that the inertia matrix predicted by SymmNet contains negative eigenvalues, and that the magnitudes of some of those negative eigenvalues are larger than the magnitudes of their counterparts produced by UnconNet. A large, negative eigenvalue can lead to problems when we use the matrix to solve forward dynamics. First, a force applied in the direction of the eigenvector corresponding to the eigenvalue can lead to a velocity change in the opposite direction, which violates Newton’s Second Law of Motion, see Fig. 4.9. Second, from Eq. 3.12 we can tell that if a predicted acceleration can be decomposed into a component vector that lies along the direction of the eigenvector (and another component that is in a different direction), the “eigen-component” can lead to a fairly large predicted force which amplifies loss. Moreover, we can tell from 4.8 that UnconNet does not naturally predict symmetric matrices because we can see complex eigenvalues, which are physically implausible. We can also see from the same figure that the Constant Inertia model produces matrices which contains slightly negative eigenvalues. This is not what we expected given the PSD enforcement in Eq. 3.3, and we believe this
is due to numerical errors in CVXPY [Diamond and Boyd, 2016].

Moving onto Tab. 4.2: here we compare the effects of different SPD layer options, including whether we use Cholesky Layer or Eigen Layer, and the positivity function [Jekel et al., 2022] we use. We notice that learning with SPDNet is only effective when we use Eigen layer and the ReLU positivity function. It makes sense that Eigen Layer outperforms Cholesky Layer, since Eigen layer preserves eigenbases learned by the MLP network from inputs, and such eigenbases are effectively latent vectors that incorporate physical and geometric properties of body segments. As far as using ReLU for positivity enforcement, we believe it is more effective than the Absolute Value function or the Square function when used with the Eigen layer because it preserves the ordering of eigenvalues Jekel et al. [2022], such that the MLP network is always penalized for predictions that lead to negative eigenvalues.
Figure 4.1: Distribution of effective inertia estimated by the constant inertia model vs computed by the ARB inertia model over all subjects. Each subject is (a) walking with left foot on ground; (b) walking with right foot on ground; (c) running with left foot on ground; (d) running with right foot on ground. The magnitudes are computed by diagonally lumping inertia matrices first, then aggregating over angles (displacements) associated with a joint (direction). We plot the distribution of magnitudes on log scale (left) and linear scale (right).
Figure 4.2: Distribution of walking vs running effective inertia with the constant inertia model and the ARB inertia model. (a) Constant inertia model’s predictions for left-foot contact gaits and (b) right-foot contact gaits. (c) ARB inertia model’s constructions for left-foot gaits and (d) right-foot gaits. The magnitudes are also computed by diagonal lumping then aggregated. We plot the distribution across all subjects on log scale (left) and linear scale (right).
Figure 4.3: Distribution of effective inertia estimated by SPDNet vs computed by the ARB inertia model over all subjects. Each subject is (a) walking with left foot on ground; (b) walking with right foot on ground; (c) running with left foot on ground; (d) running with right foot on ground. The magnitudes are also computed by diagonal lumping then aggregating, then plotted on log scale (left) and linear scale (right).
Figure 4.4: Distribution of walking vs running effective inertia with SPDNet and the ARB inertia model. (a) SPDNet’s predictions for left-foot contact gaits and (b) right-foot contact gaits. (c) The ARB inertia model’s constructions for left-foot gaits and (d) right-foot gaits. The magnitudes are also computed by diagonal lumping then aggregated. We plot the distribution across all subjects on log scale (left) and linear scale (right).
<table>
<thead>
<tr>
<th>Subj + Metric</th>
<th>SPDNet</th>
<th>SymmNet</th>
<th>UnconNet</th>
<th>Const. M</th>
<th>ARB M</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Walking Set MAE (rad, ↓), MPJPE (mm, ↓)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 MAE</td>
<td>0.019</td>
<td>0.118</td>
<td>0.170</td>
<td>0.189</td>
<td>0.661</td>
</tr>
<tr>
<td>MPJPE</td>
<td>21.4</td>
<td>121</td>
<td>106</td>
<td>169</td>
<td>226</td>
</tr>
<tr>
<td>03 MAE</td>
<td>0.031</td>
<td>0.133</td>
<td>0.137</td>
<td>0.180</td>
<td>0.332</td>
</tr>
<tr>
<td>MPJPE</td>
<td>57.9</td>
<td>115</td>
<td>110</td>
<td>102</td>
<td>195</td>
</tr>
<tr>
<td>05 MAE</td>
<td>0.031</td>
<td>0.158</td>
<td>0.207</td>
<td>1.02</td>
<td>0.538</td>
</tr>
<tr>
<td>MPJPE</td>
<td>50.9</td>
<td>105</td>
<td>132</td>
<td>619</td>
<td>228</td>
</tr>
<tr>
<td>09 MAE</td>
<td>0.034</td>
<td>0.085</td>
<td>0.114</td>
<td>0.214</td>
<td>0.476</td>
</tr>
<tr>
<td>MPJPE</td>
<td>80.2</td>
<td>73.6</td>
<td>103</td>
<td>338</td>
<td>213</td>
</tr>
<tr>
<td>17 MAE</td>
<td>0.0397</td>
<td>0.108</td>
<td>0.193</td>
<td>0.482</td>
<td>0.431</td>
</tr>
<tr>
<td>MPJPE</td>
<td>64.1</td>
<td>99.6</td>
<td>130</td>
<td>280</td>
<td>182</td>
</tr>
<tr>
<td>20 MAE</td>
<td>0.0435</td>
<td>0.127</td>
<td>0.227</td>
<td>0.345</td>
<td>0.250</td>
</tr>
<tr>
<td>MPJPE</td>
<td>43.5</td>
<td>127</td>
<td>227</td>
<td>345</td>
<td>250</td>
</tr>
<tr>
<td>Avg. MAE</td>
<td><strong>0.031</strong></td>
<td>0.124</td>
<td>0.190</td>
<td>0.414</td>
<td>0.475</td>
</tr>
<tr>
<td>Avg. MPJPE</td>
<td><strong>53.0</strong></td>
<td>107</td>
<td>135</td>
<td>309</td>
<td>216</td>
</tr>
<tr>
<td>(b) Running Set MAE (rad, ↓), MPJPE (mm, ↓)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03 MAE</td>
<td>0.019</td>
<td>0.334</td>
<td>0.118</td>
<td>1.30</td>
<td>0.137</td>
</tr>
<tr>
<td>MPJPE</td>
<td>27.8</td>
<td>329</td>
<td>161</td>
<td>707</td>
<td>84.5</td>
</tr>
<tr>
<td>04 MAE</td>
<td>0.020</td>
<td>0.789</td>
<td>0.067</td>
<td>1.54</td>
<td>0.536</td>
</tr>
<tr>
<td>MPJPE</td>
<td>28.9</td>
<td>518</td>
<td>72.1</td>
<td>687</td>
<td>146</td>
</tr>
<tr>
<td>10 MAE</td>
<td>0.0178</td>
<td>0.165</td>
<td>0.391</td>
<td>2.73</td>
<td>0.213</td>
</tr>
<tr>
<td>MPJPE</td>
<td>30.3</td>
<td>191</td>
<td>481</td>
<td>1669</td>
<td>128</td>
</tr>
<tr>
<td>17 MAE</td>
<td>0.026</td>
<td>0.511</td>
<td>0.697</td>
<td>0.939</td>
<td>0.208</td>
</tr>
<tr>
<td>MPJPE</td>
<td>26.1</td>
<td>395</td>
<td>436</td>
<td>1112</td>
<td>86.2</td>
</tr>
<tr>
<td>18 MAE</td>
<td>0.022</td>
<td>0.283</td>
<td>0.444</td>
<td>1.58</td>
<td>0.305</td>
</tr>
<tr>
<td>MPJPE</td>
<td>26.9</td>
<td>225</td>
<td>266</td>
<td>844</td>
<td>126</td>
</tr>
<tr>
<td>20 MAE</td>
<td>0.021</td>
<td>0.173</td>
<td>0.143</td>
<td>1.59</td>
<td>0.16</td>
</tr>
<tr>
<td>MPJPE</td>
<td>28.5</td>
<td>160</td>
<td>140</td>
<td>909</td>
<td>87.4</td>
</tr>
<tr>
<td>Avg. MAE</td>
<td><strong>0.021</strong></td>
<td>0.376</td>
<td>0.310</td>
<td>1.61</td>
<td>0.260</td>
</tr>
<tr>
<td>Avg. MPJPE</td>
<td><strong>28.1</strong></td>
<td>303</td>
<td>260</td>
<td>988</td>
<td>110</td>
</tr>
</tbody>
</table>

*Table 4.1:* Test set MAE and MPJPE of different models for inertia representation. We evaluated each model on six folds; for each fold we use gait cycles from one subject for testing. (a) Walking Set results. (b) Running Set results.
Figure 4.5: Box-and-whisker plots of metrics from reconstructing test set motion clips of the six folds by different inertia models: (a) Walking Set MAE and (b) MPJPE; (c) Running Set MAE and (d) MPJPE. Left: log scale, right: linear scale.

<table>
<thead>
<tr>
<th>Arch. (Layer, Positivity Function)</th>
<th>MAE (rad, ↓)</th>
<th>MPJPE (mm, ↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPDNet (Cholesky, ReLU)</td>
<td>3.74</td>
<td>2165</td>
</tr>
<tr>
<td>SPDNet (Cholesky, Abs. Value)</td>
<td>3.82</td>
<td>2968</td>
</tr>
<tr>
<td>SPDNet (Cholesky, Square)</td>
<td>4.59</td>
<td>2039</td>
</tr>
<tr>
<td>SPDNet (Eigen, ReLU)</td>
<td>0.031</td>
<td>49.4</td>
</tr>
<tr>
<td>SPDNet (Eigen, Abs. Value)</td>
<td>1.27</td>
<td>814</td>
</tr>
<tr>
<td>SPDNet (Eigen, Square)</td>
<td>4.40</td>
<td>1536</td>
</tr>
<tr>
<td>SymmNet</td>
<td>0.135</td>
<td>120</td>
</tr>
<tr>
<td>UnconNet</td>
<td>0.192</td>
<td>238</td>
</tr>
<tr>
<td>Const. M</td>
<td>0.808</td>
<td>428</td>
</tr>
<tr>
<td>ARB</td>
<td>0.475</td>
<td>216</td>
</tr>
</tbody>
</table>

Table 4.2: Mean test set MAE (↓) and MPJPE (↓) of walking gait inertia estimation from different SPDLayer configurations. We also list the MAE and MPJPE from SymmNet, UnconNet, Constant Inertia Model and the ARB Inertia Model for comparison.
Figure 4.6: Joint angles’ mean ± one standard deviation vs time from motion clips reconstructed by different models. The joint angles are grouped by anatomical planes they are situated in: e.g. the hip Sagittal angle is the hip-x axis angle shown in Fig. 3.1 (c), and the knee Horizontal angle is the knee-y axis angle shown in Fig. 3.1 (c). An empty plot in the grid means that angle was not measured in the dataset provided by [Zell et al., 2020].
Figure 4.7: Qualitative comparison of reconstructed motion between neural networks, the constant inertia model and the ARB inertia model. The ground truth motion is measured on Subject No. 17 while running, during a stance phase with right foot staying in contact with the ground. We compare at three time instances: 10 ms, 50 ms, 90 ms.
Figure 4.8: Eigenvalues of inertia matrices, either constructed or predicted on Subject No. 17 from a running clip with right foot contacting ground. We can tell SPDNet guarantees eigenvalues of predicted matrix being positive. Note that we set the minimum acceptable eigenvalue to be $1^{-10}$ but that value is too small to be printed out in the figure.
Figure 4.9: An inertia matrix with negative eigenvalues can make the simulated human produce physically implausible behaviors. Here to the simulated human we apply an eigenforce, i.e. an impulse force in the same direction as an eigenvector corresponding to a negative eigenvalue of the system’s predicted inertia matrix. (a) The generalized unitary eigenforce applied to the system. (a) The system’s responding acceleration $\ddot{q}$. We see the applied force and the induced acceleration are in opposite directions, and that is not physically plausible.
Chapter 5

Conclusions and Future Work

We developed two models, the Constant Inertia Model and SPDNet, for representing subject-specific effective joint-space inertia matrix of the lower body of a human in everyday walking and running. The estimated inertia matrices do not make use of the rigidity assumptions aimed for approximating physical and geometric properties of human body segments, incorporate effects of inertial coupling and are motion type dependent. Also, unlike prior models that were analytically constructed based on the ARB assumptions, both models are data-driven and need to be regressed from kinematic mocap data, dynamic force plate data as well as the body heights and masses of human subjects. While the Constant Inertia Model is non-parametric and thus should be estimated per-subject, per-pose, SPDNet is parametrized by pose, body height and mass, and it can generalize well to unseen poses and body proportions. Experimental results have shown that inertia matrices predicted by SPDNet are symmetric and positive definite, and that makes such matrices physically plausible.

Compared with some prior work in biomechanics which sought only to estimate individual BSIPs, our work seeks to estimate joint-space inertia matrices that can be injected into a simulation framework for reconstructing lower-body motion of a human character that were not seen by our inertia model in training. We injected matrices predicted by SPDNet into a differentiable simulator to reconstruct motion in the first 140 ms of the stance phase in a gait cycle, either when a subject is walking or running. We then showed that SPDNet’s matrices yield much lower
motion errors (MAE and MPJPE) than ARB inertia matrices. This result confirms that SPDNet is able to capture non-rigid properties of body segments as well as inertial effects of a human in motion much better than the ARB inertia model, when musculocontrol is constant.

Despite what our work has achieved, it still has many limitations. First of all, we have yet to explain the difference between estimated and running inertia. Second, character control which is important for analyzing and constructing motion in biomechanics contexts [Clancy et al., 2023, Lam and Vujaklija, 2021, Ren et al., 2007, Xiang et al., 2011, Zell et al., 2020] or driving complex human motions for graphical applications [Bergamin et al., 2019, Park et al., 2022, Shimada et al., 2021] is completely absent from our discussion. This limits our estimation and motion reconstruction to the initial single support phase of a gait cycle, since after more time elapses active musculoskeletal control and the joint torque it induces would be more significant, hence can no longer be ignored. We can implement a simple PD controller as done in prior works on graphics [Shimada et al., 2021], or resort to more complex activation-based controllers used to drive musculoskeletal models [Park et al., 2022]. Another aspect we can improve is extending our method to work on purely kinematic datasets which are generally larger and more abundant. This may involve training our neural networks on motion loss rather than per-step impulse loss which requires force plate data as an input. Finally, we believe interpreting our estimated inertia in terms of segment-wise mass distribution and geometry is possible, and more work can be done to show that we can reconstruct such information from inertia matrices and hence, generate more visually and anatomically realistic human characters.
Bibliography


41


G. Yang, S. Yang, J. Z. Zhang, Z. Manchester, and D. Ramanan. Ppr: Physically plausible reconstruction from monocular videos. In *Proceedings of the*


Appendix A

Preliminary Experiments on the Synthetic Dataset

Before training and testing our methods on the real world dataset by Zell et al. [2020], we created simple two DoF leg models in MuJoCo [Todorov et al., 2012], generated synthetic datasets of leg-ground contact clips from them, and performed inertial estimation on them using our methods. This allows us to know if the constant inertia model or neural inertia model can attain sensible estimations at least on systems that are simple and easy to debug, before trying them on a 24-DoF system that is more complex.

A.1 Synthetic Dataset Construction

We created two simple rigid double pendulum models to emulate the motion of a human leg. Each model consists of two capsules and two revolute joints; the top capsule emulates a thigh with mass set to 0.7 kg; the bottom capsule emulates a shank with mass set to 0.3 kg. The joints respectively emulate a hip and a knee. Both models have two degrees of freedom. The tendon-actuated model is a derivative of MuJoCo’s built-in model with six muscle-tendon pairs; we mapped each pair to a human leg muscle, see Fig. A.1 (b). To emulate GRF/M generated when a human leg touches ground, we make each pendulum contact the ground at a non-zero initial velocity starting from time zero, then sample state trajectory
and contact forces following the impact. Since mass matrix depends on pose, we make each pendulum contact the floor at different poses by (i) either directly setting the initial pose for the unactuated model, (ii) or defining different activation levels for muscle-tendons and waiting until the tendon-driven model settles to equilibrium poses. To make impact responses from the same initial pose not identical to each other, we vary the impact velocity and tilt the pendulum’s plane of motion so impacts take place at different tilt angles, see Fig. A.1 (a). Thus, using each model we can create two datasets: one contains simulation clips sampled at different poses (the Pose Set), another contains clips sampled at the same pose, but with segment masses being different in each simulation (the Mass Set). For the unactuated model’s Pose Set, we sampled 11,664 clips in total at 81 poses, so each pose subset contains 114 clips. For the tendon-actuated model’s Pose Set, we sampled 3645 clips in total also at 81 different equilibrium poses, corresponding to 81 muscle activation combinations. And for the tendon-actuated model’s Mass Set we sampled data from 81 different thigh-shank mass combinations.

A.2 Synthetic Data Fitting
The constant inertia model we used for the preliminary experiments is smaller than the one fitted on the real-world dataset: it contains only 4 parameters since the joint space inertia matrix of the 2-DoF tendon-driven leg model has shape $4 \times 4$. We also fit it with constrained least squares. The neural inertia model we used is less developed than SPDNet: it directly predicts inertia from pose $q$ and total body mass $m$, without using a person feature extractor to build latent vector $p$ first. The network contains fewer layers and fewer neurons per layer, since the matrix to be predicted is smaller in dimension. The SPD Layer uses Cholesky Layer. We use the $r^2$ metric to evaluate the quality of impulse fit.

We see from Fig. A.2 (a), (b) that constant inertia models which are fitted per-pose, per-mass can attain $r^2$ scores close to 1 on most test subsets. From Fig. A.2 (c), (d) we see that the neural network can obtain high $r^2$ scores on most poses or masses, despite low performance on several subsets.
Figure A.1: Synthetic dataset construction and partitioning. (a) We create two leg models to generate synthetic data from: an unactuated double pendulum and a tendon-driven double pendulum. With the unactuated pendulum, we generate data at different initial poses to obtain different effective inertia matrices. For the tendon-actuated model, we apply different muscle activations so the initial pose would be different. We also set the pendulum’s plane of motion at different tilt angles w.r.t. ground in order to diversify our data. (b) Musculotendious actuators that map to real-world leg muscles.
Figure A.2: $R^2$ scores of effective inertia fits, on synthetic data from the tendon-actuated double pendulum. (a) - (b): Fitting constant inertia models on the Pose, Mass Set. (c) - (d): Fitting Neural Network on the Pose, Mass Set.