PUBLICATIONS OF
THE UNIVERSITY OF EASTERN FINLAND

Dissertations in Science, Forestry and Technology

UNIVERSITY OF
EASTERN FINLAND

JERE LAVIKAINEN

ESTIMATION METHODS FOR MUSCULOSKELETAL MOVEMENT AND LOADING USING BIOMECHANICAL MODELS AND PORTABLE MODALITIES

BRINGING MODERN MUSCULOSKELETAL ANALYSIS
METHODS OUTSIDE THE MOTION LABORATORY

PUBLICATIONS OF THE UNIVERSITY OF EASTERN FINLAND
DISSERTATIONS IN SCIENCE, FORESTRY AND TECHNOLOGY

N:o 50

Jere Lavikainen

ESTIMATION METHODS FOR MUSCULOSKELETAL MOVEMENT AND LOADING USING BIOMECHANICAL MODELS AND PORTABLE MODALITIES

BRINGING MODERN MUSCULOSKELETAL ANALYSIS METHODS OUTSIDE THE MOTION LABORATORY

ACADEMIC DISSERTATION

To be presented by the permission of the Faculty of Science, Forestry and Technology for public examination in the Auditorium SN201 in Snellmania Building at the University of Eastern Finland, Kuopio, on August 16th, 2024, at 12 o'clock.

University of Eastern Finland
Department of Technical Physics
Kuopio 2024

PunaMusta Oy
Joensuu, 2024
Editors: Pertti Pasanen, Nina Hakulinen, Raine Kortet,
Matti Tedre and Jukka Tuomela

ISBN: 978-952-61-5275-2 (print)
ISBN: 978-952-61-5276-9 (PDF)
ISSNL: 2954-131X
ISSN: 2954-131X (print)
ISSN: 2954-1484 (PDF)

Author’s address: University of Eastern Finland
Department of Technical Physics
Kuopio, Finland
email: jere.lavikainen@uef.fi

Supervisors: Professor Pasi Karjalainen
University of Eastern Finland
Department of Technical Physics
Kuopio, Finland
email: pasi.karjalainen@uef.fi

Professor Rami Korhonen
University of Eastern Finland
Department of Technical Physics
Kuopio, Finland
email: rami.korhonen@uef.fi

Adjunct Professor Mika Mononen
University of Eastern Finland
Department of Technical Physics
Kuopio, Finland
email: mika.mononen@uef.fi

Adjunct Professor Lauri Stenroth
University of Eastern Finland
Department of Technical Physics
Kuopio, Finland
email: lauri.stenroth@uef.fi

Project Researcher (PhD) Paavo Vartiainen
University of Eastern Finland
Department of Technical Physics
Kuopio, Finland
email: paavo.vartiainen@uef.fi

Reviewers: Senior Lecturer Steffi L. Colyer
University of Bath
Department for Health
Bath, United Kingdom
email: S.Colyer@bath.ac.uk

Assistant Professor Arielle G. Fischer
Technion, Israel Institute of Technology
Department of Biomedical Engineering
Haifa, Israel
email: ariellef@technion.ac.il

Opponent: Professor Anthony M. J. Bull
Imperial College London
Department of Bioengineering
London, United Kingdom
email: a.bull@imperial.ac.uk

Jere Lavikainen
Estimation methods for musculoskeletal movement and loading using biomechanical models and portable modalities — bringing modern musculoskeletal analysis methods outside the motion laboratory
Kuopio: University of Eastern Finland, 2024
Publications of the University of Eastern Finland
Dissertations in Science, Forestry and Technology: 50

ABSTRACT

Mechanical loading of the knee joint is connected to pathologies involving structural degeneration of the knee, including knee osteoarthritis. Information of knee joint loading during, e.g., walking, could help in planning more sustainable movement strategies in rehabilitation and in predicting further degeneration of the knee joint. Although knee joint loading cannot be directly measured *in vivo* without invasive surgery, musculoskeletal modeling provides a way to estimate biomechanically informed knee joint loading through simulations of human movement. However, these simulations are not routinely available for clinical use because they require a trained expert to measure the subject in a motion laboratory and to conduct the simulations and other analyses. Additionally, the simulations can be cumbersome to conduct.

Machine learning algorithms, especially artificial neural networks, have been proposed to eliminate the simulation phase and to estimate knee joint loading directly from experimental motion data. However, existing solutions still either utilize motion capture data or are designed for specific portable measurement technologies. For out-of-laboratory environments such as clinics, a technology-agnostic solution that estimates knee joint loading from a simple set of inputs is required.

This doctoral work presents modular artificial neural network models for estimating knee joint loading peaks from simple input data that can be retrieved from a multitude of portable modalities, such as a video camera or inertial measurement units. It also presents a method for computing the kinematics of motion in real-time from inertial measurement units. These methods are a step towards enabling biomechanically informed estimates of human kinematics and kinetics effortlessly outside the motion laboratory, e.g., in clinical environments.

In study **I**, a software library was developed to enable the inverse kinematics of OpenSim in real-time for inertial measurement units. Throughput and delay of inverse kinematics calculations during walking were quantified using commonly used musculoskeletal models. The software library allows real-time analysis of kinematics using wearable motion sensors, thus allowing immediate information about kinematics outside the motion laboratory. Both the code and the publication are available open access.

In study **II**, artificial neural networks were trained to predict knee joint contact force peaks during walking. The networks were trained on data from existing motion capture data sets. The networks were capable of predicting knee joint contact force peaks from predictor sets of simple input data. Although the predictor set was retrieved from motion capture data, it was hypothesized that the data could be calculated without any motion capture data, which would enable estimation of knee joint loading peaks without the need for measurements in a motion laboratory.

In study **III**, inertial measurement units and computer vision-based human pose estimation from video camera data were separately used to retrieve a simple predictor set for estimation of knee joint loading. The predictors were fed to artificial neural networks that predicted knee joint loading peaks. It was shown that no motion capture data is required to estimate knee joint loading peaks outside the motion laboratory. The developed methods could enable biomechanically informed treatment decisions, rehabilitation planning and prevention of disease progression in, e.g., a physician's office using a simple webcam or wearable motion sensors.

National Library of Medicine Classification: *WE 300, QT 34.5, QS 26.5*
Medical Subject Headings: *Knee Joint / physiology; Weight-Bearing / physiology; Kinetics; Biomechanical Phenomena; Neural Networks, Computer; Musculoskeletal System; Computer Models; Computer Simulation*
Yleinen suomalainen ontologia: *polvinivel; rasitus; neuroverkot; liikeoppi; dynamiikka; tuki- ja liikuntaelimet; mallintaminen; simulointi*

ACKNOWLEDGEMENTS

I'd like extend my gratitude to my supervisors Pasi, Rami, Mika, Lauri, and Paavo. Special thanks go to Lauri, who often provided invaluable help with musculoskeletal modeling, Mika, without whose funding assistance my work in UEF would have likely been cut short, and Paavo, who was often present to help with preparing the motion laboratory and troubleshooting issues.

I am grateful for the funding provided by the European Regional Development Fund, Research Council of Finland, EIT Digital, Ministry of Social Affairs and Health, and Sigrid Juselius Foundation.

I want to thank Will and Kim for peer support; Heta, Jiri, Petri, and others whose room I would occasionally intrude in to ask about funding applications and similar necessary by-products of doctoral work; Amir, Joose, Rasmus, and tens of others who contributed in some way to my doctoral journey.

I also wish to thank my partner Maria, who has supported me throughout my doctoral work. Above all, I owe a debt of gratitude to my schnauzer Cirdan, whose input provided mental support and made sure that I took frequent breaks from writing this dissertation.

Kuopio, January 17, 2024

Jere Lavikainen

LIST OF PUBLICATIONS

This thesis consists of the present review of the author's work in the field of human motion analysis and the following selection of the author's publications:

- **I** J. Lavikainen, P. Vartiainen, L. Stenroth, P. A. Karjalainen, "Open-source software library for real-time inertial measurement unit data-based inverse kinematics using OpenSim," *PeerJ – Life and Environment* (2023).
- **II** J. Lavikainen, L. Stenroth, T. Alkjær, P. A. Karjalainen, R. K. Korhonen, M. E. Mononen, "Prediction of knee joint compartmental loading maxima utilizing simple subject characteristics and neural networks", *Ann Biomed Eng*, **51**, 2479–2489 (2023).
- **III** J. Lavikainen, L. Stenroth, P. Vartiainen, T. Alkjær, P. A. Karjalainen, M. Henriksen, R. K. Korhonen, M. Liukkonen, M. E. Mononen, "Predicting knee joint contact force peaks during gait using a video camera or wearable sensors", *currently under peer review in Ann Biomed Eng*.

Throughout the overview, these papers will be referred to by Roman numerals.

AUTHOR'S CONTRIBUTION

The publications selected in this dissertation are original research papers on human motion analysis and how it could be brought outside the motion laboratory. The author was the main contributor to all studies. The author participated in study planning and musculoskeletal analysis in studies **I**, **II**, and **III**, developed the software and conducted the measurements and analysis in study **I**, trained and evaluated the artificial neural networks in studies **II** and **III**, and collected the motion data and magnetic resonance images and designed and conducted the analysis of video camera and inertial measurement unit data in study **III**. Unlike in studies **I** and **III**, the author did not collect the motion data in study **II**. The author was the main writer in all studies.

Glossary

This dissertation describes objects and methods that are used by a variety of different fields and backgrounds. There is no standardized nomenclature shared by all fields, and different sources use the same terminology differently. To mitigate this confusion, the terminology of this dissertation will follow the definitions below. We do not claim that these are the only correct definitions.

backpropagation an algorithm for calculating the gradient of the loss function in a neural function by moving backward along the layers of the network.

deep learning a subset of machine learning that involves artificial neural networks that can form layers of increasingly complicated features from simpler ones.

external force a force that is applied between a physical system and its surroundings.

feature information in data that can be used to relate variables to one another; can be latent or explicit; used by some sources as an alias to predictor.

feature map the output image produced by a kernel when convolved over an input image in convolutional neural networks.

filter in machine learning, a feature detector in a convolutional neural network that comprises the kernels for each channel; in signal analysis, an algorithm that removes noise or frequency content from a signal.

forecasting the task of estimating the values of a time series from previous values of the time series; subset of prediction.

gradient descent an optimization algorithm for finding the minimum of a function; in deep learning, utilized often in backpropagation training.

inference study of the effect of the parameters of a model to the output of the model.

internal force a force that is applied between the components of a physical system, such as the human body.

kernel a matrix containing the weights of a feature detector in a convolutional neural network; kernels of each channel (e.g., RGB color) constitute a filter.

knee joint contact force the physical force exerted on the articular surfaces of the knee joint as a result of external forces (e.g., ground reaction force) and internal forces (e.g., muscle contraction forces); in this work, we focus particularly on the compressive tibiofemoral contact force.

machine learning the use of computational algorithms that can adjust their own parameters and extract patterns from data to solve problems.

node a single functional unit in an artificial neural network that processes the linear combination of weights and inputs and a bias term in a nonlinear function and produces a scalar output; analogous to a biological neuron.

prediction the task of estimating the output of a model from input samples or the output that an artificial neural network gives during a prediction task.

predictor a variable that is used as input in supervised learning tasks.

regression class of algorithms where the parameters of a statistical model are estimated; in machine learning, used by some sources as an alias for prediction.

response the variable that is estimated in prediction tasks; note that the estimate given by the prediction model is the prediction, not the response.

supervised learning the use of labeled datasets to train machine learning algorithms.

tensor a multidimensional array whose rank describes its dimensionality; a scalar is a zero-rank tensor, a vector a first-rank or one-dimensional tensor, and a matrix a second-rank or two-dimensional tensor.

Acronyms

ACL anterior cruciate ligament.
ANN artificial neural network.
API application programming interface.

BMI body mass index.

CNN convolutional neural network.
COP center of pressure.

DL deep learning.

FNN feedforward neural network.

GRF ground reaction force.

HPE human pose estimation.

ICD intercondylar distance.
IMU inertial measurement unit.

KFA knee flexion angle.
KJCF knee joint contact force.

LCL lateral collateral ligament.
LS least squares.

MCL medial collateral ligament.
ML machine learning.
MOCAP motion capture.
MS musculoskeletal.

NRMSE normalized root mean square error.

PCL posterior cruciate ligament.

ReLU rectified linear unit.

RMSE	root mean square error.
SDK	software development kit.
STD	standard deviation.
VC	video camera.
VRAM	video random-access memory.

TABLE OF CONTENTS

1 Introduction 1

2 Knee joint 3
- 2.1 Structure and function 3
- 2.2 Pathologies and knee joint loading 5

3 Motion analysis 7
- 3.1 Motion tracking technologies 7
  - 3.1.1 X-ray imaging of motion 7
  - 3.1.2 Marker-based motion capture 8
  - 3.1.3 Force plates 10
  - 3.1.4 Wearable motion tracking devices 11
  - 3.1.5 Video analysis and markerless motion capture 13
- 3.2 Pre-processing of motion data 14
- 3.3 The human gait cycle 17
- 3.4 Musculoskeletal modeling and simulation 18
  - 3.4.1 Kinematics and kinetics of motion 19
  - 3.4.2 Modeling and simulation workflow 19
  - 3.4.3 OpenSim 22
- 3.5 Challenges in estimating knee joint loading 23

4 Artificial neural networks 27
- 4.1 Basics of artificial neural networks 27
- 4.2 Feedforward neural networks 28
- 4.3 Convolutional neural networks 29
  - 4.3.1 Computational efficiency 31
  - 4.3.2 Layer structure 32
- 4.4 Activation functions 33
- 4.5 Training of artificial neural networks 34
  - 4.5.1 General principles 34
  - 4.5.2 Training, validation, and test sets 35
  - 4.5.3 The backpropagation algorithm 36
- 4.6 Neural networks in motion analysis 38
  - 4.6.1 Human pose estimation with convolutional neural networks 39
  - 4.6.2 Neural networks in estimating joint dynamics 42

5 Aims 45

6 Materials and methods 47
- 6.1 Motion laboratory 47
  - 6.1.1 Motion capture equipment 47
  - 6.1.2 Inertial measurement units 47
- 6.2 Magnetic resonance imaging for calculating knee intercondylar distance 48

6.3 Collection of the Kuopio gait dataset ... 48
6.3.1 Marker and IMU placement ... 50
6.3.2 Walking measurements ... 51
6.4 Collection of existing gait datasets ... 52
6.5 Enabling real-time processing of inverse kinematics with OpenSim ... 53
6.6 Musculoskeletal modeling and simulation ... 54
6.7 Designing the model for predicting knee joint loading ... 55
6.7.1 Response variables ... 56
6.7.2 Predictor selection ... 56
6.7.3 Constructing the prediction model ... 58
6.7.4 Evaluating the prediction model ... 59
6.8 Estimating predictors from video data ... 59
6.8.1 Identifying keypoints from video frames ... 59
6.8.2 Correcting artefacts in keypoint data ... 60
6.8.3 Estimating walking speed using keypoint trajectories ... 61
6.8.4 Estimating knee flexion angle using keypoint trajectories ... 64
6.9 Estimating predictors from inertial measurement unit data ... 64
6.9.1 Estimating walking speed using the orientation data of sensors 66
6.9.2 Estimating knee flexion angle using orientation data of sensors 66
6.10 Post-study analyses ... 66
6.10.1 Correlation analysis between kinematics and knee joint contact forces ... 67
6.10.2 Multiple linear regression of knee joint loading peaks ... 68
6.10.3 Nonlinear effect of predictors on knee joint loading peaks ... 69

7 Results **73**
7.1 Performance of computing real-time inverse kinematics ... 73
7.2 Musculoskeletal simulation-estimated knee joint contact force peaks . 73
7.3 Estimating the walking speed and knee flexion angle predictors ... 73
7.4 Predicting knee joint contact force peaks ... 75
7.4.1 Nonlinear effect of predictors on knee joint contact force peaks 76
7.5 Linear correlation between kinematics and knee joint contact force peaks ... 77
7.6 Multiple linear regression of knee joint loading peaks ... 78

8 Discussion **81**
8.1 Real-time inverse kinematics ... 81
8.2 Validity of estimated knee joint contact force ... 81
8.3 Estimating predictors from portable modalities ... 82
8.3.1 Walking speed from IMU data ... 82
8.3.2 Walking speed from video camera data ... 82
8.3.3 Knee flexion angle from IMU data ... 83
8.3.4 Knee flexion angle from video camera data ... 84
8.4 Predicting knee joint loading ... 85
8.4.1 Practical feasibility of predicting knee joint contact force peaks 85
8.4.2 Clinical potential of the methods ... 86
8.4.3 Defining the criteria for prediction error ... 88
8.4.4 Modeling the relationship between knee joint loading and simple predictors ... 89
8.4.5 Comparison to multiple linear regression ... 90

8.4.6 Nonlinear effect of predictors on knee joint contact force peaks 91
8.4.7 Our studies among the existing corpus of machine learning in biomechanics 91
8.5 Linear correlation between kinematics and knee joint contact forces .. 92
8.6 Future directions 93

9 Summary and conclusions **95**

BIBLIOGRAPHY **97**

1 Introduction

Knee is an important joint that facilitates most of human locomotion [1,2]. It is subjected to heavy physical loading, generated not only from the weight of the body but the effects of soft tissues like muscle contraction as well [3–7]. This physical loading is believed to be linked to structural degeneration of the joint, e.g., cartilage damage [7–11]. Therefore, ways to quantify the loading exerted on the joint could help in preventing the onset and progression of knee pathologies and assist in physical rehabilitation to reduce abnormally high knee loading [12].

Knee joint loading is difficult to measure because all available technologies for its direct measurement are invasive [6]. The physical measurement of knee joint loading requires cutting through live tissue to place sensors, which is currently only feasible with some knee implants. However, knee implants are only placed in surgical total knee replacement operations, where the original knee joint is already so damaged it has to be replaced [13, 14]. Thus, knowledge of knee joint loading after the surgery is too late to help in analyzing the onset or progression of structural degeneration of the joint. Furthermore, gait involving implanted knees may not represent healthy gait because of other gait-affecting conditions surrounding the cause of the knee degeneration, and because the implant cannot perfectly replicate the original knee in its structure and function [14]. Therefore, knowledge of knee joint loading in healthy subjects before the development of any pathology is required.

Knee joint loading can be computationally estimated with musculoskeletal modeling and simulation [6]. It involves measuring the movement and the ground reaction forces related to it, creating a subject-specific computational model of the subject's musculoskeletal system, simulating the movement using the model and the collected data, and extracting the forces exerted to the knee joint of the model in the simulation [6,15,16]. Although its validity is based on how well the musculoskeletal model represents the subject and the quality of the measured data, musculoskeletal modeling and simulation is noninvasive and therefore, the best available option for estimating knee joint loading.

There are two factors that limit the accessibility of musculoskeletal modeling and simulation. First, measuring movement and ground reaction forces requires a specialized motion laboratory with expensive motion capture equipment and software [17]. Second, musculoskeletal modeling and simulation is a computationally demanding workflow that is too slow in use cases where kinematic or kinetics outputs are needed immediately, such as in real-time feedback for rehabilitation [18–20]. Furthermore, an expert must conduct the measurements and analyze the data [21].

Enabling accurate measurement of motion outside the motion laboratory would help in bringing biomechanically relevant information to routine use in clinics, home environments, and other out-of-laboratory settings [22–24]. To make the estimation of knee joint loading accessible in routine clinical contexts, technologies to measure and analyze motion must be usable in an office environment and without special expertise. Only then estimates of knee joint loading could be used to make risk assessments of loading-related conditions [25], guide rehabilitation exercises and gait retraining [26], and develop predictive models for the onset and progression of joint diseases [27] in a clinical setting.

Inertial measurement units and video cameras represent an alternative to motion capture systems for measuring human motion [17,25,28–32]. Inertial measurement units are wearable multi-sensor devices that can be used to track the kinematics of body segments. They transmit the data wirelessly to a computer. Video cameras are another alternative to motion capture systems. Modern computational algorithms can be used to extract motion data comparable to 2D marker trajectories of motion capture systems from video feed [33–36]. While inertial measurement units and video cameras are portable and inexpensive compared to a fully equipped motion laboratory, their data is considered less accurate and they cannot measure ground reaction forces directly [22,23,37]. Thus, their accessibility comes at a cost that must be overcome to enable their clinical use.

The increasingly more widespread use of machine learning algorithms has presented solutions to improve upon the disadvantages of portable measurement modalities [38]. Computer vision algorithms can automatically extract anatomical landmarks from video data [39–42], and some can even estimate their depth from a 2D image [43,44]. Algorithms have been developed to estimate ground reaction forces from the kinematic data that can be extracted from inertial measurement units or video frames [17]. Additionally, the function approximation capability of feedforward artificial neural networks has been utilized to estimate the biomechanical output parameters of the musculoskeletal modeling and simulation, enabling analysis in real-time without domain knowledge [18–20]. These machine learning solutions have the potential to make biomechanical evaluation of knee joint loading during dynamic motion available in routine out-of-laboratory, possibly even clinical, use.

The purpose of this thesis is to utilize portable modalities for computing real-time kinematics of motion and for estimating knee joint loading peaks. Our aim is to leverage existing musculoskeletal modeling and simulation software for calculating real-time kinematics and for predicting knee joint loading peaks from simple input variables. Those input variables are designed to be obtainable from portable modalities such as inertial measurement units or video cameras. We envision a workflow where anyone can use an ordinary webcam or wearable sensors to measure a person's walking technique and retrieve immediate personalized estimates of knee joint loading for that person.

2 Knee joint

The knee joint is a pivotal structure to human locomotion. It transmits great physical loads and actually comprises two joints, the patellofemoral joint and the tibiofemoral joint. The knee joint is often characterized as a hinge joint, although its structure is actually somewhat more complex than that. In this chapter, we present the structure and function of the knee joint, the clinical relevance of its pathologies, and the importance of physical loading in evaluating the health of the knee.

2.1 STRUCTURE AND FUNCTION

The knee joint is the largest joint in the human body [1]. It actually comprises two joints, the tibiofemoral joint and the patellofemoral joint [1,2]. The patellofemoral joint connects the patella and the femur, while the tibiofemoral joint connects the tibia and the femur. The latter is a bicondylar joint where two condyles of the femur articulate with two corresponding condyles on the tibia. These joints and their function involve four different bones, articular cartilage, four major ligaments, tendons, muscles, various other soft tissues such as bursae, and in the case of the tibiofemoral joint, fibrocartilage such as menisci [45].

The bones involved in the function of the knee joint are the femur, the patella, the tibia, and the fibula (Figure 2.1) [45]. The femur and the tibia have the most important roles in the general function of the knee, while the patella facilitates knee extension, protects the knee joint and provides an attachment surface to soft tissues. The fibula does not connect to the knee joint itself, but attaches to the tibia via the superior tibiofibular joint; however, it still plays a role in the function of the knee joint by providing an attachment surface for muscles, ligaments, and tendons [2]. Articular cartilage coats the surfaces of the femoral and the tibial condyles, as well as the surfaces between the patella and the femur [45].

In the tibiofemoral joint, there are two fibrocartilage structures known as menisci [1,2,45] (Figure 2.2). They lie between the femoral and the tibial condyles [45] and their function is to make the otherwise fairly flat articular surface of the tibia deeper to enhance support between the femur and the tibia, to transmit axial forces in the knee, and to provide shock absorption [1,2].

The four major ligaments of the knee joint are the medial collateral ligament (MCL), the lateral collateral ligament (LCL), the anterior cruciate ligament (ACL), and the posterior cruciate ligament (PCL) [45]. The MCL runs approximately from the medial epicondyle of the femur to the medial condyle and superior medial surface of the tibia [2]; the LCL from the lateral epicondyle of the femur to the lateral side of the head of the fibula [2]; the ACL from the posterior part of the lateral femoral condyle to the anterolateral aspect of the medial intercondylar tibial spine [46]; and the PCL from the lateral aspect of the medial femoral condyle to the posterior surface of the tibia between the two posterior horns of the menisci [2]. The MCL mainly resists forces that would twist the tibia peripherally (away from the center of the body, causing knee valgus), while the LCL mainly resists forces that would twist the tibia centrally (towards the center of the body, causing knee

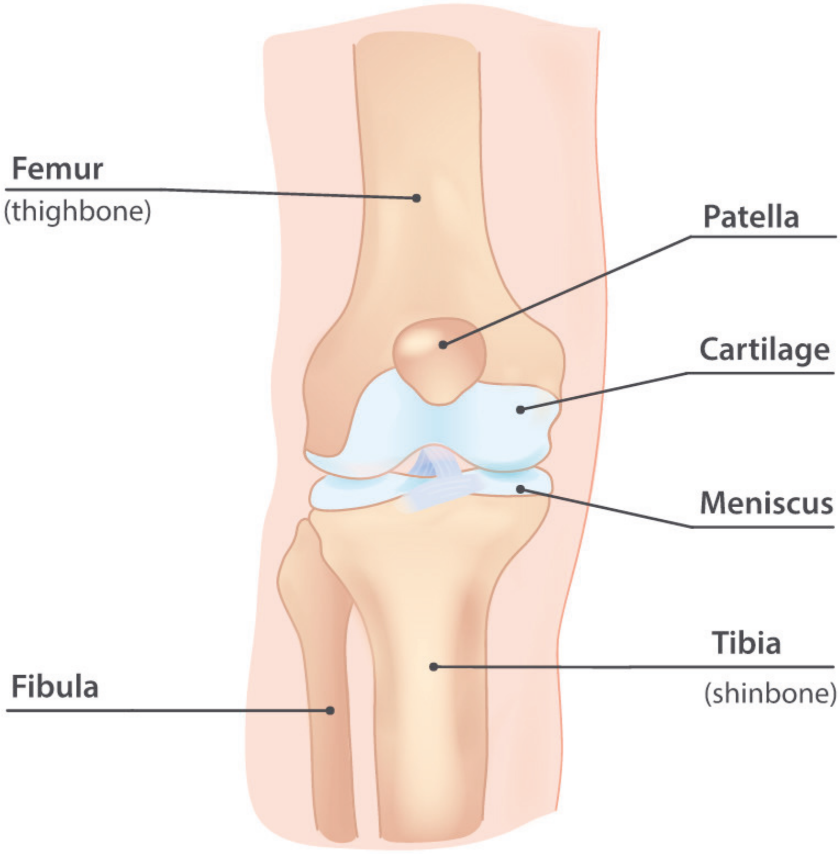

Figure 2.1: Bony and some cartilaginous anatomy of the knee joint. Image taken unedited from `https://www.injurymap.com/free-human-anatomy-illustrations` under the CC BY 4.0 License.

varus) [2]. The ACL resists anterior translation of the tibia with respect to the femur [2,46], while the PCL resists posterior translation of the tibia with respect to the femur [2]. The ligaments also contribute to resisting other forces in smaller quantities [46]. While the aforementioned four ligaments support the function of the tibiofemoral joint, other ligaments also support the knee; in the patellofemoral joint, these include the medial [47] and lateral [48] patellofemoral ligaments. Together, ligaments provide stability to the knee joint as a whole [2].

The muscles quadriceps femoris, semitendinosus, semimembranosus, gracilis, sartorius, popliteus, tensor fasciae latae, gastrocnemius, and biceps femoris move the knee joint [2]. Many of the muscles contribute to motion of the knee joint along more than just one degree of freedom, some facilitate motion in the hip or ankle joints as well as the knee joint, and a few contribute to opposite movements along the same degree of freedom depending on the position of the knee. The knee is

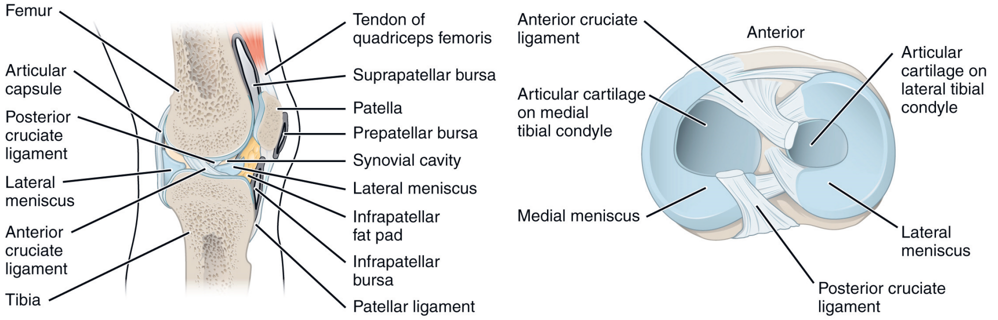

Figure 2.2: Sagittal and horizontal superior views of the knee joint. Image taken from `https://openstax.org/books/anatomy-and-physiology/pages/9-6-anatomy-of-selected-synovial-joints` under the CC BY 4.0 License. Edited.

mainly flexed by the semimembranosus, semitendinosus, biceps femoris, sartorius, gracilis, popliteus, and gastrocnemius muscles, and extended by the quadriceps femoris muscle group [2]. Muscles may also provide internal or external rotation and stabilize the joint, the magnitude sometimes depending on the flexion of the joint [2].

Although a small amount of abduction-adduction and internal-external rotation is possible depending on how flexed the knee is, the knee is constrained to mostly function as a hinge joint in the flexion-extension direction [2], although it is not a single-axis joint [49]. During common movements such as walking, great stresses are exerted on the articulating surfaces of the knee joint. These stresses are generated not only by the forces from human body mass, but also by soft tissues like muscles and ligaments that counteract the torques enabled by the moment arms of external forces [50–52], which results in additional compression in the articular surfaces of the joint.

2.2 PATHOLOGIES AND KNEE JOINT LOADING

Economically the most important knee pathology is knee osteoarthritis. It can be onset by damage to the articular cartilage by injury, but often develops over time even without known injuries [53]. No unambiguous evidence is available on all the suspected pathways of developing knee osteoarthritis. Possible causes include hereditary factors [10,53,54], metabolic factors [53], obesity [7,10,54], joint malalignment [54–56], and physical activity [10,54,57]. It has been shown that in addition to age [10], gender is a factor, as KOA is more likely to develop in women [10,53].

Structurally KOA manifests as loss of articular cartilage and can also cause, e.g., loosening of ligaments [10], weakening of muscles [10], inflammation of the synovium of the joint [10,53], and formation of osteophytes [53,55]. The changes manifest as pain [53,55] and stiffness [10] in the joint.

There is no cure and treatment is reactive, only trying to minimize the effects of KOA and slowing its progress [53, 58, 59]. Therefore, facilitation of proactive methods, e.g., by predicting the onset and progression of the disease is of paramount importance so that preventive steps can be taken.

The assessment of knee osteoarthritis can be approached from two perspectives: clinical and structural [7, 57]. Clinical knee osteoarthritis considers the disease from the perspective of clinical characteristics, e.g., radiographic observations, patient demographics, how much pain is experienced by the patient, and how much the disease limits the patient's quality of life [10, 60]. Structural knee osteoarthritis focuses on the physical degradation of the articular cartilage, often evaluated utilizing magnetic resonance imaging [57, 61, 62]. These two perspectives do not always go hand in hand, as patients with MRI-identified characteristics of structural KOA are not guaranteed to fulfil the clinical criteria of KOA diagnosis [10, 57, 62]. Therefore, clinical and structural KOA are not necessarily driven by the same factors.

Biomechanics is believed to affect the development of KOA [10]. Because biomechanical factors are modifiable, intervention involving them could be a way to prevent KOA [53]. Although the exact effect of biomechanics on KOA has been debated [58], physical loading of the knee joint is believed to be linked to the progression of structural KOA [7,9,10]. Therefore, ways to estimate knee joint loading could help in preventing the onset and progress of the disease. Furthermore, knowledge of the personalized loading of the knee could help in designing joint implants that retain proper joint function and experience less wear [7, 14], which in turn would help in managing the aftereffects of surgical KOA intervention.

Biomechanical estimation of the physical loading of the knee joint can be conducted through the analysis of human motion, particularly gait [6], which is important to study because it is the primary method of human locomotion and exerts the knee joint. Knee adduction moment is a commonly investigated surrogate measure for the distribution of knee joint loading between the medial and lateral compartments [4, 11, 63]. It increases in magnitude as the disease progresses [64] and is believed to contribute to the structural progression the disease [7, 8, 11]. This increase has been hypothesized to lead to gait changes that further promote the progression of KOA [12].

Instead of estimating knee adduction moment, compressive contact forces in the tibiofemoral knee joint can also be estimated, allowing more direct insights into knee joint loading [3–5]. These knee joint contact forces (KJCFs) describe how much force is exerted in the joint as a result of external forces (e.g., ground reaction force resulting from the leg supporting body weight) and internal forces (e.g., muscle contraction and ligaments) [3, 4, 6]. The latter forces are particularly important, as muscles forces contribute significantly to the physical loading in the joint [4]. If abnormal magnitudes and patterns of KJCFs damage articular cartilage, then interventions aiming to modify the KJCFs could be useful in preventing and managing structural KOA [5,7].

Knee joint contact forces cannot be noninvasively measured directly, but can instead be estimated using motion analysis and musculoskeletal modeling and simulation [3–6, 65]. We introduce and expand upon these concepts in chapter 3.

3 Motion analysis

Before computational analysis, experimental data must be collected using a variety of measurement technologies. These technologies can include X-ray imaging, marker-based motion capture, force plates, wearable motion tracking devices (particularly inertial measurement units), or standard video capture. Few of these methods are self-sufficient and some are used to complement one another. Although we present all commonly used technologies, we focus on marker-based motion capture, wearable motion devices, and video-based methods because they were utilized the most in this doctoral work, with chapter 4 further expanding upon video-based methods.

Computational analysis of human motion involves solving physical measures that drive the motion, including kinematics and kinetics, using experimental data. We present the main events of the human gait cycle, briefly describe how experimental data is pre-processed, especially from the perspective of digital filtering of signals, and describe how musculoskeletal simulations are conducted in biomechanical studies of human motion. Finally, we will look into some challenges involved in using musculoskeletal modeling and simulation to estimate physical loading in the knee joint and how those challenges could be overcome.

3.1 MOTION TRACKING TECHNOLOGIES

3.1.1 X-ray imaging of motion

Accurate information about the position of the skeletal system can be acquired via X-ray imaging modalities [49, 66]. The absorption of X-rays travelling through a medium depends on the wavelengths of the X-rays and the densities of the materials in the medium [49]. Commonly, dense materials like bone cause more absorption than sparse materials like soft tissue, which allows the use of X-rays to see bone tissue through soft tissue [49]. Therefore, X-ray imaging can provide direct information about the position and orientation of bones, unlike other motion tracking methods which seek to estimate the position and orientation of bones by following the motion of the surrounding soft tissue. An exception is the surgical insertion of bone pins into the bone so that the bones can be tracked using reflective markers in the extracutaneous part of the inserted device [67]; in addition to motion capture, this method still typically utilizes X-ray imaging to associate marker positions to bony structures [68].

X-ray imaging modalities use one or several image projections, with 3D reconstructions requiring at least two [49]. The motion of bones can be tracked with implanted radio-opaque materials or by tracing bony landmarks or contours [49, 66]. Imaging sequences are adjusted to the use case to find a middle ground between radiation exposure and image quality [66]. Furthermore, the images can be and often are used in combination with other motion tracking or tissue morphology information and technologies [66, 69, 70]. For instance, 3D models reconstructed from static MRI or CT scans can be used to estimate the position and orientation of bones by fitting the 2D projections of the 3D model to low-exposure X-ray images taken during dynamic movement [52,66,69,70].

Dynamic motion tracking methods utilizing X-ray imaging involve one or two radiation sources that are placed to image movements that occur within a limited volume [66] or even move with the subject [70] (Figure 3.1). These methods are not widely available, not only because they constrain allowed movements, but also because of the expensive equipment and the health risks of exposing live subjects to ionizing radiation.

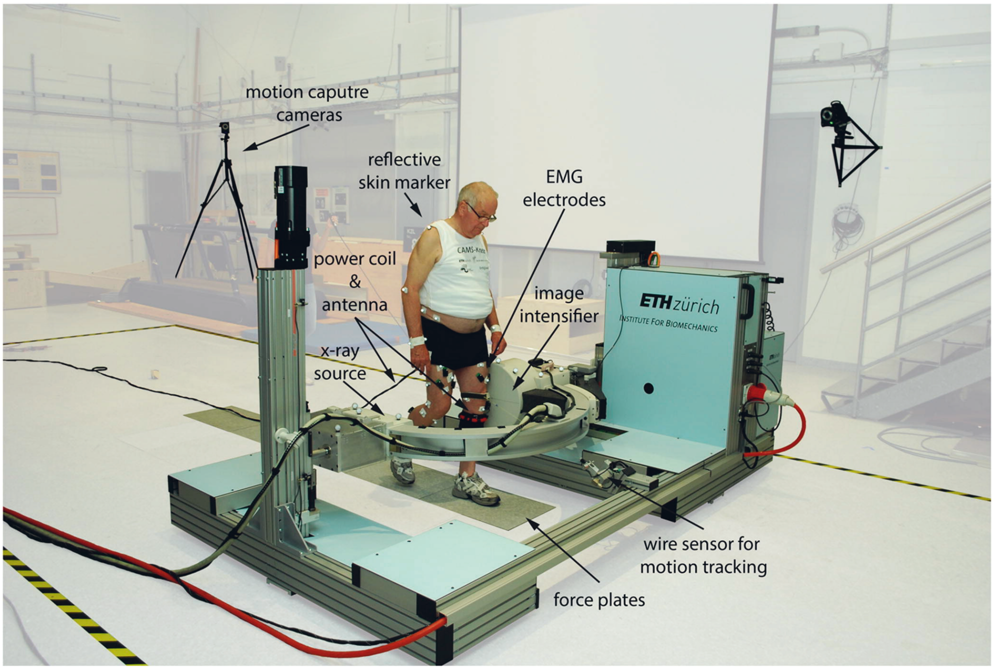

Figure 3.1: An X-ray fluoroscopy device mounted on a trolley that follows along the movement of the subject. Image taken unedited from "A comprehensive assessment of the musculoskeletal system: The CAMS-Knee data set" by William R. Taylor et al. (2017) [70] under the CC BY-NC-ND 4.0 license (`https://creativecommons.org/licenses/by-nc-nd/4.0/`).

3.1.2 Marker-based motion capture

Marker-based motion capture is the usual method of modern human motion tracking. It involves the use of reflective spheres, known as markers, that are placed on the subject's skin [67,71–77]. During the motion capture measurement, cameras detect the electromagnetic radiation reflected from the markers or emitted by them. Reflective markers are called passive markers, while emitting markers are called active markers [49]. From the positions of the detected markers, the subject's pose can then be reconstructed in each image frame [49].

Three-dimensional reconstruction of the movement of the subject requires the use of several cameras at different angles. Individual cameras only see the 2D trajectories of markers from their perspective, so methods to combine these 2D trajectories into 3D trajectories (e.g., the direct linear transform method) must be used [49,67,78,79]. Parameters such as the positions and orientations of the cameras must be calibrated prior to measurement [76,79], and therefore, the cameras must not move during measurements [49]. A minimum of two cameras is required for 3D analysis, but adding more cameras helps in ensuring that each marker will be seen by at least two cameras at any given time point [79].

Marker placement is an important consideration to ensure high-quality data in marker-based motion capture [80]. Markers should be placed so that

1. there is minimal soft tissue between the marker location on the skin and the underlying bony segment whose motion we assume the marker to follow [67, 71,78,79,81]

2. markers are clearly visible to as many cameras as possible during all movements [78,79]

3. markers are not immediately adjacent to one another [79].

Point (1) is important as marker motion is assumed to represent the motion of underlying bony segments, which are assumed to be rigid bodies [71,78,82,83]. The more soft tissue there is, the more "wiggle room" the marker has with respect to the underlying bone. In addition to soft tissue, direct motion of the skin should be considered because loose skin may be present even in areas of little soft tissue. Additionally, soft tissue surrounding the bones may cause errors in scaling musculoskeletal models and thus, in the final results of biomechanical studies [80,81,84,85].

Failure to follow point (2) will result in occluded markers. The results are single frames or several consecutive frames where a marker trajectory disappear from the data [78,79]. Small gaps in marker trajectories can be filled with spline fitting, assuming the missing marker to follow the trajectory of other markers on the same limb, and other methods. However, marker occlusion can cut a continuous marker trajectory into several trajectories that the analysis software cannot identify as the same, which requires additional work in the data pre-processing stage. Filling the gaps left by marker occlusion are presented in detail in section 3.2. Finally, in addition to improving the detection rate of markers, increasing the number of cameras reduces noise in marker trajectories [79].

Disregard of point (3) may result in two adjacent markers being identified as a single marker (Figure 3.2). If a segment has three markers and only two are detected by the system, we are unable to track all six degrees of freedom for that segment [79]. Therefore, it can be beneficial to have more than three markers per segment – the more markers there are, the better the segment can be tracked [86], assuming the markers can be separated by the camera system. Marker separation is usually simple to achieve, but can be problematic with small segments or segments where viable marker locations are very limited (e.g., the foot). In this respect, placing markers along the same line-of-view from any camera is ill-advised because they may be detected as a single marker by that camera even if the markers are not adjacent.

Provided the above three points can be followed, marker-based motion capture is very accurate [87], although measured kinematics always suffer from some soft tissue error [67,71]. For instance, Merriaux et al. (2017) reported that an 8-camera Vicon motion capture system is capable of mean marker positioning accuracy below 0.5 mm during dynamic motion [87]. Marker occlusion by the subject's own body

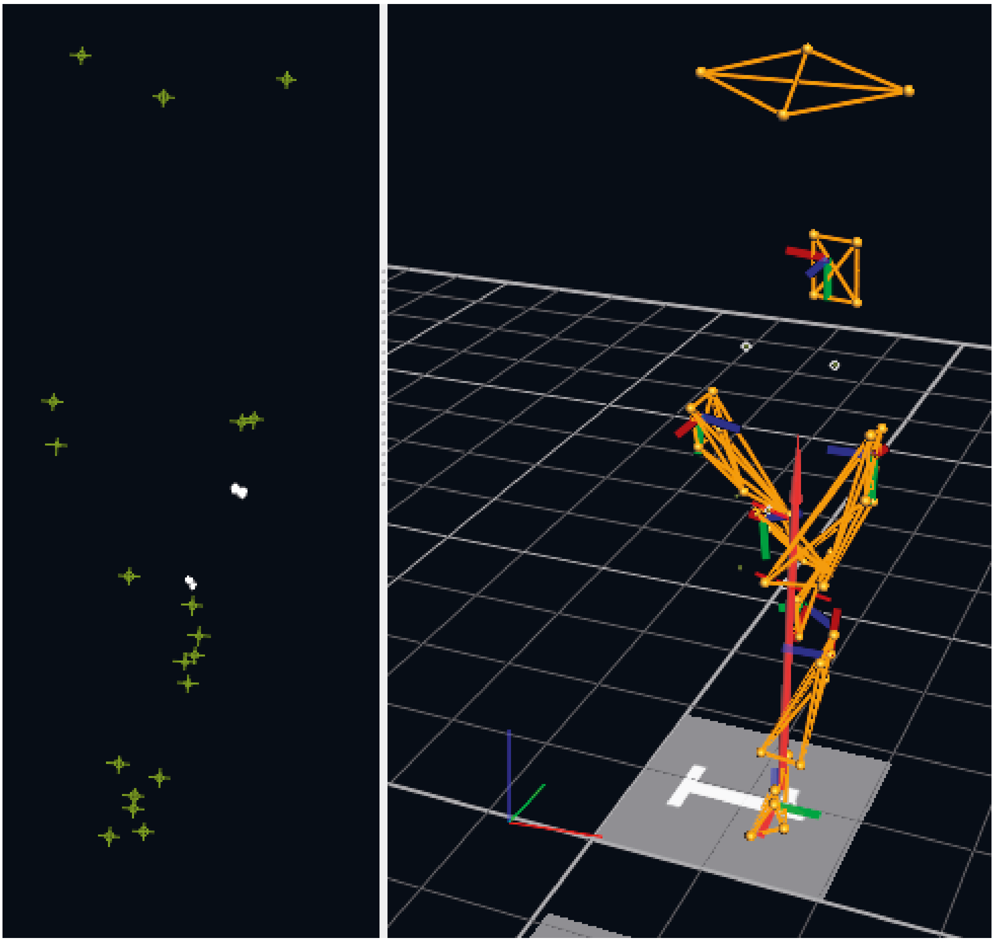

Figure 3.2: View of a motion capture frame from the Vicon Nexus software. The left side shows markers worn by the subject from the perspective of a single camera, while the right side shows a 3D reconstruction of the markers from the perspective of the same single camera. The left side shows markers which cannot be identified as separate markers because of their proximity in the image in white, while properly identified markers are in green. The image demonstrates how multiple camera views are required to properly identify markers and form the reconstruction on the right.

or objects in the measurement volume can seldom be completely avoided, but the placement of cameras can be planned to alleviate the issue [78].

3.1.3 Force plates

When you walk, you exert a force on the surface your foot is in contact with [50,51, 79]. In accordance with Newton's third law of motion, the contact surface exerts an equal but opposite force on your foot [79]. This force is called the ground reaction force (GRF) and it is of paramount importance in motion analysis [79].

Force plates are measurement devices that measure the ground reaction force using piezoelectric sensors, strain gauges, or similar technology [50, 79, 88]. They are typically rectangular plates that can be embedded in the floor [50,76,79] or built into instrumented treadmills [89]. Modern force plates track the center of pressure (COP), i.e., the point of application of the GRF, and ground reaction moments, i.e., moments that are exerted around the COP, in addition to the GRF [50, 79]. This information enables an in-depth three-dimensional analysis of the GRF. When coupled with, e.g., marker trajectories from marker-based motion capture systems, it also enables musculoskeletal modeling to calculate forces and moments affecting the human body [90].

3.1.4 Wearable motion tracking devices

Traditionally wearable devices for human motion tracking include electrogoniometers and accelerometers, among others [49]. Electrogoniometers are placed over the joint of interest and experience a change in voltage as their components move with the body segments around the joint, thereby tracking the relative change in joint orientation. Accelerometers measure the acceleration that forces affectuating their motion cause.

Modern wearable devices incorporate several different sensors in a single device. The most common such device is the inertial measurement unit (IMU), also sometimes called magnetic inertial measurement unit which emphasizes the presence of a magnetometer sensor [91]. IMUs usually contain triaxial accelerometers, gyroscopes, and magnetometers, and may contain other auxiliary sensors such as thermometers or barometers [11, 92–94]. In IMUs worn by people, these individual components are very small and packed within the frame of the IMU, which is usually small enough to fit on a palm (Figure 3.4).

Many modern IMUs communicate with computers wirelessly. They have rechargeable batteries and onboard microprocessors that use sensor fusion algorithms (e.g., the Kalman filter [31, 95, 96]) to estimate the desired output, which is usually the orientation of the IMU, before it is transmitted to a computer [92, 96]. There, orientation data from several IMUs are combined and may be saved for later analysis or in some cases, analyzed in real-time [94]. Depending on the IMU model, raw data (i.e., accelerations, angular velocities, and magnetic fields) may also be retrievable for analysis.

During measurement, IMUs are usually attached to body segments (Figure 3.3) [11, 96]. The orientation data is assumed to match the motion of the underlying segments, such as limbs, which are modeled as rigid bodies [71]. However, as devices worn on the skin, IMUs are subject to soft tissue artifact [74, 93, 97].

The advantage of wearable motion tracking devices, particularly IMUs, is that they can be used outside the motion laboratory [74, 93, 94]. Being constrained to a motion laboratory may cause bias in studies, as many studies measure their data in laboratory conditions and generalize results to daily life, even though a laboratory setting may change how the subject moves [5,98]. Furthermore, IMUs do not necessitate expensive motion capture equipment [91,93,94] and their calibration methods can be simpler than those of laboratory-grade motion capture equipment [74, 99], even if they involve the subject. Although full kinetic analysis with IMUs requires force plate data which is normally only available in motion laboratories, IMU data has been used to estimate it [17]. Therefore, IMUs can be used to measure motion

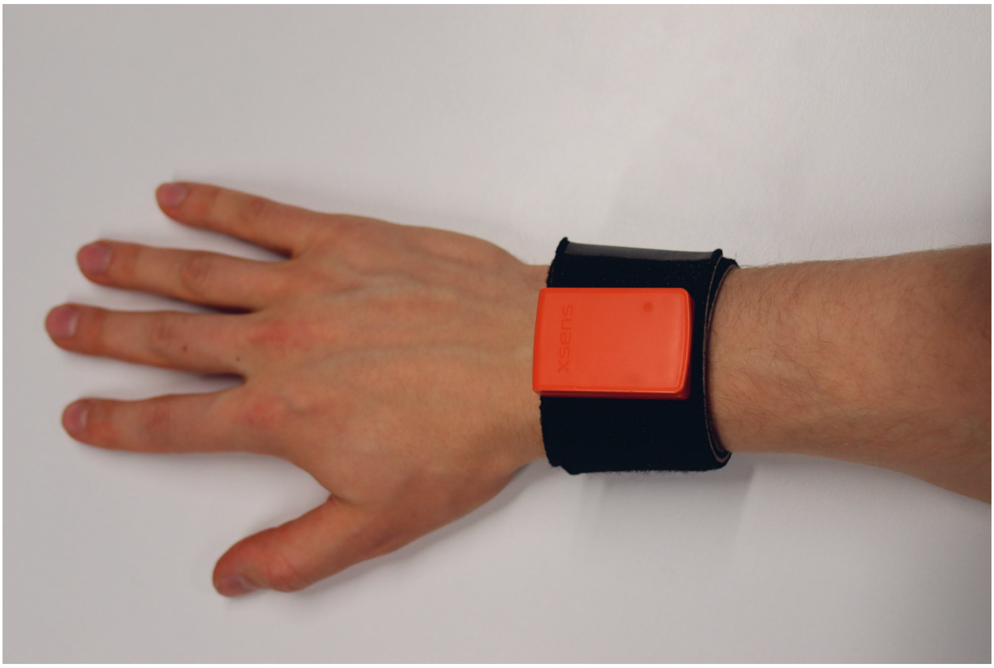

Figure 3.3: IMUs are usually strapped to body segments and their orientation is assumed to equal the orientation of the underlying bony structure. In this case, the IMU is assumed to follow the orientation of the radius.

in the real context of that motion, e.g., outdoors on a skiing lap to measure skiing technique.

The downside of IMUs is that they are slightly less accurate than laboratory-grade motion capture [74,93]. Some sensors, such as the magnetometer, may also be affected by ferromagnetic materials in the environment, thus increasing noise in the sensor readings [92]. As a result, some IMU-based solutions ignore the magnetometers data altogether [94, 100]. Furthermore, IMUs are sometimes characterized by a phenomenon called drift, where the heading orientation of the IMU accumulates error over time [93, 96]. This phenomenon can make long-lasting measurements prone to accumulating error [96,100], especially in cases where magnetometer data is not present [92, 94]. Finally, IMU data can be sensitive to how sensor calibration is done [94,101].

IMUs have been studied in human motion analysis with movements including walking [11, 32, 74, 93, 100, 102], squatting [31], jumping [31, 32], and running [11, 32]. These studies often involve comparing the retrieved parameters of motion to reference values obtained from a laboratory-grade motion capture system [31,32,74, 93, 100].

The differences in kinematics between IMUs and laboratory-grade motion capture systems are small and outshadowed by errors caused by other factors like sensor misalignment or skin movement [93]. A review by Poitras et al. (2019) noted that IMUs are a suitable alternative to motion capture systems particularly when measuring flexion-extension angles in the lower limbs during simple movements [91].

They noted that when measuring joint angles on the transverse plane, IMUs had lower reliability and validity than on the frontal and sagittal planes (commonly abduction-adduction and flexion-extension joint angles, respectively). Al Borno et al. (2022) observed that different joint angles are affected by magnetic disturbances and drift differently [103]. They noted that in the lower limbs, knee flexion angle is particularly resistant to these errors. Hip rotation angle (which occurs mostly on the transverse plane) experienced the greatest errors, possibly because of the proximity of foot IMUs (whose heading angles were in this case used to calculate hip rotation angles) to ferromagnetic materials on the ground. Because of their ease of use and similar accuracy to laboratory-grade motion capture systems, IMUs have often been proposed as a suitable technology for measuring human motion, particularly walking, in clinical settings or patients' homes [11, 31, 74, 93, 102].

Finally, smartphones have also been used as wearable motion tracking devices (Figure 3.4). They usually have integrated accelerometers, gyroscopes, and magnetometers as well as other components like GPS trackers that can be used in motion tracking. While these components in smartphones have not been designed for biomechanical analysis of human motion, most people carry a smartphone with them nearly all the time and sometimes the smartphones can be fitted snugly to pockets in normal clothing, which makes them convenient for everyday motion tracking.

3.1.5 Video analysis and markerless motion capture

Historically, motion analysis could be conducted through manual labeling of video films [73, 79]. More recently, increase in computational power has enabled the use of several digital video cameras for markerless motion capture. These markerless motion capture systems work similarly to marker-based motion capture systems, but instead of recording the trajectories of physical markers on the subject, they use manual labeling [79] or, more commonly, computational algorithms to track the trajectories of virtual markers, i.e., keypoints, on the subject [73, 104]. Although "classical" implementations use image analysis techniques, many modern algorithms utilize convolutional neural networks [73, 104], which are described in section 4.6.

Markerless motion capture systems are subject to many of the same limitations as marker-based motion capture systems are regarding marker occlusion [73] and cumbersomeness of setting up and calibrating multiple cameras. However, subject preparation is shorter and virtual markers cannot fall off during the measurement. While typical soft tissue error is not present because there are no physical markers on the skin, movement of clothing and soft tissue can still affect the detected virtual markers. Furthermore, although markerless motion capture systems can be easier to move than marker-based motion capture systems, they are generally designed for multi-camera laboratory conditions and may be proprietary, limiting their usability because it is impossible or costly to access and utilize their internal working principles. Nonetheless, they have no operator-specific bias (unless the videos are manually labelled) and may be suitable for cost-effective and accurate motion analysis in clinical settings where non-invasive methods involving minimal subject preparation are encouraged [73, 104, 105].

Figure 3.4: Wearable devices that can be used in motion capture include inertial measurement units and even smartphones. Inertial measurement units are particularly useful as portable modalities to measure motion due to their small size. The three devices on the left of the smartphone are inertial measurement units, with the bottom-most also including electromyographic sensors.

3.2 PRE-PROCESSING OF MOTION DATA

Force plate data and marker trajectories contain measurement noise [50, 106]. In marker trajectories, the noise is generated by temperature changes in cameras, the partial occlusion of the markers to some cameras, artifacts from reflective material that is not part of the markers, and soft tissue error [71, 78, 81]. Additionally, noise can be induced by erroneous labeling of markers (e.g., when automated labeling of markers switches places of markers on the same segment or between different segments, or a marker cannot be detected in some frames; see Figure 3.5). In force plate data, the noise is generated by vibration in the material of the force plate [79]. Noise should be removed from raw data before further analysis to keep it from accumulating error in the results of the study [106]. Noise removal is especially important because many motion analysis studies involve inverse dynamic simulations, where smoothness of input data is emphasized and sudden changes (e.g., by noise) can cause strong artifacts in the output of the simulation [20]. Smoothed data is especially important when derivatives of the data are calculated, because otherwise the derivatives will have sudden noise-induced changes [79,106].

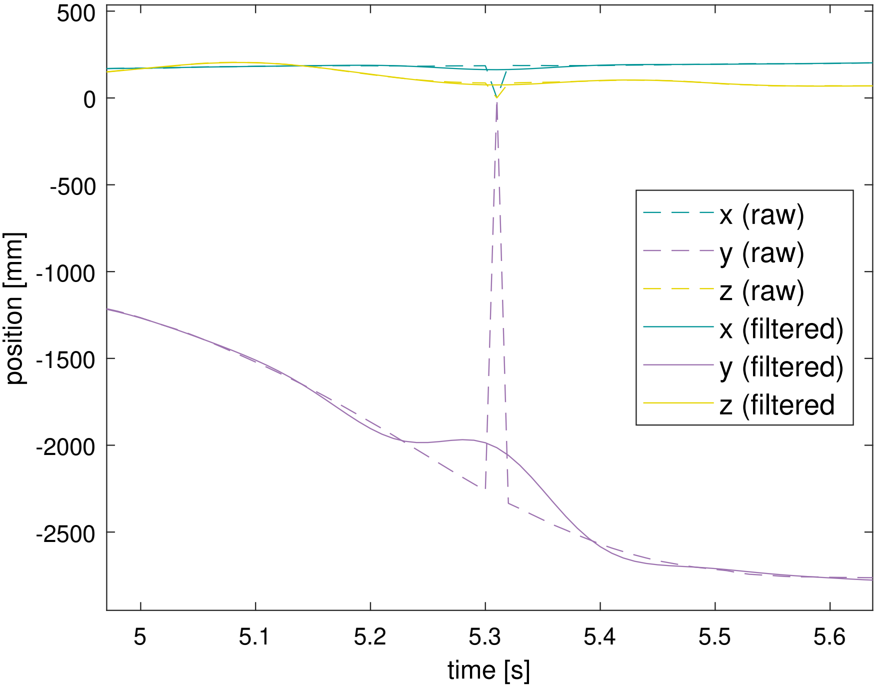

Figure 3.5: Trajectory of an ankle marker sampled at 100 Hz before and after 6 Hz low-pass filtering. Filtering can mostly eliminate single frames of erroneous marker data; however, a visible filtering artifact remains even after low-pass filtering.

Digital filtering is a common way to reduce assumed random noise from motion capture data [79]. Particularly low-pass filters can be utilized to reduce noise, which often has higher frequency content than the information used in motion analysis [79]. Butterworth zero-phase low-pass filters are often used in biomechanics analyses because of their good performance and robustness compared to alternative filtering methods [106,107]. Commonly used cutoff frequencies are at least 6 Hz for force plate data [3,4,11,20,32,63,76,89,107–109] and between 3–10 Hz for trajectory data [4,8,50–52,63,74,76,77,79,99,104,106–109], although the cutoffs can also be higher [11,75,107]. The commonly used cutoff frequencies for force plate data have a greater range than those for marker data, which is often filtered with a smaller cutoff frequency. Figures 3.5 and 3.6 demonstrate the effect of filtering on noisy force plate data and marker trajectories, respectively. Low-pass filtering marker and force plate data with a matching cutoff frequency may be useful, because frequency mismatch may between kinematic (marker) and kinetic (force plate) data can create artifacts in the data during musculoskeletal modeling and simulation [107]. This artifact can lead to, e.g., abnormally high knee contact forces [107].

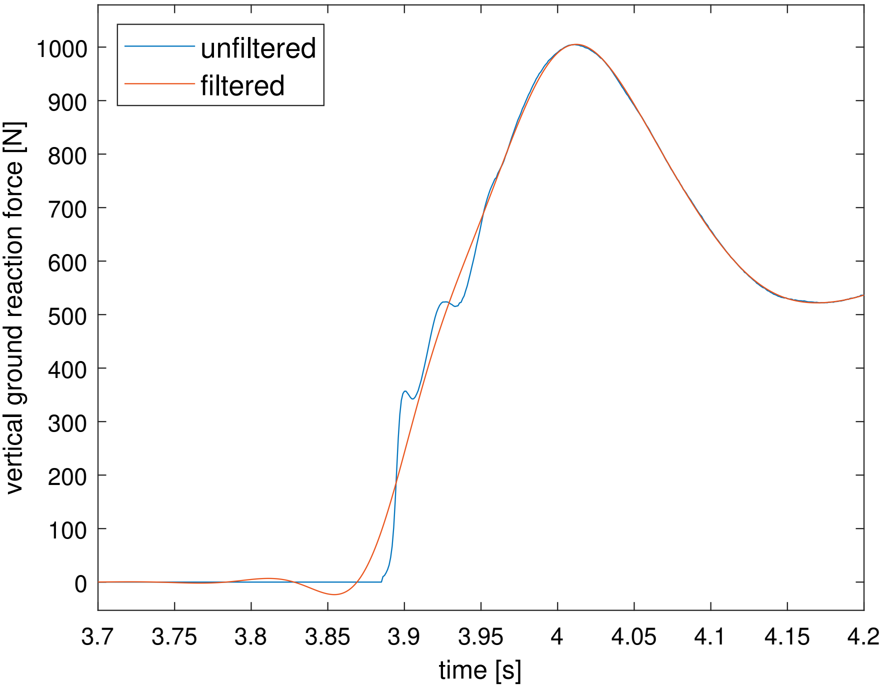

Figure 3.6: Vertical ground reaction force during transition to stance measured by a force plate at 1000 Hz before and after 12 Hz low-pass filtering. Low-pass filtering eliminates the higher-frequency oscillation present in the raw signal. However, a filtering artifact indicating negative force remains just before the actual foot contact begins.

Filtering does not remove only the irrelevant noise from signals, but often also removes relevant information [79]. When low-pass filtering at a given cutoff frequency, the frequencies above the cutoff frequency are assumed to contain no information, only noise. In real cases, usable information is seldom present only in an easily definable frequency range, but the frequencies of noise and information overlap. Therefore, some information is lost during filtering. Furthermore, the power spectrum of the signal does not immediately drop to zero at the cutoff frequency, but will do so gradually depending on parameters such as the order of the filter. The designer of the filter must use their expertise to define a cutoff frequency that results in a signal that retains enough information for the use case while sufficiently removing noise [79]. In some cases, valid assumptions of the data can be used to model the signal and remove noise efficiently using modified filters such as the Kalman filter [71,95].

In addition to digital filtering, gap filling is an effective pre-processing step before the motion capture data is used. It involves interpolating the trajectory of the marker trajectories to fill frames where the marker has not been identified [79]. Interpolation methods include fitting a spline function to frames where the trajectory is known, estimating the the trajectory from the motion pattern of the known trajectory, or assuming the marker and a group of other markers are fixed rigidly on a body and then using known trajectories of the other markers to estimate the trajectory. This latter rigid body filling is often viable with motion capture data, because it is fairly common to place markers on rigid clusters worn on the subject [67,71,79].

3.3 THE HUMAN GAIT CYCLE

During human gait, motion occurs in three dimensions. Typically, the largest motion occurs in the sagittal plane [110]. The gait cycle begins when one foot first touches the ground. It comprises the following sequential events:

1. initial contact: the foot makes contact with the ground usually with the heel, which is why this event is often called heel strike or heel contact [49]
2. toe off of the opposite leg: the opposite leg lifts up and stops touching the ground, with toes being the last parts of the foot to touch the ground [110]
3. heel rise: after the opposite leg swings past the foot and the center of mass of the person moves past the foot, the heel is lifted up while the toes remain in contact with the ground [110]
4. initial contact (heel strike) of the opposite leg [110]
5. toe off [110]

For a single leg, the gait cycle can be separated into a stance phase and a swing phase [49,111]. During the stance phase, the foot is in contact with the ground, and during the swing phase, the leg moves in air. The stance phase usually takes 60% of the gait cycle, while the swing phase takes 40% of it; as a result, both feet are in stance phase simultaneously for about 10% of the gait cycle during an event known as double support [110].

Because the leg bears the greatest load during the stance phase, it is of greater interest than the swing phase in this doctoral work. The stance phase lasts from initial contact to toe off and thus, includes steps known as loading response and terminal extension [110]. During loading response, the physical stress exerted on the leg increases as body weight is supported on it (Figure 3.7). This increase may begin with the heel strike transient, a sudden peak in the vertical force exerted between the foot and the ground (i.e., ground reaction force) caused by the impact of the heel on the ground [110]. Whether heel strike transient is present or not, a rapid increase in vertical ground reaction force follows as more of the body weight is supported on the leg and the opposite leg lifts off ground during the toe off event. Between loading response and terminal extension, the opposite leg swings past the supporting leg and vertical ground reaction force typically drops momentarily. In terminal extension, the heel rise event shifts weight towards the front of the foot and the increase in the downward acceleration of body mass center during mid-stance is balanced out by a foot push that causes another rapid increase in vertical ground reaction force [110]. This second force peak drops again as the opposite leg makes contact with the ground and the support of body weight starts to be shifted to the opposite leg.

Note that the definitions of loading response and terminal extension presented here may cover larger parts of the stance phase than in some sources, where the stance phase is divided into more gait cycle phases and the terminal extension is called terminal stance [110]. Our choice of simplifying the stance phase will become evident later in this work in the context of knee joint loading maxima.

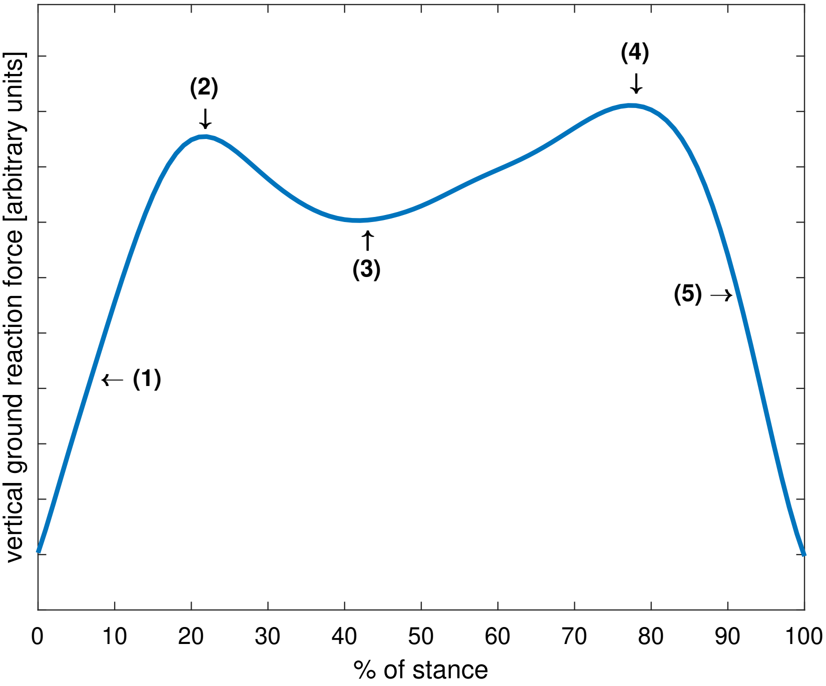

Figure 3.7: Vertical ground reaction force between ground and the supporting leg during stance. After heel strike, body weight is increasingly supported on the leg during loading response, and ground reaction force experiences rapid increase (1). The force culminates in the loading response peak when the opposite leg is lifted off ground (2). When the opposite leg swings past the supporting leg, the center of mass moves towards the toes and the force decreases momentarily (3). The terminal extension peak (4) occurs right before the heel strike of the opposite leg, after the ground reaction force drops (5) when weight is shifted to the opposite leg. The heel strike transient is not present in the image.

During each gait cycle, the knee flexion-extension angle has two flexion peaks and two extension peaks [110]. It is extended before initial contact, flexes during the loading response phase and the early part of mid-stance, then extends during late mid-stance. Finally, it starts flexing again and reaches a flexion peak during the initial swing of the leg before extending again before the initial contact of the following gait cycle.

3.4 MUSCULOSKELETAL MODELING AND SIMULATION

The reader has been introduced to different technologies for collecting experimental measurement data and how that data may have to be processed before using it in musculoskeletal simulation and modeling. We have also presented a simple model of the human gait cycle. Next, we will look into how the pre-processed experimental data can be used to calculate important biomechanical parameters, e.g., in studies of human gait.

We start by defining the terms that describe the physical state of a human in motion. Then, we present the various analysis steps that a musculoskeletal modeling and simulation study can comprise. Finally, we introduce a software that can be used to conduct the analyses.

3.4.1 Kinematics and kinetics of motion

Kinematics describes the motion of bodies, whereas kinetics describes the forces that are exerted on bodies [111]. Kinematics is often used to describe the motion of joints, i.e., joint angles, and translative motion of body segments [79, 90, 111]. For instance, knee flexion angle is a commonly investigated kinematic variable [42, 74, 77, 89, 112, 113].

On the other hand, kinetics is often used when discussing the moments that effectuate motion around joints, and the internal and external forces that body segments are exerted to [79, 90]. Internal forces are applied between segments within the human body, while external forces are exerted between the human body and its surrounding environment [78,79,90]. Ground reaction force between the foot and its contact surface is a commonly investigated external force [50, 51, 78]. However, external forces do not always require a contact surface, e.g., in the case of air resistance or the weight of the human body [78,79,90].

As a whole, kinematics and kinetics can be used to describe the physical state of a human in motion [79, 90]. In musculoskeletal modeling, this state is explicitly programmed as a combination of joint angles, moments and internal and external forces during a given point in time.

3.4.2 Modeling and simulation workflow

In order to retrieve in-depth biomechanical information (e.g., the kinematics and kinetics of individual segments in the human body) that is usable in real-life scenarios such as gait retraining [18,19,65,114], we need to run simulations on sufficiently realistic computational models of the human body [15]. These models are called musculoskeletal models because they incorporate information about the skeletal system, including segment mass, inertia, size, alignment, connecting joints, and relative positions; and the muscles, including the bones they attach to through tendons, the joints they actuate motion over, contraction dynamics, and the strength and cost of activating them [71, 72].

Available musculoskeletal models are usually generic in the sense that they represent the average properties for some population sample. These properties are obtained from studies on cadavers [65, 115]. However, the aim of musculoskeletal modeling is usually, at the very least as an intermediary step, to simulate the kinematics and kinetics of motion for individual subjects [72, 108]. Because people have differently built bones and musculature, population averages cannot accurately describe their biomechanical properties. Therefore, during biomechanical studies, these generic models must be modified so that they are specific to each subject. This process of modifying the properties of a generic musculoskeletal model to make it a subject-specific musculoskeletal model is called model scaling. Note that the model scaling described here is a common method to generate subject-specific musculoskeletal models, but musculoskeletal models can also be made subject-specific, e.g., by imaging the individual subject's skeletal system or muscle geometry or conducting experimental measurements of muscle force [116].

Model scaling of the skeletal system is done by matching the dimensions of the bony segments of the generic model to experimental measurements [15,72,108,115,117]. Usually, bone shape is assumed to remain unchanged, while segment mass is adjusted to match the mass of the individual subject, and the inertial properties are then calculated from the scaled properties [15,117]. Ideally, this scaling would be done individually for each bone, but in effect, the available experimental data usually requires grouping the bones by limbs or body segments and scaling all bones in that group similarly. Instead of using experimental data (typically utilizing marker trajectories from marker-based motion capture [15,72,115]), the generic models can also be scaled using manually defined scale factors. Recently, fitting a statistical shape model of the skeletal system rather than matching experimental markers from MOCAP with virtual markers on the model has been proposed as an alternative for conducting subject-specific scaling of generic musculoskeletal models and has been shown to be more resistant to marker placement-related errors [84]. Similarly to scaling of the skeletal system, the properties of the muscular system and other soft tissues can be scaled, e.g., by assuming that they change proportionally to the anthropometrics, height, or weight of the subject [4,15,108,115,118].

After model scaling, the next step is solving the kinematics of motion. This step requires experimental measurement data, which usually means marker trajectories. The basic assumption is that body segments are rigid bodies whose motion can be approximated by tracking markers on the skin [71,82]. If marker trajectories from at least three markers per body segment are available, it is possible to directly rotate and position the corresponding body in the model according to those three markers in three-dimensional space and solve kinematics from the resulting poses [90]. However, because of measurement error, this method can lead to unrealistic poses of the model. To alleviate this effect, musculoskeletal models have constraints on the motion of their joints (e.g., the knee cannot overextend far) [99]. Therefore, instead of directly aligning each body of the model according to measurement data, solving kinematics is about finding the set of joint angles that aligns and positions the bodies in a way that minimizes the error between the experimental data and the valid poses of the model (as defined by the constraints on joint angles) [72,99,114]. The error is usually defined as the error of the sum of squares of individual errors, which for marker data are differences in position between experimental markers and virtual markers in the model [72,114]. The process of calculating kinematics (i.e., joint angles) this way is called inverse kinematics. Finally, in cases where the experimental measurement data contains orientation data from wearable sensors instead of marker trajectories, the error can be defined as the difference in a parametrization of rotation between experimental data and model poses [99].

Using mathematical notation, inverse kinematics in a single frame of data can be solved by finding the set of generalized coordinates that minimizes the objective function

$$f(\vec{q}) = \sum_{i}^{N} w_i ||\vec{x}_i^* - \vec{x}_i(\vec{q})||^2 + \sum_{j}^{M} \omega_j ||q_j^* - q_j||^2, \tag{3.1}$$

where N is the number of markers (the same number for experimental markers placed on the subject and virtual markers defined in the musculoskeletal model), w_i

is the weight of the i–th marker, $\vec{x}_i^*$ is the position of the i–th experimental marker, $\vec{x}_i(\vec{q})$ is the position of the i–th virtual marker (and dependent on the set of generalized coordinates $\vec{q}$), M is the number of generalized coordinates to solve in the model, ω_j is the weight of the j–th generalized coordinate, q_j^* is the value of the j–th experimental generalized coordinates, and q_j is the value of the j–th generalized coordinate in the model [119].

Here, generalized coordinates describe the rotations of joints (i.e., joint angles) and sometimes the translational offsets of bodies (e.g., the position of the parent body of the model, usually pelvis, with respect to the reference coordinate system or ground). Experimental generalized coordinates may be, e.g., defined manually or calculated from IMU data. If only marker data is available (and no manual goals for the values of solvable generalized coordinates are set), the second sum term of the equation may be left out, and if only IMU data is available, the first sum term may be left out. Note that equation 3.1 describes one way, but not the only way, to solve inverse kinematics.

Once the kinematics of motion are solved, the dynamics of motion can be calculated. Following a similar principle as inverse kinematics, where data about the pose was known and the joint angles leading to that pose had to be solved, in inverse dynamics the kinematics are known and the problem is to find the net forces and torques that lead to those kinematics [4,79,83]. Experimentally measured external forces, namely the ground reaction forces, are incorporated into the computations to find the internal forces (i.e., joint moments) that together with the external forces create the motion [72,109]. Inverse dynamics is an iterative process that moves proximally from foot (or another segment subjected to external forces) to other segments of the human body and finally leaves a set of residual forces and moments that have no physiological meaning [15,65]; to counteract this, forward dynamics provides an alternative [108].

Next, muscle activation levels can be solved. Because the dynamics of motion (from experimental measurements and inverse dynamics) and the properties of muscles (defined in the musculoskeletal model) are known, it is possible to estimate how much force different muscles must generate to produce the known dynamics [4,72,114]. Because the association between muscle activation and the force it produces is part of the musculoskeletal model, the set of estimated muscle forces can be linked to muscle activations [16]. In other words, the optimization algorithm tries to find the set of activations of individual muscles that produce the dynamics of motion while minimizing the sum of those activations. Additionally, instead of solving a pure optimization problem without measured muscle activation data, EMG-informed methods exist to utilize electromyographically measured experimental muscle activation data for optimizing muscle activations in the musculoskeletal simulation [120].

For instance, muscle activations at each frame can be solved using a process known as static optimization by minimizing the objective function

$$f(\vec{a}) = |\sum_{m}^{n}(a_m)^p| + |\sum_{m}^{n}(a_m F_m^0) r_{m,j} - \tau_j|, \tag{3.2}$$

where a_m is the activation of muscle m, F_m^0 is the maximum isometric force of muscle m, $r_{m,j}$ is the moment arm of muscle m about joint axis j, τ_j is the generalized force about joint axis j (solved during inverse dynamics), and p is a constant [121]. Note that different implementations of solving muscle activations exist, and the equations may vary depending on how muscle is modeled [122].

Although static optimization is a computationally efficient and commonly used method for solving muscle activation levels [88], it has several known drawbacks. Anderson and Pandy have summarized that the validity of static optimization relies heavily on the accuracy of input kinematics for estimating the dynamics of the model (as explained in section 3.2), that the time-independence of static optimization can make it difficult to model muscle physiology properly (determining properties like force-length-velocity can be difficult when observing just a single time point at once), and that static optimization cannot incorporate time-dependent performance criteria which may be relevant for the motion task (e.g., utilizing distance moved or energy expenditure over a period of movement in the performance criterion) [88, 123]. Dynamic optimization methods have been proposed as a solution to some of these drawbacks. These dynamic methods include the computed muscle control algorithm, which combines the computation of desired accelerations in a future time step, static optimization, and forward dynamics [124]. Dynamic optimization allows solving muscle activation levels over a motion trajectory rather than just at individual time steps [116] and even enables predictive modeling [88]. Nonetheless, static optimization provides a computationally faster solution that results in dynamics of motion similar to dynamic optimization during walking [123].

Finally, when muscle activation levels and external dynamics are known, the joint reaction forces (e.g., tibiofemoral compressive knee joint contact forces) can be solved [4]. This process combines information about the external forces acting around different generalized coordinates and the internal forces generated by, e.g., muscle contractions.

The validity of musculoskeletal modeling and simulation can be evaluated by comparing the estimated kinetic outputs of the simulation to corresponding ground truth information. However, real joint loading information is difficult to obtain because there is no instrumentation in healthy joints to give researchers direct quantitative information about loading. Therefore, researchers have instead utilized loading data measured with instrumented joint implants [5, 6]. For instance, in the context of knee joint loading, knee replacement surgery patients have been implanted with instrumented knee implants that measure loading in the replaced joint [6,125]. Comparison between estimated and implant-measured loading allows the evaluation of not only the accuracy of the loading estimate, but indirectly also the validity of the underlying modeling assumptions [5].

The intermediary kinematics and kinetics are often low-pass filtered between the analysis steps to reduce simulation-related artifacts that could affect the results of the analyses that follow. Although all the steps of musculoskeletal modeling and simulation can be implemented manually, doing so would be cumbersome, and several software are available to provide the underlying optimization algorithms, standardized models and data protocols, and user-friendly interfaces. These software include OpenSim [15], which we will introduce as it was utilized extensively in studies **I**, **II**, and **III**.

3.4.3 OpenSim

OpenSim is a software system for musculoskeletal modeling and simulation that provides a graphical user interface (Figure 3.8) and application programming interfaces on various programming languages [15]. It implements model scaling, solving of kinematics and kinetics, and other musculoskeletal analyses as user-accessible and modular tools [15]. The user of a tool has to provide the input data and settings, all of which can be conveniently saved in an XML file [16]. This file allows simple human-readable access to modeling parameters and allows their effortless modification. Researchers can pick the musculoskeletal model that best suits their research question from the wealth of different models that the OpenSim community has created and validated; these models are likewise in XML-compatible format [16]. Furthermore, the data files (e.g., joint kinematics and kinetics) generated by OpenSim are in human-readable format and can be easily opened in MATLAB, Python or other programming languages if the user so desires. OpenSim also has an application programming interface (API) to several languages, including C++, Java, MATLAB, and Python [16]. Unlike many of its competitors, OpenSim is completely free and open-source [15].

OpenSim's ease of use, contribution, and collaboration in studies involving musculoskeletal modeling and simulation have elevated OpenSim from just another analysis software to a biomechanical research community [16]. Additional analysis software and plugins utilizing OpenSim or supporting OpenSim analyses include OpenSense (elaborated in the next paragraph), OpenSimRT (a tool for real-time kinematics and kinetics analysis from marker and IMU data) [29], OpenSenseRT (a wearable system and software real-time kinematics analysis from IMU data) [94], rtosim (a software for real-time kinematics analysis from marker data) [126], NMS-Builder (a software for creating subject-specific OpenSim models from segmented imaging data) [127], CEINMS (a plug-in for predicting neural control solutions) [120], and OpenSimLive (which was developed in study **I** and is explained in the methods chapter).

OpenSense is a software tool for kinematics analysis of IMU orientation data [103]. It extends the API of OpenSim and is included in the current version of OpenSim. It operates on same principles as OpenSim's marker-based inverse kinematics tool, i.e., by finding joint kinematics that minimize the differences between measured and modeled data. However, instead of marker trajectories as the data, OpenSense uses IMU orientations. OpenSense is included in current versions of OpenSim.

3.5 CHALLENGES IN ESTIMATING KNEE JOINT LOADING

Now that we have described the fundamentals of musculoskeletal simulation and its relevant tools, we have the means to approach the problem introduced in section 2.2: estimating the loading of the knee joint. Conventional estimation of knee joint loading can be separated into two major steps: **(1)** data collection and **(2)** musculoskeletal modeling and simulation.

Among the methods we presented for data collection in section 3.1, at least marker-based motion capture systems and force plates are required to collect the necessary data (i.e., marker trajectories for kinematics and force plate data for dynamics simulation) for estimating joint dynamics, although the data of some portable technologies (e.g., IMUs) can be used for approximating force plate data [17]. The requirement of a motion laboratory with expensive and cumbersome equipment sets an accessibility barrier for estimating the loading of the knee joint routinely for people who need those estimates the most (e.g., patients of physiotherapists and clinicians).

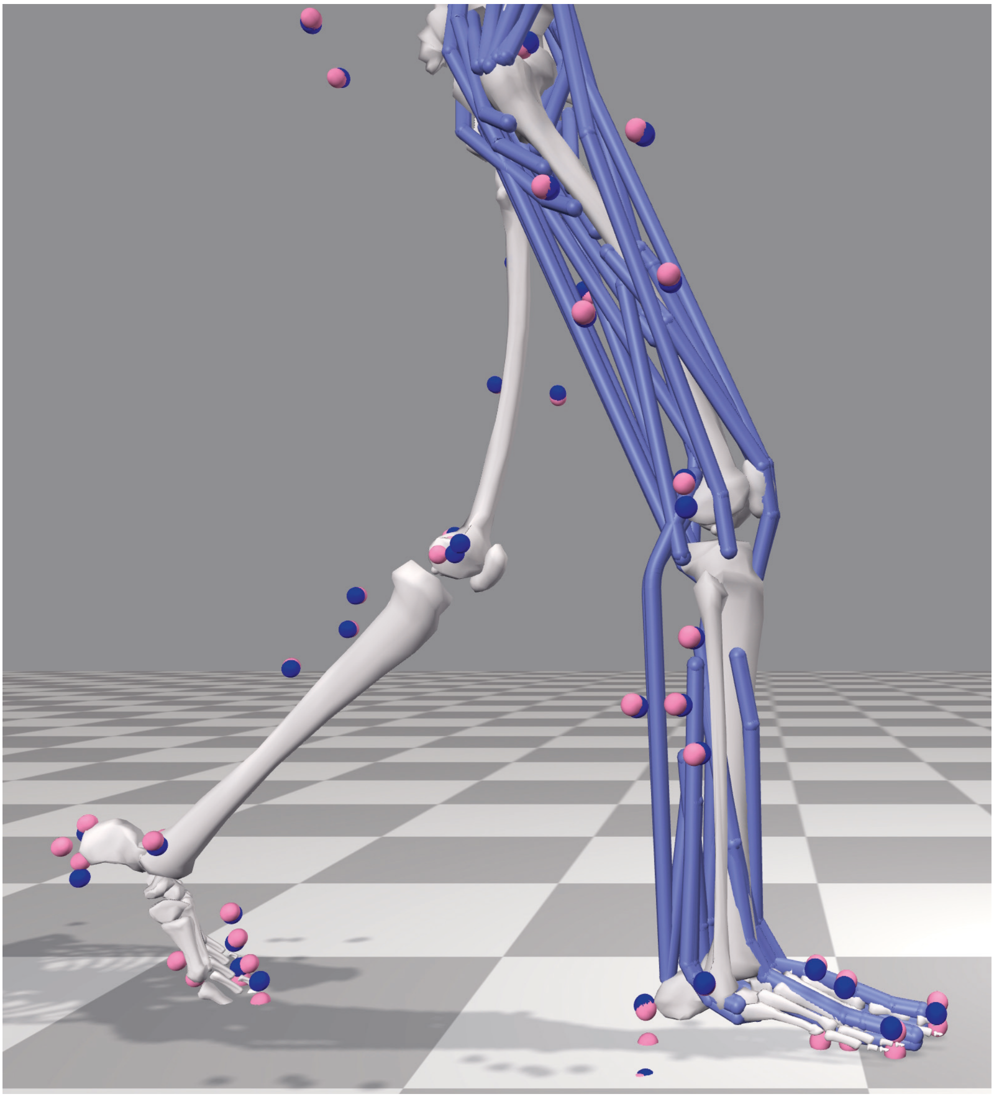

Figure 3.8: A snapshot from the model visualization of the OpenSim graphical user interface. The musculoskeletal model and its motion can be visualized in 3D. The illustrated model shows the skeletal structure of the lower limbs, with muscles on the right leg shown in blue. Virtual markers defined on the model are shown as magenta spheres and experimentally measured marker positions matched to them during inverse kinematics are shown as blue spheres.

Musculoskeletal modeling and simulation further raises this accessibility barrier, as physiotherapists and clinicians seldom have the expertise required to run these analyses [21]. Additionally, the computational steps involved are too cumbersome to enable real-time results during the measurement session (e.g., in gait retraining) [18–20]. These challenges are presented in more detail in section 4.6.

Therefore, to enable the simple and routine estimation of knee joint loading, the challenges related to these two steps must be overcome. In the past decade, AI-based solutions, namely machine learning algorithms, have seen increasing use to solve these challenges [128] [38]. Among machine learning algorithms, various types of artificial neural networks have been among the most used algorithms [38], e.g., to extract useful data from portable modalities [42,129,130], reduce the analysis time [20], and estimate joint dynamics [18,20,128].

4 Artificial neural networks

In this chapter, the reader is introduced to the basics of artificial neural networks, particularly fully connected feedforward networks and convolutional neural networks. We focus on artificial neural networks for supervised learning tasks. We explain what artificial neural networks are and how they work in very simple terms. This includes their basic structure (layers, weights, nodes, and activation functions) and how the two architectures (fully connected feedforward and convolutional) differ from one another. We also introduce the basic principles of training artificial neural networks. Lastly, we briefly review the use of artificial neural networks in human motion analysis and estimation of knee joint loading. Unlike the previous sections where we introduced basic theory, at that point we also present some real applications of artificial neural networks.

4.1 BASICS OF ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) are computational systems that can learn to solve problems. They are modeled after and derive their name from biological neural networks of the central nervous system, which consists of a series of connected neurons, or nerve cells [131–133]. Neurons receive information as signals from other cells through dendrites, combine and modify that information in the body of the neuron, and send it to other cells through axons [134]. The projections of neurons, i.e., dendrites and axons, connect through synapses and modify the neural signal as it travels between cells. In ANNs, these projections are called weights and the bodies of neurons are called nodes (or units or simply neurons, depending on source material). Similarly to biological neural networks, nodes are organized into layers with one or more nodes each. Thus, an ANN comprises layers, which in turn comprise nodes.

A typical layer structure contains an input layer, an output layer and one or more layers, known as hidden layers, between them (Figure 4.1) [38, 134]. The input layer has nodes corresponding to the number of inputs of the ANN, and the output layer has nodes corresponding to the number of output values of the ANN. The hidden layers may have any number of nodes each. When there is just one hidden layer, the ANN is called a shallow neural network. When there are several, it is called a deep neural network; the depth of a network is a property that describes the number of hidden layers it has [135].

ANNs are suitable for a wide variety of tasks including classification, regression, optimization, and clustering [132, 134, 136]. Although the reputation of ANNs as "black boxes" may make them daunting to approach [137], the fundamentals of ANNs actually rely on simple mathematics. In ANNs, numerical information from the input nodes is propagated through the network in a series of mathematical operations, most notably through linear combinations of the weights of the network and through nonlinear activation functions [132]. Weights enable the ANN to learn how to better handle its task, while activation functions enable the ANN to solve nonlinear problems. The latter are presented in more detail below in their own subsection.

The weights of the network are where its information is stored [134, 135]; they can be thought of as the main parameters or coefficients of the network [135]. They enable the ANN to learn how to better handle its task. Weights act as connections between nodes (in feedforward networks, they provide one-way connections). Between the nodes of adjacent layers, weights connect at least some of the nodes, although every node does not necessarily connect to all nodes in the following layer [134]. Whenever the output of a layer connects to a node in another layer, it undergoes a dot product with the weights assigned between the original layer's nodes and the target node [134]. This way, the original input is modified by the weights between each layer. Furthermore, each node typically has a bias that is added to the dot product before the result is inputted into the activation function of the node; this activation function is often nonlinear [134]. Modifying the biases and weights will change the output of the ANN and the ANN is said to learn. The process of modifying the weights to make the ANN suitable for its task is called training the ANN. Network training aims to find the optimal weights for solving a problem [134].

When we know what value or label the ANN should output for given input data and the weights are changed such that the difference between the output of the network and the known target output is minimized, the training process is called supervised learning [131, 132, 138]. Supervised learning aims to produce models that can then be used to make predictions about unseen or future data [138]. Otherwise, such as when we do not know the target values but can define some kind of objective function the output should minimize, the training process is called unsupervised learning. Several different training methods exist, e.g., modifying the weights according to the derivative function of the activation function for supervised learning, and generating random weights and seeing if the output improves for both unsupervised and supervised learning [131, 134, 136].

Several different architectures of ANNs exist and are suited for different tasks [75, 131, 134, 136]. For this thesis, the relevant architectures are fully connected feedforward neural networks and convolutional neural networks. The operating principles of these two architectures are presented in their respective subsections.

4.2 FEEDFORWARD NEURAL NETWORKS

Feedforward neural networks (FNNs) received their name from the fact that the nodes are connected acyclically. In other words, the connections between the nodes do not loop back and the input propagates only in one direction (Figure 4.1): from the input to the output through the layers [134, 135, 139]. In feedforward neural networks, the output of a node depends on the outputs of the neurons in previous layers, but not on the outputs of neurons in the following layers [131]. They are often used in regression and classification tasks; the difference is that in regression tasks the output is a continuous value while in classification it is discrete and typically an integer [138].

In this thesis we focus on FNNs that solve regression problems. The universal approximation theorem states that given the right parameters and structure, a multilayer FNN can approximate any real function arbitrarily well, meaning that FNNs can be used to solve regression problems where no analytical solution is known [136].

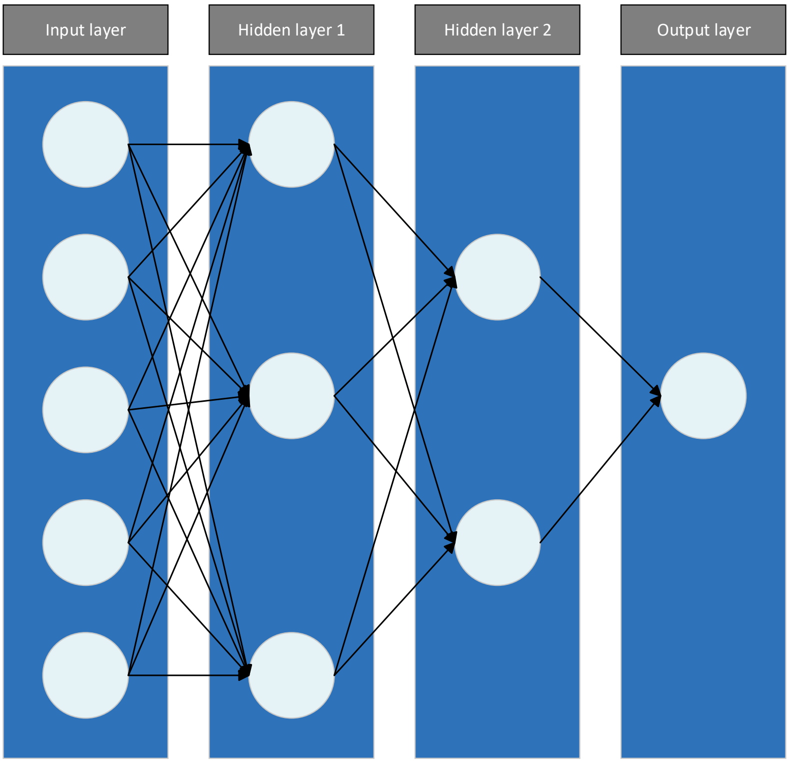

Figure 4.1: A layer-level illustration of an artificial neural network with an input layer, two hidden layers and an output layer. The nodes are connected with black arrows, each of which represents a weight between two nodes. The input layer has five nodes, the first hidden layer has three, the second hidden layer has two, and the output layer has one node. Therefore, the illustrated neural network would take five scalars as an input and output a single scalar. The illustration describes the fully connected feedforward network architecture.

4.3 CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) are feedforward networks designed for input data that is arranged in a grid-like pattern, such time series and images [133, 135, 138, 140]. In CNNs, the weights of the network are placed in arrays called kernels [133]. For one-dimensional data such as time series, the kernels are one-dimensional tensors (i.e., vectors), and for two-dimensional data such as images, the kernels are two-dimensional tensors (i.e., matrices) [135]. If the input data has several channels, such as RGB images, where the channels describe the intensity of the colors red, green, and blue for each pixel, then there is a kernel for each channel and the combination of the kernels in that layer is called a filter [133]. A filter with several kernels can be thought of as a three-dimensional tensor, although some sources treat filters as two-dimensional tensors and may even use the term "filter" and "kernel" interchangeably [138]. A layer may have several filters, i.e., filters with different sizes and weights. In this dissertation, we will focus on CNNs for image-format data.

When applied on images, filters are feature detectors that detect visual patterns (Figure 4.2) [134]. Typically, filters in early layers detect low-level features such simple geometrical shapes and layer by layer the filters gradually detect higher-level features [134, 138]. For instance, in a CNN trained to identify human poses, the filters in the first layers only detect edges and corners, but in the later layers they detect simple objects like eyes, noses, and limbs, and finally in the last layers they detect entire people and their activities.

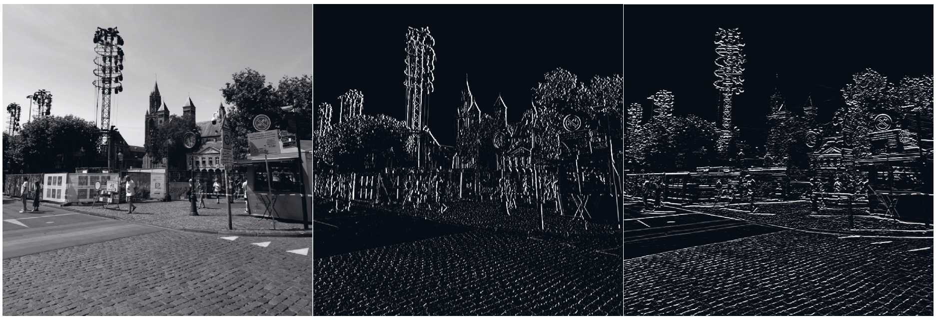

Figure 4.2: Filters in convolutional neural networks extract visual patterns from the image. In this example, we demonstrate how different visual patterns can be extracted from the input image (left) using a horizontal (center) and vertical (right) edge detection filters. Note how the different filters highlight different patterns in the image (e.g., ground texture and the church in the background). Here, the filters are 3x3 matrices. For demonstration purposes, the elements of the filters are integers from known edge detectors; in convolutional neural networks, the elements are weights of the network that are optimized during training, and more features are extracted sequentially from the previously extracted ones using other filters.

When a filter is applied in a convolution layer, the kernels of the filter sequentially slide over the image data and calculate the dot product between the weights in the kernel and the image data it overlaps [134]. The result of this convolution operation is another two-dimensional matrix, i.e., another image [134,138]; the use of the convolution operator gives CNNs their name [135]. Sometimes, zeros are appended to the edges of the input image before applying the filter because otherwise a kernel with a size greater than 1x1 will output an image that has a smaller resolution than the input image (Figure 4.4). This technique is called zero padding [134, 138, 140]. Furthermore, the kernel will not necessarily slide over the image pixel by pixel, but may move several pixels before the next convolution operation. The amount of pixels the kernel moves between convolution operations is called stride and is often an adjustable parameter in CNNs [134, 138, 140]. The output image has a lower resolution than the original image if the stride is greater than one (4.3).

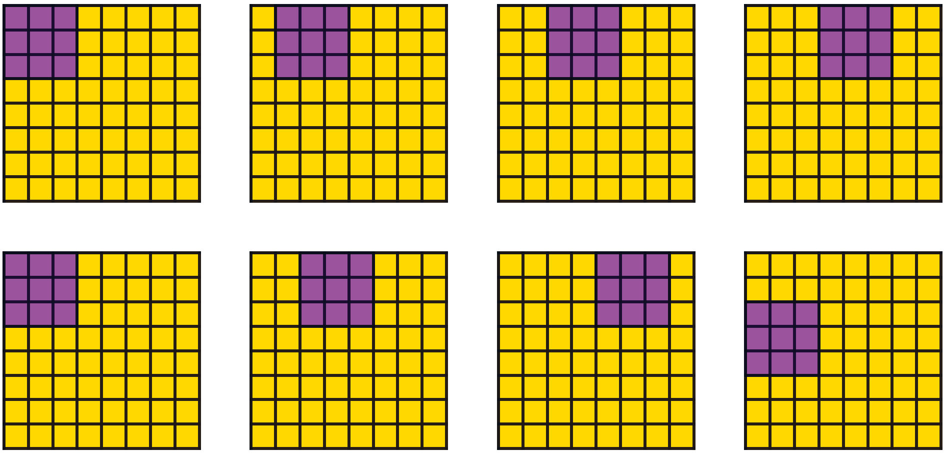

Figure 4.3: Illustration of a 3x3 kernel (dark blue) with two different stride parameters sliding over an 8x8 image. The first four positions of the kernel are shown from left to right. When stride is 1 (top row), the kernel moves one pixel between operations and produces a 6x6 image. When stride is 2 (bottom row), the kernel moves two pixels between operations and produces a 3x3 image. In the illustrated case, the image resolution is reduced even when stride is 1 because the kernel does not slide over the boundaries of the image.

A pooling operation is often applied on the image to downsample its data [135]. This is similar to the filter in the sense that it also sequentially picks values in the 2D matrix of the original image and produces an output based on them [134]. However, unlike the weights in the kernels of the filter, the pooling operator has no learnable parameters. If it has weights, they are not changed during training. A common pooling method is the max pooling method, which picks the maximum value from the elements it overlaps with [134]. The max pooling operator is an example of a pooling operator without weights.

4.3.1 Computational efficiency

Compared with "traditional" artificial neural networks, CNNs are computationally more scalable [134]. For a traditional ANN, all pixels in an image represent an input value that is connected to each node in the following layer by a unique weight. In CNNs, every pixel is not connected to the following layer by a unique weight [140]. Instead, the weights are placed in a number of filters such that each filter is small in comparison to the input image [134]. For instance, the filter size may be as small as 3x3 [33,141,142], meaning there are nine weights in each kernel of the filter. Several different filters are used to extract different features (for low-level features, these could be different kinds of edges in the image), but the number of filters is limited enough to keep the number of unique weights low. This sparse connectivity and reusability of the same weights (organized in kernels) is made possible because many features of interest in image data (e.g., edges, corners, shapes) are non-unique

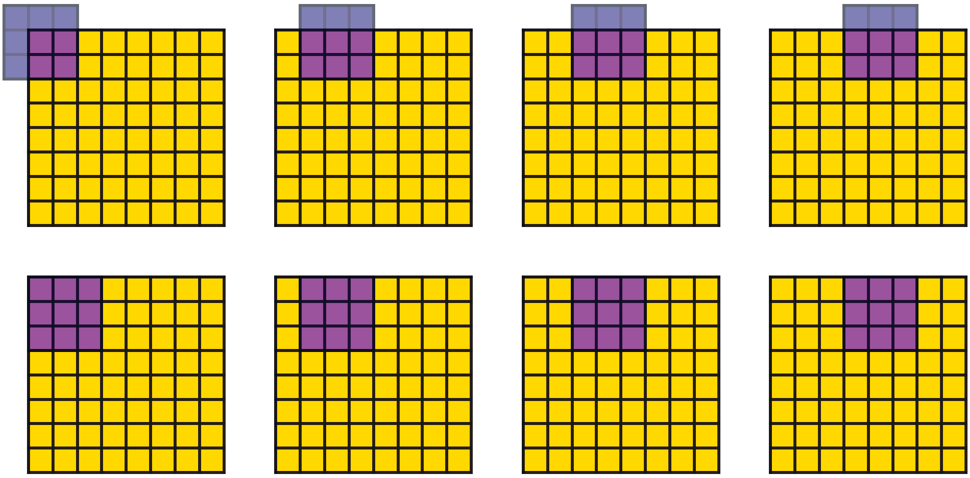

Figure 4.4: Illustration of a 3x3 kernel (dark blue) sliding over an 8x8 image. The first four positions of the kernel are shown from left to right. When zero padding is used (zeros not shown in image), the kernel is allowed to slide past the original boundaries of the image (top row) and produces an 8x8 image, i.e., the resolution remains unchanged. Without zero padding (bottom row), the kernel produces a 6x6 image.

and reoccur in different parts of the image [135]. Lastly, in addition to faster training because of the limited number of trainable weights, the limited number of parameters reduces the memory requirement of storing trained CNN models compared with fully connected ANN models [135].

For instance, a 64x64 image has 4096 pixels. Between the input layer and the following layer with, say, ten nodes, there are $4096 \times 10 = 40960$ weights to train in a traditional fully connected ANN where a unique weight connects each input to each output in the following layer. To reach the computational load (i.e., cumbersomeness, not equal performance) comparable to traditional ANNs with 40960 weights, there would have to be approximately 4500 3x3 single-kernel filters (which would equal $4500 \times 3 \times 3 = 40500$ trainable weights). Usually, CNNs reach pattern recognition capability comparable to the overcomplicated fully connected ANN with a fraction of that number. In other words, the number of trainable parameters in a CNN can be several orders of magnitude smaller than the number of trainable parameters in a fully connected ANN of comparable performance. Therefore, CNNs are computationally more efficient than traditional ANNs.

4.3.2 Layer structure

Like traditional artificial neural networks, convolutional neural networks have a biological analogy [134, 138]. The visual cortex of the brain is organized in layers. Each layer processes the features detected by the previous layer into more complicated features that are passed on to the following layer: visual input is processed into lines, lines into shapes, and so forth until finally, distinct objects are detected. Within a layer, the feature detectors are replicated over the features to be processed.

Convolutional neural networks are feedforward artificial neural networks with an input layer, one or more hidden layers, and an output layer [134]. Commonly used layers are convolutional layers, pooling layers, and fully connected layers [135, 138, 140].

Convolutional layers are the eponymous layers of CNNs. They contain one or more filters that perform the convolution operation over the input image to produce the feature map of the following layer [134, 135]. The size of the feature map can change in the process, usually by decreasing its resolution. Convolutional layers can be thought to arrange the nodes of traditional ANNs in three dimensions: width, height, and depth [134, 140]. Width and height describe the resolution of the resulting feature map and are determined by the stride parameter, zero padding, and filter size. Depth of the feature map equals the number of filters used on the input [134]. To introduce nonlinearity, the outputs of the convolutional layer are usually activated (see section 4.4.

Pooling layers can be used after the convolution layer to reduce the resolution of the image [134]. They split the input feature map into equally sized tiles and create a condensed feature map by picking an element from each tile into the output feature map [134]. Pooling can be non-overlapping, where no tiles share any elements, or overlapping, where the boundary elements of a tile are shared with one or more other tiles [134]. The commonly used pooling method is max pooling, where the maximum element from each tile is selected into the condensed feature map [134]. Max pooling introduces local invariance, where small changes in input do not affect the output. Furthermore, max pooling can help in avoiding overfitting.

Fully connected layers simplify the data pattern back to a vector and treat is as the input to a standard fully connected feedforward neural network. This process produces the desired final output. The last layer of a CNN architecture is often a fully connected layer [133, 140].

4.4 ACTIVATION FUNCTIONS

In a typical node, a bias is added to the linear combination of inputs and weights and the result is inputted in an activation function 4.5 [132]. The bias is a special weight that applies an offset to the activation function. The final output of the node is the output of that activation function [139]. This activation function is typically nonlinear, and ensures that the ANN is not simply a series of linear combinations [136]. If it were, it could only solve linear problems, e.g., approximating linear functions [134].

Common activation functions include sigmoid functions [132, 134, 139] (as illustrated in 4.5) and the rectified linear unit (ReLU) function [20, 134, 135]. Sigmoid functions are bounded and differentiable for all real-valued inputs, which makes them excellent for ANNs. The fact that they are bounded ensures that the output of the network is limited to a manageable range, and because the range is between 0 and 1 (logistic sigmoid) or -1 and 1 (hyperbolic tangent) for common sigmoid functions, they have uses in binary classification tasks. Differentiability in the entire real domain enables the use of gradient-based learning methods such as backpropagation [132]. However, because the derivative can approach zero at very small and very large inputs, sigmoid functions are vulnerable to the vanishing gradient problem (see section 4.5.3).

The ReLU function is defined as [134]

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} . \tag{4.1}$$

While it has no upper bound, it is differentiable for all real-valued inputs except zero (where the derivative can be chosen to be 0 or 1 in practical implementations) and both it and its derivative function are very simple to compute. The use of the ReLU function instead of sigmoid functions can eliminate the vanishing gradient problem. However, it is insensitive to negative inputs, which may in some cases reduce its usefulness. To cimcumvent the problem, "leaky ReLU" has been introduced as

$$\text{LReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ ax & \text{if } x < 0 \end{cases} , \tag{4.2}$$

where a is a positive constant greater than but usually close to zero.

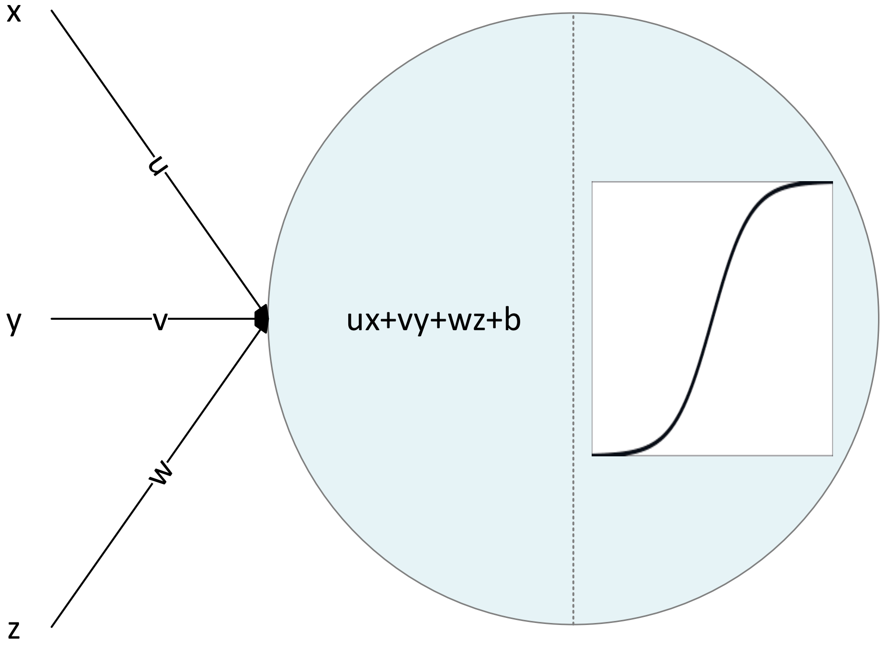

Figure 4.5: Illustration of a node in an artificial neural network. A bias term b is added to the linear combination of inputs $[x, y, z]$ with weights $[u, v, w]$. The result is inputted to a nonlinear activation function whose output is the final output of the node.

4.5 TRAINING OF ARTIFICIAL NEURAL NETWORKS

4.5.1 General principles

The training of artificial neural networks involves adjusting their weights (and possibly other parameters) such that the output of a loss function is minimized [134]. In supervised learning, the loss function is usually defined as the error between the output of the network given some input data and and the "true" value for that input data [134]. Therefore, training in supervised learning ideally adjusts the weights such that the inputs produce the corresponding, already known outputs [139].

Variables that control the training process are called hyperparameters [134,135]. They define how much each training sample affects the weights during a single iteration, how many iterations are looped during training, what kind of function or process directs the training, and so forth. Some hyperparameters define the network's structure and remain in the network even after training, such as the number of nodes in each layer or the activation functions in those nodes.

The motivation of training artificial networks is to make the network able to solve a particular problem or a type of problems [134]. Usually, not all data related to all possible cases of the problem are available [134] and thus, the network must be trained such that "generalization error" is minimized [136]. Generalization error describes the performance of the network in solving the problem type in the general case, not only constrained to the kind of data that is available during training [38, 134]. In other words, it is a metric of how well the network performs on data it has not "seen" before [136].

4.5.2 Training, validation, and test sets

In terms of data, the training of ANNs involves three subsets of data: a training set, a validation set, and a test set [38,63,75,134,143]. The subsets consist of paired input and output values, with each input having a corresponding output and vice versa [139]. In the pair, the input itself may be a single scalar or a vector of scalars; the same applies to the output. Furthermore, the dimensions of input and output in the pair do not have to match (e.g., an input-output pair may consist of 6 input scalars and one output scalar). The output constituent of the pair is called the desired output or the ground truth output [136]. This term distinguishes it from the output that the network gives for the input [136], which in prediction tasks is called the predicted output or the prediction.

The training set comprises the data that is used to train the network, i.e., adjust its weights [75, 143]. This is done iteratively by feeding an input from the training set into the network, computing the output, comparing it to the target output from the training set and adjusting weights such that the difference between the network output and the target output decreases. This difference, or a function of it, represents training error. A set of operations ending with calculating the difference between network output and target output and adjusting the weights of the network is called an epoch [134]. Training of complicated networks may take a very large number of epochs before finishing. Ideally, the training process will reduce the generalization error of the network [134]. However, it is usually undesirable to completely minimize the difference between the network output and the target output using the training set [134,135]. After a point, decreasing the difference further usually no longer decreases generalization error but will instead start increasing it again as the network becomes trained to solve the problem type represented only by the relationships in the training set [134]. This situation is called overfitting, and it is one of the main obstacles to overcome during training [134,135]. Similarly, too little training will result in underfitting, where the generalization error remains high and could still be reduced by training the network using the training set.

The validation set is used to prevent overfitting [134]. The idea of a validation set is to separate a subset of the data used during training and instead of using it directly to change the values of the weights during training, using it to evaluate the error of the network between training epochs on the data in the validation set [75,134]. The error evaluated on the validation set is called the validation error and it provides a rough estimate of the generalization error during the training process. Because validation error has local increases and decreases as epochs go by, halting training after the validation error stops decreasing the first time may underfit the model. On the other hand, if validation error is ignored and training processes until training error is minimized, the model is overfitted [38]. Hence, a way to achieve a good balance between underfitting and overfitting is to stop training after the validation error has not decreased for a set number of epochs (e.g., 10), and take the model's state at the last epoch that resulted in decreased validation error as the final trained model [75,136]. This method is known as early stopping [136]. The validation set can also be used to optimize the other hyperparameters of the model [134].

After training, the ANN is typically used on data that wasn't used during its training. To evaluate how the ANN performs on such data, a test set is required [75]. This test set must be independent from the training and validation sets or its use with the network will result in misleading performance metrics [38,134,143]. Because of this independence, test error is usually reported as the best estimate of the generalization error of the network, and used to compare different networks [38,63].

An important distinction between the validation and the test subsets is that the validation set is used during the training process, while the test set must be kept separate from it and only used to evaluate the model after its training is completed [75,138,143]. Failure to keep the test set separate from the training set will result in "data leakage" where the model can learn from features in the test subset, resulting in the perceived generalization error calculated against the test subset to be lower than it actually would be if used with independent data [38,143].

4.5.3 The backpropagation algorithm

Backpropagation is an algorithm that has been used extensively in training multilayer ANNs [134,136,139]. It has a strong mathematical foundation and is fairly understandable. Furthermore, the neural networks used in studies **II** and **III** employ backpropagation training. For these reasons, the working principles of the algorithm are briefly presented here.

Backpropagation consists of two phases: the forward pass and the backward pass [139]. In the forward pass, input is fed to the network to calculate the predicted output [136,139]. The predicted output is compared to the desired output to evaluate the error of the network [136]. The error is evaluated using a loss function which can be the simple difference between the prediction and the desired output or a more complicated function involving them. In the backward pass, the error is used to adjust the weights of the network so that the error of the next iteration is reduced. The process of alternating between forward pass and backward pass is repeated for a desired number of iterations or until the error is smaller than some threshold.

In the backward pass, the weights are adjusted starting from the output layer and passing backwards all the way to the weights between the input layer and the first hidden layer [134]. Computational complexity increases as the algorithm passes towards the first layer of the network.

Mathematically, the necessary equations can be found by applying the chain rule of derivation. The steps won't be repeated here but are presented in many literature sources on artificial neural networks, e.g., [132,134,139].

Although backpropagation is very useful, it has caveats. Two major ones are the vanishing gradient problem (also called "network paralysis" in some sources [139]) and getting stuck in local minima [134]. Fortunately, solutions have been proposed to both problems.

The vanishing gradient problem occurs when the derivative of the activation function is close to zero [139]. Because the derivative of the activation function affects how much the weights are adjusted each iteration, a derivative that is effectively zero makes training extremely slow [139]. For instance, sigmoid functions are common activation functions and their derivatives are close to zero at very small and very large inputs (Figure 4.6, left side). Hence, the vanishing gradient problem is quite common.

Solutions to the vanishing gradient problem include modified activation functions and careful pre-processing of the input data. Activation functions with manageable derivatives have been proposed, notably including the ReLU function (Figure 4.6, right side) and its variations (e.g., leaky ReLU if nonzero derivative is desired with negative inputs). Pre-processing of the input data (e.g., by centering mean to zero and normalizing to a standard deviation of one) can help in managing the magnitude of the inputs to each activation function.

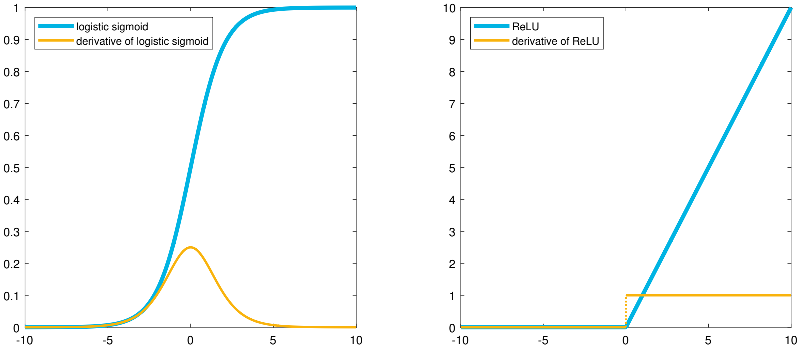

Figure 4.6: The logistic sigmoid (left) and ReLU (right) activation functions and their derivatives. Note that although the derivative of the ReLU function has a discontinuity at 0, the derivative can be chosen to be 0 or 1 for the purpose of using it in training algorithms.

Instead of finding the global minimum of the loss function, the training can get stuck in a local minimum [134,139]. In an arbitrarily small neighbourhood of a local minimum of a function, the gradient of the function guides the weights to converge to the local minimum (Figure 4.7 shows an example of a loss surface with local minima) [134]. Because training usually occurs in small steps, it is not uncommon for the algorithm to happen upon and get stuck in a local minimum. In such a case, the optimal weights are never found.

Introducing momentum to the gradient has been proposed as a way to avoid getting stuck in local minima [139]. The idea of momentum is that the weights are

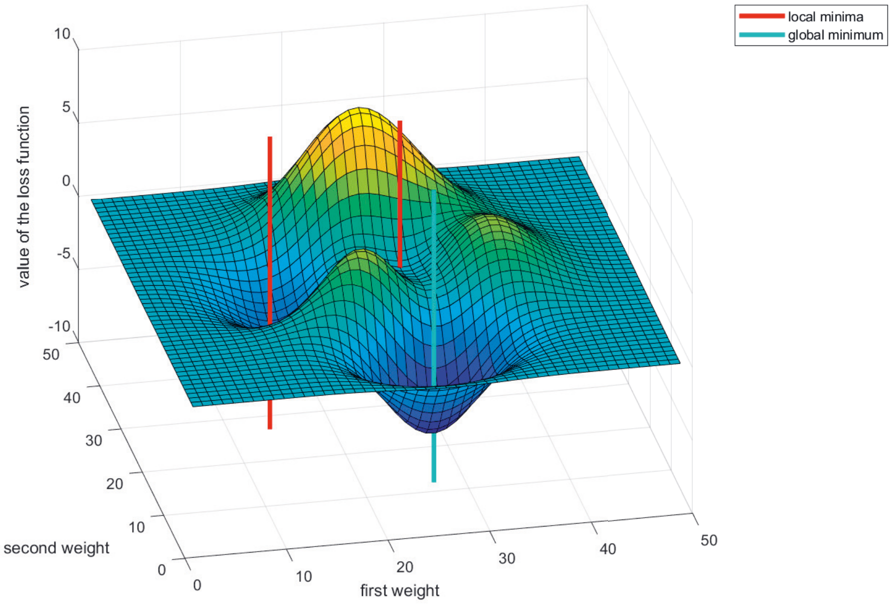

Figure 4.7: Demonstration of a loss surface with two local minima and a global minimum. The minima are indicated by the vertical lines running through them. Note that for ease of visualization, there are only two weights to optimize; in most artificial neural networks there are more than two weights.

not updated only according to the gradient at the solution of the current iteration, but also according to the direction of the updates from previous iterations [134]. Therefore, the training process has "momentum": if the updates have been done in a particular direction for several recent iterations, a single iteration of an opposite or otherwise different gradient has only a small effect on the adjustment of the weights. Hence, during a single iteration, the weights are updated according to the combined effect of several recent iterations rather than relying only on the current iteration (which could throw the weights into a local minimum instead of towards a more global solution). This method can help the algorithm in escaping the neighbourhoods of local minima when they are encountered.

4.6 NEURAL NETWORKS IN MOTION ANALYSIS

In section 3, the reader was introduced to how motion data is captured and analyzed through musculoskeletal modeling and simulation. Section 4 presented the basic principles of two types of neural networks, fully connected feedforward and convolutional. Here, we describe how neural networks are applied to solve two prominent issues in motion capture and musculoskeletal modeling and simulation: 1) the cumbersomeness of capturing motion capture data and 2) the complexity of analyzing the captured motion data (briefly presented in section 3.5).

Capturing motion capture data and the subsequent analysis is troublesome for the following reasons:

1. motion capture requires a large indoor space with expensive equipment [63,79]
2. said equipment needs to be moved to fit the needs of individual studies and calibrated afterwards [76,78]
3. the subject of study must travel to the motion laboratory for the measurements [79]
4. the measurements involve a preparatory stage where markers are placed on the skin of the subject [79]
5. markers must be placed by trained personnel [63,79]
6. trained personnel must operate the measurement devices and software [21]
7. the analysis involves a multi-stage workflow which takes time [19,20]

Point 1 creates a monetary barrier to using motion analysis. Points 2, 3, and 4 increase the time a study of motion study requires before even allowing the researchers to start analyzing the collected data, while points 2, 4, 5, and 6 employ special expertise. While the individual steps of point 7 can be implemented in real-time so that the results of the analysis are immediately available during the measurement [29,126], existing studies have focused on the inverse kinematics [94,126] and sometimes the inverse dynamics stage [29] or are not modular and adaptable to different musculoskeletal models [26]. Additionally, sometimes achieving real-time throughput has necessitated the reduction of the accuracy of algorithms used by the workflow [94].

Luckily, artificial neural networks can be leveraged to overcome these obstacles. Here, we split the use of artificial neural networks into two partially overlapping categories: those where ANNs are used to estimate some intermediary outcomes of the study (e.g., keypoints in an image) and those where ANNs are used to estimate the final outcomes of the study (e.g., joint moments or forces). Among the former, human pose estimation algorithms stand out as their own solution. Among the latter, knee dynamics are a common subject of study, sometimes utilizing the former in the process. Next, we will introduce some existing work in both categories.

4.6.1 Human pose estimation with convolutional neural networks

Although both feedforward and convolutional neural networks have been directly applied to predict joint dynamics, the use of convolutional neural networks also enables human pose estimation (HPE) algorithms [33,35], which have been adapted in the study of human biomechanics [23,42,144]. In biomechanics, HPE is used to infer the kinematics of a subject or subjects from images, typically from a video recording. Most HPE algorithms achieve this by identifying keypoints, which are anatomical locations on the human body [33,141]. Keypoints are usually discernible locations between limbs within the human body (e.g., ankle, knee, hip, shoulder, and elbow joints) or otherwise recognizable features (e.g., eyes, nose, mouth, fingers, toes), but can also be inferred behind occlusions by some HPE algorithms [23,43, 77,145]. The trajectories of keypoints are typically given in the 2D coordinates of the image, but can be combined from multiple camera views to reconstruct 3D trajectories [39,42].

HPE algorithms are trained on large datasets of labeled images that are available online. These datasets include the ImageNet [146], COCO [147], and MPII [148] datasets [145]. Although the amount of available training data is satisfactory, the images are usually labeled by amateurs rather than experts of human physiology. Therefore, some disagreement exists between the positions of the keypoints identified by HPE algorithms and true positions of the anatomical features those keypoints represent [22,23].

Because HPE algorithms are not created for the sole purposes of biomechanical studies, the researcher must take their output and use it to calculate biomechanically relevant parameters [23]. This can be achieved by tracking the trajectories of keypoints and treating them as motion capture markers [22,149]. Usually this sort of analysis is done on a plane (e.g., to calculate 2D kinematics) because the HPE algorithms usually operate on 2D images, but augmented neural networks solutions exist where the HPE algorithms can infer the 3D trajectories of the keypoints from single camera view 2D images [150], depth camera data is utilized to transform 2D keypoints to 3D keypoints [43,44], or 2D keypoints from multiple camera views are transformed to 3D [39,42,145]. The use of keypoints serves the needs of biomechanical studies well, because keypoint trajectories can be used in place of marker trajectories from laboratory-grade motion capture systems and fed to musculoskeletal modeling and simulation pipelines [145].

Currently available HPE algorithms include AlphaPose [35], BlazePose [151], DeepLabCut [34], OpenPose [33], and YOLOv7 [36], just to name a few. Their performance in different tasks has been studied in literature. For instance, Needham et al. (2021) found that between AlphaPose, DeepLabCut, and OpenPose, the accuracy of joint center localization when compared against MOCAP-based joint centers depended on the joint but systematic differences differences of 3–5 cm were observed for all three HPE algorithms [22]. These differences were attributed to mislabeling of ground truth datasets and are important to acknowledge because hip and knee are pivotal joints in the study of human biomechanics. Moreover, HPE localizations of the ankle joint had low systematic and random error when compared to the hip and knee joints. Needham et al. believed that this relatively low error resulted from the ankle joint being easy to identify and label in the training data. It seems plausible that the more soft tissue or clothing surrounds the bony landmark, the more difficult pinpointing its location in images becomes. Therefore, obese subjects may present additional challenges to accurate HPE-based localization of joint centers, which could be problematic because some conditions that are studied using motion analysis have been linked to obesity (e.g., knee osteoarthritis) [7,10,54].

One of the best-known HPE algorithms, OpenPose, works by iteratively refining estimates of body part positions in the image (Figure 4.8) [33]. OpenPose is very robust and identifies keypoints on many different human shapes as demonstrated in Figure 4.9. In biomechanics, OpenPose has been used, e.g., to estimate the hip-knee-angle of KOA patients from frontal images [129], to estimate spatiotemporal gait parameters [40], to estimate joint kinematics during walking, running, and cycling [23,40,77,145,152], and to estimate joint dynamics of cyclists [104]. It has been shown to estimate knee kinematics as accurately or even more accurately than many competing freely available HPE algorithms [22,40,152]. For instance, Washabaugh et al. (2022) compared OpenPose to the Tensorflow MoveNet Lightning, Tensorflow MoveNet Thunder, and DeepLabCut pose estimation algorithms against marker-based optical motion capture and found that OpenPose outperformed the others in estimating the kinematics of healthy gait and spatiotemporal gait parameters, noting that OpenPose excelled over the others particularly in estimating knee kinematics [40]. Van Hooren et al. (2023) made similar observations when comparing OpenPose and DeepLabCut against marker-based optical motion capture [152].

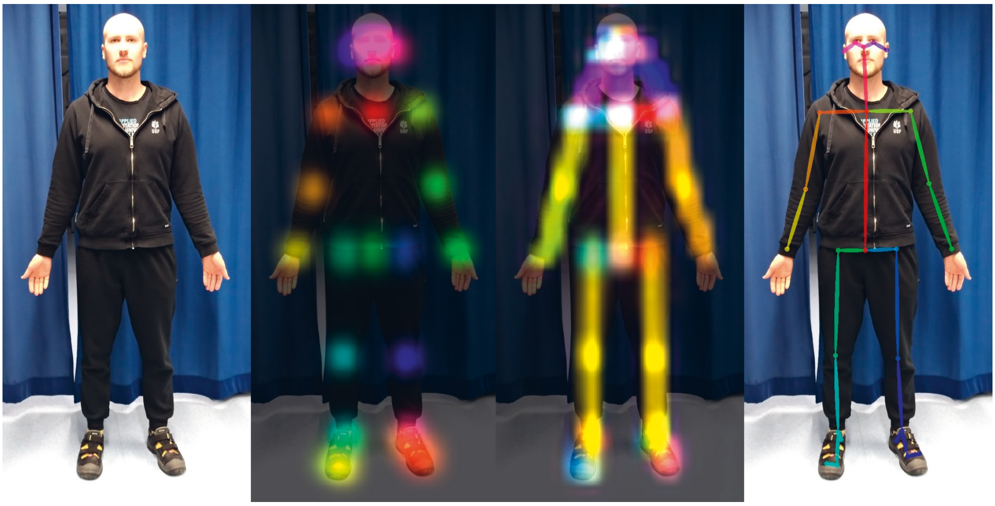

Figure 4.8: The workflow OpenPose uses to detect keypoints from an input image using the BODY_25 model. Confidence maps of body part positions are estimated in the original image and body parts of the person are associated with one another to locate the keypoints for each body part.

OpenPose follows a bottom-up approach, meaning that it first detects body parts in the image and then assigns each detected body part to a person. This approach is an alternative to top-down approaches, which first detect different people in the image and then try to detect the body parts belonging to each person [43]. All body parts do not have to be visible at the same time, although detection works best when they are [145].

Like in many other HPE algorithms, the main outputs of OpenPose are called keypoints. They are the positions of various anatomical landmarks on a 2D image. In addition to the 2D position, OpenPose also outputs a confidence value for each keypoint. This value describes how sure OpenPose is of the correct identification of the keypoint and ranges from 0 to 1. Confidence of OpenPose keypoints has been shown to correlate nonlinearly with keypoint detection accuracy [41].

Although OpenPose can fail to correctly identify keypoints especially in the case of limb occlusions, challenging environments and motions, or high velocity [23, 24, 104], methods exist for correcting keypoint trajectories similarly to motion capture marker trajectories, i.e., by utilizing assumptions of the motion and information about the previous positions of the keypoints [23, 104]. Due to its usability, OpenPose has been investigated as an alternative to laboratory-grade motion capture systems in clinical and other out-of-the-laboratory environments [22, 23, 39–41, 77, 104, 129, 152]. However, it likely performs best in controlled environments instead of sports venues where the lighting and camera conditions are far from ideal [24]. Furthermore, while OpenPose can detect keypoints on occluded limbs, their detection accuracy is lower than those of visible limbs [23].

4.6.2 Neural networks in estimating joint dynamics

Where human pose estimation algorithms rely mostly on convolutional neural networks, dynamics of motion can be estimated with a variety of different types of networks including not only convolutional neural networks, but also fully connected feedforward neural networks and recurrent neural networks [19,75]. In some cases, these solutions utilize human pose estimation algorithms, e.g., to substitute keypoint trajectories for marker trajectories [39,40].

Existing applications of artificial neural networks in estimating joint dynamics mostly involve feedforward networks. For instance, Ardestani et al. (2014) trained feedforward network models to predict knee joint contact forces in the medial compartment after gait modification [18]. They evaluated three architectures of networks; the first trained on pre-rehabilitation marker and GRF data from a single subject to predict post-rehabilitation KJCFs of the same subject, the second trained on pre-rehabilitation GRF and EMG data of several subjects to predict the subjects' post-rehabilitation KJCFs, and the third trained on post-rehabilitation marker, GRF, and EMG data to predict post-rehabilitation KJCF of a subject whose data was not used during training. They found that the model trained on subject-specific data to predict the same subject's loading after gait modification was the most accurate. However, that model required motion capture data (marker trajectories and GRFs) during both training and its actual use. Similarly, Rane et al. (2019) used CNNs to predict medial KJCFs with combinations of marker and GRF data [19]. Giarmatzis et al. (2020) trained feedforward networks to predict compartmental KJCFs from marker and GRF data or just marker data, and observed that excluding GRF data did not greatly increase prediction error [20]. While the elimination of GRFs makes the method slightly more portable, marker data still requires motion capture equipment or a means to accurately estimate 3D marker trajectories.

Utilizing more accessible data, Mundt et al. (2020) used marker trajectories to calculate simulated IMU signals and used them for predicting joint kinematics and kinetics using fully connected feedforward network and long short-term memory network architectures [75]. Stetter et al. (2019, 2020) trained ANNs to predict net knee joint forces from IMU data during different movements and noted that some movement types are better suited to the prediction task than others [11,30]. In an interesting development, Zhang et al. (2022) incorporated laws of physics as a soft constraint into the neural network training process to develop a physics-informed CNN for predicting muscle forces and joint angles from surface EMG data [142].

Finally, in addition to artificial neural networks, other machine learning algorithms have also been used for similar purposes. For instance, Brisson et al. (2021) used regularized least squares regression to predict medial KJCF from gait speed, vertical knee reaction force, and knee adduction moment [128]; Long et al. (2023) predicted the kinematic and kinetic time series of ankle and hip joints with random forest regression using a single IMU on the ankle [32]; di Raimondo et al. (2023) used IMU data and a trained probabilistic PCA model to estimate GRFs and then used the estimated GRFs together with IMU-based kinematics to estimate knee joint contact forces [25].

These studies noted that machine learning algorithms provide a faster way to estimate knee joint loading than traditional musculoskeletal modeling and simulation and pave the way for real-time applications outside the motion laboratory, [18–20,25,30,32,75,128,142] where wearable devices could be used to eliminate the need for motion capture data [11,25,30,32,75].

Futhermore, the existing literature presented here exhibits two approaches to using trainable machine learning models for predicting the dynamics of motion: personalized and non-personalized [11,18,19].

In the personalized approach, the ANN models are trained using the data from the subjects that are also present in the test set, i.e., the models are personalized to particular subjects. This approach is in slight contrast to the principles presented in section 4.5; however, the personalized approach wilfully accepts a degree of data leakage as networks are trained and tested on data from the same subjects [20,89]. Such networks could be used, after initial thorough collection of training data, to make accurate intra-subject predictions with less cumbersome measurements. Thus, the networks would generalize poorly to other subjects, but provide more accurate subject-specific predictions after a thorough measurement session for obtaining the training data for that subject [89].

The non-personalized (alias generalized) approach is more stringent about the independence of the training, validation, and test subsets. It generally restricts the data samples belonging to one subject to just one of the subsets [19]. Its purpose is to develop prediction models that can be used for the general population.

The existing literature has also proposed solutions that use several independent networks in sequence (instead of just one multi-layer network), e.g., to first estimate kinematics with one network and then to use another network to estimate kinetics from the output of the first network [89]. This workflow could increase modularity, usability, and explainability of the methods as it separates the learning problems and allows training and re-training the networks independently from one another [89].

Despite these advances in research, existing studies are often limited to either requiring motion capture data in their use case or support only a specific portable modality [130]. Hence, there is a need for modular methods that can extract predictors from a variety of different portable modalities (instead of being tied to a specific modality) and utilize those predictors to predict knee joint dynamics. Such methods could facilitate the effortless prediction of knee joint dynamics out of the laboratory in various different settings without being tied to specific equipment or technologies.

Figure 4.9: OpenPose is very robust and works even with cartoon depictions of people. On the left, public domain images "Together We Win", a WWI era poster by James Montgomery Flagg, and the cover illustration of "The Dialogue Concerning the Two Chief World Systems", a 17th century book by Galileo Galilei. On the right, the keypoints detected by OpenPose are overlaid on the images. Note that the algorithm detects legs even though they are covered with gowns.

5 Aims

As described in the previous sections, structural damage in the knee joint may be linked to physical loading exerted in the joint. The loading cannot be measured non-invasively *in vivo*, and must instead be estimated using musculoskeletal modeling and simulation. This estimation requires collection of experimental motion data from live subjects and computational analysis of the data, which together require expensive equipment, trained personnel, a specialized location for the measurements where the subject must be be present for the length of the measurements, and time to compute the musculoskeletal simulations. These requirements rule out routine estimation of knee joint loading in clinical or other environments and prevent accessing the results in real-time or immediately after the measurements. Furthermore, the entire process is too cumbersome for routine use with people even in a motion laboratory. Available solutions to these problems include the use of artificial neural networks and other machine learning algorithms to estimate knee joint loading, but those solutions either require motion capture data, are developed for a specific measurement technology, thus making its adoption to wider use difficult, or are cumbersome and effectively unavailable for portable use.

Therefore, this thesis explores ways to estimate human movement with more accessible, modular, and portable ways. Particularly, we present methods to estimate human kinematics (i.e., joint angles) in-real time and to estimate knee joint loading from simple input data that can be measured with little effort. Specifically, the aims of this doctoral dissertation were as follows:

1. To develop a method for real-time analysis of inverse kinematics from inertial measurement unit data and quantify its performance on standard desktop and laptop computers (study **I**). We designed our method to use wearable inertial measurement units extend an existing open-source and commonly used software for musculoskeletal modeling and simulation. In the long term, we hoped that this could better enable real-time sports training and physical rehabilitation.

2. To develop simple and modular methods for estimating subject-specific knee joint loading outside the motion laboratory (study **II**). To this effect, we trained feedforward artificial neural network models for predicting compartmental tibiofemoral compressive contact force peaks from predictors (i.e., input variables) that do not necessitate the use of motion capture equipment. Although the predictors were designed to be obtainable without motion capture data, at this point we still calculated them from motion capture data, leaving actual demonstration of out-of-laboratory use for later. To obtain the tibiofemoral compressive contact force peaks, we used musculoskeletal simulationg with human models whose properties were adjusted according to subject-specific information. Although the musculoskeletal models were made subject-specific, we trained the prediction models to be non-personalized.

3. To test the knee joint loading prediction methods of study **II** using predictors that have been calculated from portable modalities usable outside the motion

laboratory (study **III**). In effect, the data of a single video camera or inertial measurement units; however, the prediction models themselves were designed to be technology-agnostic and do not necessarily require either of the two modalities. Thus, we showcased the modularity and portability of predicting knee joint loading peaks. Note that although we developed methods to estimate knee walking speed and knee flexion angle from the two modalities, our aim was not to develop state-of-the-art methods for doing so but to simply demonstrate possible out-of-laboratory use cases of our knee joint loading prediction models.

During the course of the studies and based on the like-numbered aims, we examined the following hypotheses:

1. Analysis of musculoskeletal model-based kinematics during gait can be conducted portably and in real-time using a laptop and wearable sensors. The workflow enables the use of different musculoskeletal models and sensor configurations.
2. Compartmental knee joint contact force peaks can be estimated from a very simple set of predictors.
3. The predictors used in predicting knee joint contact force peaks can be estimated using different portable modalities, namely inertial measurement units or a video camera recording motion in the sagittal plane.

6 Materials and methods

6.1 MOTION LABORATORY

All experimental motion capture data that we collected during studies **I** and **III** was collected in the HUMEA motion laboratory of the Department of Technical Physics, University of Eastern Finland. The HUMEA laboratory has a host of equipment for motion capture, presented in the subsections below.

6.1.1 Motion capture equipment

We used a marker-based Vicon motion capture system with 10 Vicon Vero optical cameras and a single Vicon Vue video camera (Vicon Motion Systems Ltd, UK). The Vicon motion capture system is one of the most common systems used in studies of human motion [3,4,11,20,52,72,75–77,82,89,96,99,102,109,115,117,153]. The cameras were embedded on rails running along the ceiling and on tripods on the floor so that they tracked reflective markers in the capture volume from many different angles. The placement of the Vero cameras was designed to minimize occlusion of markers by objects and human subjects, so that as many markers as possible would be visible to as many cameras as possible at any given frame.

During study **III**, the Vue video camera was placed approximately 4 meters from the line of walking at the height of the hip. The video camera was aimed and focused to record walking in the sagittal plane of the participants.

The floor of the capture volume includes three embedded force plates measuring at 1000 Hz the ground reaction forces and moments and the center of pressure of the object pushing against them (Figure 6.1). The participants were instructed to step on the middle one (AMTI OR 6–7, Advanced Mechanical Technology Inc, MA, USA) during the collection of the Kuopio gait dataset in study **III**.

The laboratory contains a Motek M-Gait (Motek Medical B.V., Netherlands) dual-belt instrumented treadmill (Figure 6.2). It has integrated force plates under each belt. The treadmill is placed within the capture volume of the Vicon Vero cameras and was used while collecting motion capture data in study **I**.

6.1.2 Inertial measurement units

We used the MTw Awinda (Movella Inc, Henderson, NV, USA) wireless inertial measurement units (IMUs) as one of the two portable motion tracking modalities (Figure 6.3). The IMUs utilize a proprietary Kalman filter that estimates the 3D orientation of the sensor from its tri-axial magnetometer, accelerometer, and gyroscope readings. The sampling rate of the orientation data depends on the number of IMUs in concurrent use. We used a total of seven IMUs concurrently with a sampling rate of 100 Hz.

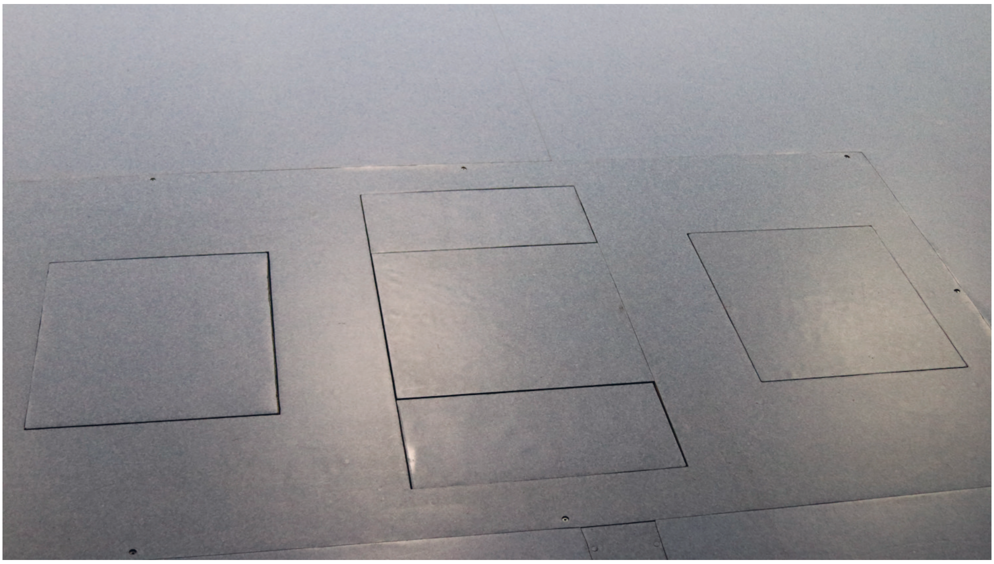

Figure 6.1: Three force plates are embedded in the floor in the HUMEA laboratory to ensure they do not hinder gait. Data from the middle force plates was used in study **III**.

6.2 MAGNETIC RESONANCE IMAGING FOR CALCULATING KNEE INTERCONDYLAR DISTANCE

Participants of study **III** underwent a low-field MRI (0.18T E-Scan XQ, Esaote SpA, Italy) scan of the knee of their dominant leg. The images were taken in the sagittal plane and reconstructed to retrieve projections in the frontal and transverse planes. From the projections, we extracted the intercondylar distances (ICDs) between the center points of medial and lateral condyles of the distal femur.

To extract the ICDs, we first located the boundaries of the condyles of the femur. Next, we calculated the midpoint of each condyle as the point exactly between the boundaries. Finally, we calculated ICD as the distance between the midpoints (Figure 6.4). The ICD was used to scale the knee of the musculoskeletal model in the mediolateral direction in study **III** (see section 6.6).

6.3 COLLECTION OF THE KUOPIO GAIT DATASET

In study **III**, for testing the knee joint contact force (KJCF) prediction models trained using the combined dataset (put together in study **II**), we collected a dataset of 51 willing participants in the HUMEA motion laboratory, University of Eastern Finland, Kuopio, Finland.

The participants participated in walking measurements where their gait was captured with the Vicon Nexus marker-based motion capture system and floor-embedded force plates. Concurrently, sagittal-plane video of walking was recorded with the Vicon Vue video camera, and inertial data was collected using the MTw Awinda IMUs. Finally, magnetic resonance images were taken of the knee of the leg

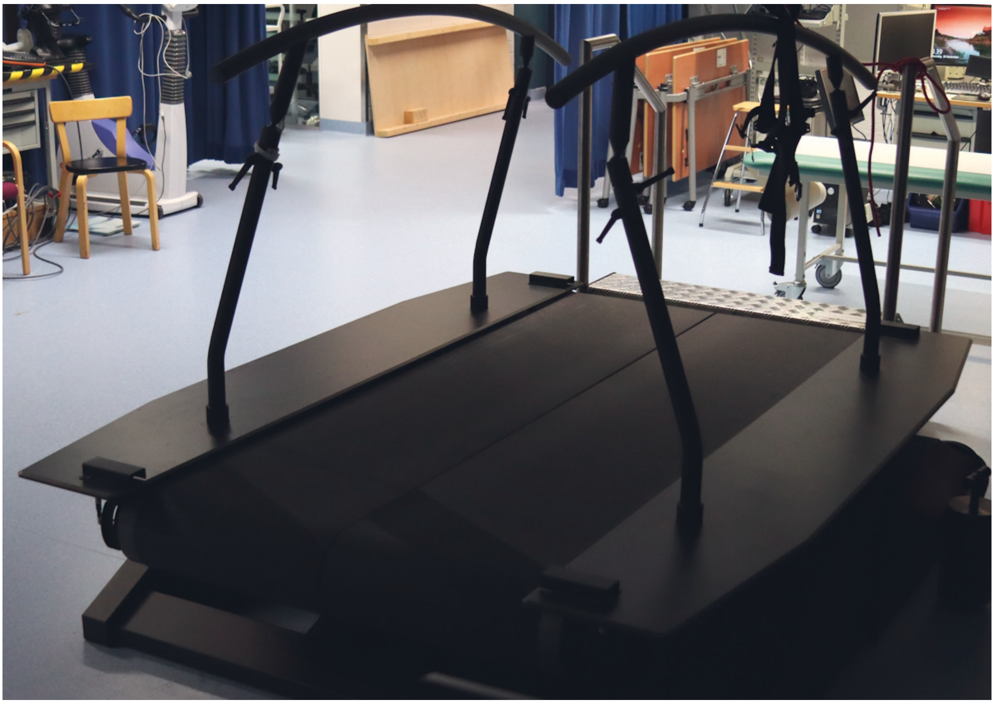

Figure 6.2: An instrumented Motek M-Gait treadmill was used while collecting motion capture data in study **I**.

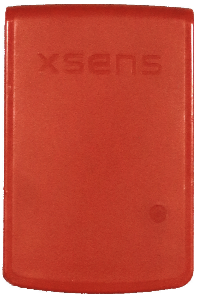

Figure 6.3: In studies **I** and **III**, we used MTw Awinda IMUs. Their small size of 47 mm × 30 mm × 13 mm makes them convenient to wear.

the participants identified as dominant.

Due to data processing difficulties, the data of five participants were not included and analyzed in study **III**. The remaining 46 participants (29 male, 17 female) were between 20 and 45 years of age (mean ± std 28.7 ± 5.9) and had masses between 54 and 136 kg (76.3 ± 14.5 kg), heights between 1.61 and 1.89 meters (1.74 ± 0.7 m), and body mass indices between 18.8 and 40.4 (25.1 ± 4.1).

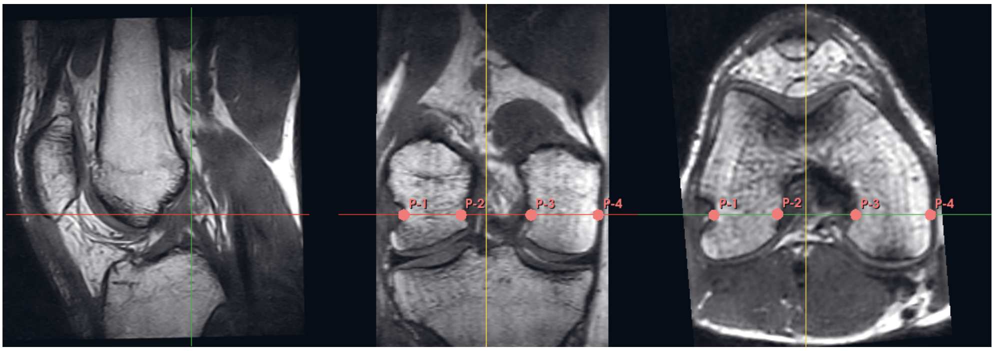

Figure 6.4: Magnetic resonance images in the sagittal plane (left) were used to reconstruct images in the frontal (center) and transverse (right) planes. Annotated points are the boundaries of the condyles at the level of the lateral depression of the lateral condyle (i.e., the popliteal groove). Knee intercondylar distance was calculated as the distance between the midpoints (between P-1 and P-2, and P-3 and P-4) of the condyles. Red line shows the transverse plane, green the frontal plane, and yellow the sagittal plane.

6.3.1 Marker and IMU placement

When possible, we wanted to subject the IMUs to the same soft tissue movement as the markers. Thus, to aid in the placement of markers and IMUs, we designed and 3D printed marker clusters for the pelvis and the thighs (Figure 6.5). The base shape of the clusters was rectangular, with extending posts in each corner, an indent slot for an IMU in the middle, and two rails to slip through a strap that would also hold the IMU in place. The back of the pelvis cluster was flat, while the backs of the thigh clusters were curved to better fit against the convex surface of the thigh.

IMUs were fitted to a slot on the pelvis marker cluster and the thigh marker clusters. On both shanks, a single IMU was strapped on the skin distolaterally. On both feet, a single IMU was taped centrally on the metatarsal. This resulted in a total of seven IMUs, tracking the inertial motion of the pelvis, the femurs, the tibias, and the feet.

A total of 42 reflective markers were placed on each participant. On the torso, single markers were placed on the shirt on the manubrium of the sternum, on the 7th cervical vertebra, and on the acromion of the left and right scapula. Four markers were placed asymmetrically on a marker cluster that was strapped behind the pelvis. On both thighs, four markers were placed asymmetrically on a marker cluster that was strapped to the skin on the lateral side, and single markers were placed on the medial and lateral epicondyles of the knee. On both shanks, four markers were placed on a marker cluster that was taped on the skin on the lateral side, and single markers were placed on the medial and lateral malleoli of the ankle. On both feet, single markers were placed on the skin posteriorly to the calcaneus, on the first distal phalanx and on the fourth proximal phalanx. Finally, two markers were placed anteroposteriorly on the IMU on the metatarsal.

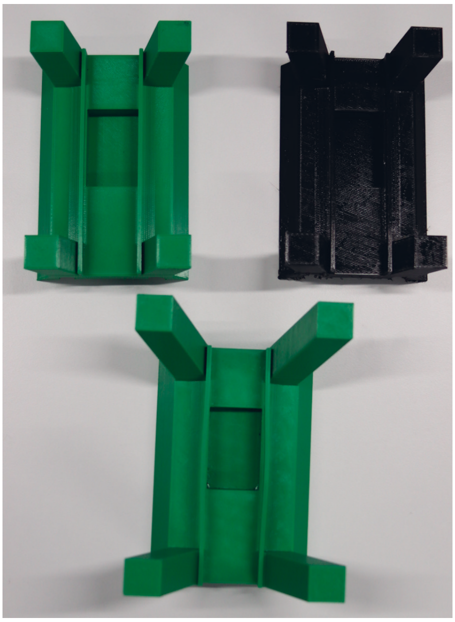

Figure 6.5: 3D printed clusters for markers and IMUs were used in study **III**. During measurents, a marker was attached to each of the four posts in each cluster.

6.3.2 Walking measurements

All walking trials were synchronously recorded with the optical motion capture system (marker trajectories and force plate data), IMUs, and video camera capturing sagittal-plane motion — a plane where joint angles can be measured with good agreement with respect to standard methods [42, 86]. A single walking trial started with the participant standing still outside the view of the video camera, accelerating to an approximately constant walking speed, walking over the force plates while stepping on the middle force plate with their dominant leg, and decelerating after passing the force plates. The trial ended with the participant standing still outside the view of the video camera a few meters after the force plates, facing the same way as they did when the trial started. Because the trials involved no systematic rotation around the vertical axis and a single trial only lasted a few seconds, we could ignore the effect of accumulating drift error [93, 100] and make assumptions about the motion for later estimation of predictors (section 6.8).

A walking trial and its recording started when the operator of the motion laboratory clicked a button to start the IMU measurement. Clicking the button sent a voltage signal to the Vicon motion capture system as a cue to start capturing. Thus, the motion capture system started the collection of marker trajectories and force plate data and the video camera recording. When the participant finished, the operator clicked a button to stop the IMU measurement, which sent a terminating voltage signal to the motion capture system. This procedure ensured that the collected data from marker trajectories, force plates, video camera, and IMUs were synchronized.

Ten successful walking trials were recorded per three instructed walking speeds:

the participant's comfortable walking speed, a walking speed 25% slower than comfortable, and a walking speed 25% faster than comfortable. The walking speeds were not controlled and they were based on the participants' own judgment. On average, the participants walked at $1.11 \pm 0.18\frac{\mathrm{m}}{\mathrm{s}}$, $1.35 \pm 0.17\frac{\mathrm{m}}{\mathrm{s}}$, and $1.55 \pm 0.23\frac{\mathrm{m}}{\mathrm{s}}$ during slow, comfortable, and fast trials, respectively (Figure 6.6).

Furthermore, because we wanted to assess how occlusion by the non-dominant leg affects video analysis of the dominant leg, these 30 trials were repeated with the participant walking in the opposite direction. In summary, we had six trial configurations and collected a total of 60 successful walking trials per participant (10 slow, 10 comfortable, and 10 fast walking speed trials, and another set repeated in the opposite direction).

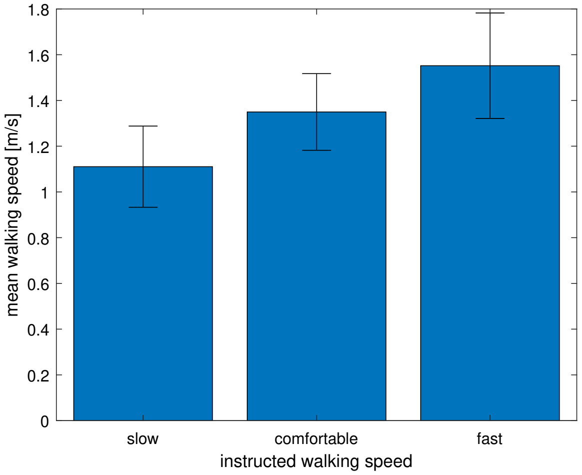

Figure 6.6: The inter-participant mean (bar height) and standard deviation (error bars) of intra-participant mean walking speeds for different instructed walking speeds (slow, comfortable, and fast). The walking speeds are calculated from pelvis marker trajectories of the Kuopio gait dataset collected in study **III**.

6.4 COLLECTION OF EXISTING GAIT DATASETS

We combined five existing motion capture datasets: four open datasets of healthy subjects [154–157] and one dataset of knee osteoarthritis patients [158] obtained from research collaborators. This sort of pooling of different datasets is recommended by Halilaj et al. (2018) in their literature review, where they observed that datasets used in biomechanics studies usually only involved data from a single laboratory [38]. All five datasets included marker trajectories from optical motion capture and ground reaction forces and moments from floor-embedded force plates, measured during walking trials of the subjects. The datasets were in similar formats and all datasets contained overground walking trials, so there was little need to standardise the data. However, because they used different marker sets, we defined unique model scaling and inverse kinematics setup templates for each dataset.

The data was analyzed with musculoskeletal modeling and simulation and used in studies **II** and **III**, where it is referred to as the "combined dataset". In study **II** the data was used to train and evaluate feedforward neural networks for predicting KJCF peaks. In study **III**, the data was used to train feedforward artificial neural networks, but data from the dataset we collected ourselves (see section 6.3) was used as the test set.

6.5 ENABLING REAL-TIME PROCESSING OF INVERSE KINEMATICS WITH OPENSIM

In study **I**, our plan was to use the OpenSim API to write a software tool that implements the inverse kinematics functionalities of OpenSense in real-time. Because OpenSim is written in C++, we also chose to write our software tool using C++, which allowed us to leverage the classes of OpenSim directly.

We based our programmatical real-time IMU IK class on that of OpenSense, modifying it to read and analyze data one frame (i.e., orientations of an arbitrary number of IMUs in a single time point) at a time. To improve the throughput of our tool, we used thread pooling to enable parallel processing of IMU orientation frames.

To pass data between IMUs and the OpenSim API, we wrote classes for reading data from two IMU manufacturers (Xsens and Delsys) using the Xsens Device API and the Delsys Software Development Kit. These classes interfaced with the manufacturers' programs to read IMU orientations into a data structure that could be utilized by the OpenSim API.

Finally, we evaluated the performance of the software tool by analyzing previously recorded IMU orientations frame by frame in a simulated real-time analysis, and by analyzing walking on a treadmill in real-time during actual live measurements. The simulated real-time analysis was performed on two computers, one a standard motion laboratory desktop computer and the other a standard laptop, which represented a computer one could easily use outdoors or in other portable situations. The live measurement was analyzed using the desktop computer to see how the possibly reduced sampling rate caused by real-time analysis would affect the range of motion of joint angles compared to offline measurement at the full sampling rate of the IMUs.

The IMU orientation data from 12 IMUs was measured from a volunteering participant. Ten walking trials, lasting approximately a minute each, were recorded on the treadmill. The placement of IMUs during the measurements is shown in Figure 6.7.

When developing the software tool, our aim was to acknowledge the drawbacks of existing real-time analysis studies and develop a solution that is not reliant on a specific data set, measurement setup, or musculoskeletal model [26,28,37,159–161], and could instead be used plugged in different studies to provide an analytical and physics-informed solution, much like any other OpenSim tool. The software tool can be built from its source code, which is available at `https://github.com/jerela/OpenSimLive`.

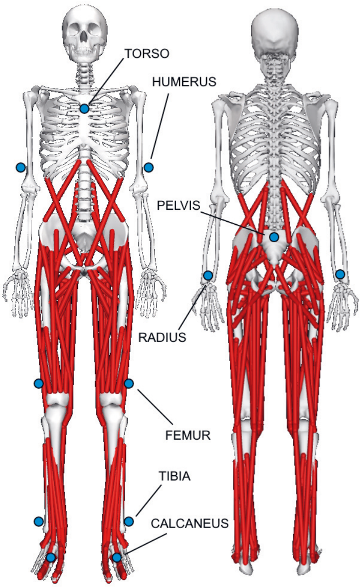

Figure 6.7: The placement of 12 inertial measurement units (blue circles) on the subject in study **I**. Image is taken from the article of study **I**, published in PeerJ (`https://doi.org/10.7717/peerj.15097`).

6.6 MUSCULOSKELETAL MODELING AND SIMULATION

In study **I**, we conducted musculoskeletal modeling and simulation to solve the joint angles of a musculoskeletal model in real-time. We used two different musculoskeletal models, the Gait2392 lower extremities and torso model [88,112,162,163] and the Hamner full-body model [164]. Although the models were generic, we did not scale their bony dimensions or muscles because we worked directly with orientation data and wanted to solve only joint kinematics. Therefore, the pipeline comprised the registration of worn IMUs on the musculoskeletal model using a static standing calibration pose and then the estimation of the pose of the model that best matched the experimental orientations measured from IMUs.

In studies **II** and **III**, our musculoskeletal modeling and simulation pipeline was more thorough and comprised the following steps:

1. model scaling to turn generic musculoskeletal models subject-specific
2. esimation of model kinematics to solve joint angles during the motion
3. estimation of model kinetics to solve the forces and moments in each joint during the motion
4. estimation of muscle activations to solve how much force different muscles produced during the motion
5. estimation of internal forces to solve how much force is applied on joints.

Similarly to [114], in studies **II** and **III**, we used the Rajagopal model [165] but removed the arms while adding their mass to the torso, and included a bicompartmental knee model developed by Lerner et al. (2015) [166]. During step 1, the bony dimensions of the modified model were scaled according to subject anthropometry [15] and allometric scaling with respect to subject mass with an exponential factor suitable for lean populations [167] was used to adjust muscle strengths [118].

The scaling step also involved tuning the positions of virtual markers on the scaled model and setting the initial pose of the model. To this end, in study **II**, we used marker data from static standing, whereas in study **III**, we additionally used functional calibration methods to determine the centers and axes of rotation of the hip and knee joints [81, 168–170]. Although those methods were mainly developed to reduce soft tissue errors in marker data and enable accurate localization of joint centers and axes of rotation, we used the estimated joint positions to calculate scaling factors for the segments of the model, as they have been shown to perform relatively well compared to other similar methods [153]. Because of possible error in lateral position of the hip joint center [171] (which we also observed during initial analyses), we did not use the functionally determined hip joint centers to scale the pelvis. Instead, we used palpation-based measurement of the participant's inter-ASIS distance (i.e., the distance between the two anterior superior iliac spines of the pelvis) to scale the pelvis segment. In study **II**, the knee abduction-adduction angle during static standing was also estimated in the scaling step.

Finally, we included small reserve actuators in hip, knee, and ankle joints with a stronger reserve actuator in the hip flexion direction to account for moment-inducing effects of elastic structures (e.g., ligaments) not explicitly included in the musculoskeletal model [172].

The pipeline in study **II** was validated against in vivo data; in study **III**, we changed some parameters, most importantly by scaling knee width from magnetic resonance images rather than based on subject height like we did in study **II**. However, this change should only make the modeling more realistic and subject-specific [166].

The outputs of the musculoskeletal modeling and simulation pipeline included joint angles (where knee abduction-adduction angle during static standing was used in study **II** and knee flexion-extension angle during walking was used in study **III**; the use of both is elaborated in section 6.7.2) and knee joint contact force time series (elaborated in section 6.7.1).

The musculoskeletal modeling and simulation pipelines were implemented using the OpenSim API. We used version 4.0 in study **I**, version 4.1 in study **II**, and version 4.3 in study **III**.

6.7 DESIGNING THE MODEL FOR PREDICTING KNEE JOINT LOADING

In this section, we describe the ANN-based models for predicting KJCF peaks. We introduce the response variables (i.e., variables that a prediction model estimates with its output) and the predictors (i.e., input variables to a prediction model). We also present the architecture of the utilized ANNs and how their prediction accuracy was evaluated.

6.7.1 Response variables

In studies **II** and **III**, we trained models to predict nine response variables. During musculoskeletal modeling and simulation, we solved three time series for tibiofemoral compressive KJCFs: those in the medial compartment, those in the lateral compartment, and their sum. In each of these three, we extracted the maximum value of the loading response peak (i.e., first peak), the maximum value of the terminal extension peak (i.e., second peak), and the maximum value over the full time series (i.e., the full-stance peak, which always equals one of the two aforementioned peaks). These response variables were extracted from each walking trial. Thus, we had three response variables per three time series, resulting in nine response variables in total (Figure 6.8).

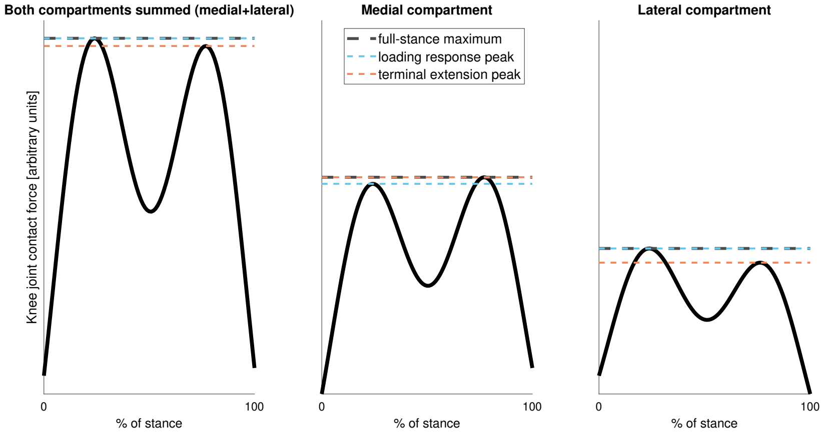

Figure 6.8: The nine response variables predicted in studies **II** and **III**. In each walking trial, the full-stance maximum equaled the loading response peak or the terminal extension peak, whichever was higher.

6.7.2 Predictor selection

After establishing the response variables, we wanted to build prediction models to estimate the values of these response variables given some input variables (i.e., predictors). For our predictor candidates, we established that the predictor must be retrievable without motion capture data. From demographic data we selected the mass, height, body mass index (BMI), age, and sex as the predictor candidates because they are easily collected, generally available in studies involving motion capture, and usually when at least one of these variables is collected, so are the others [8]. Furthermore, mass has been found to correlate with maximum knee joint contact force [173].

However, we also wanted information that describes gait. For this purpose, walking speed was chosen because it is simple to measure and has both intuitive physical connection to knee joint loading as well as a scientifically demonstrated one [76, 128, 173]. At this point, we had two measures of walking speed: (1) the mean walking speed over the stance phase and (2) the instantaneous walking speed calculated from the frame of heel strike detection and the frame just before it.

Additionally, because we wanted information that could differentiate loading between the medial and the lateral compartments of the knee, we chose knee frontal plane alignment as a predictor in study **II**; it has been shown to correlate with the loading ratio between the medial and the lateral compartment [13]. We extracted knee frontal plane alignment as the knee abduction-adduction angle of the musculoskeletal model after scaling it using marker data from a static standing trial and the inverse kinematics procedure of OpenSim. Estimates of knee frontal plane alignment using marker data have been shown to correlate with the mechanical axis of the lower extremity measured from radiographic images and KOA severity [56], which further makes frontal plane alignment from motion capture a potentially useful predictor.

In study **III**, the knee frontal plane alignment was replaced by knee flexion angle (KFA) because, unlike frontal plane alignment, it was easy to estimate using inertial measurement units and video cameras [42] (although OpenPose could be used to estimate knee frontal plane alignment from image data [129]). There, we could extract a time series instead of a single value, so we calculated the KFA predictor as the difference between the maximum of the first half and the minimum of the second half of the KFA time series during stance (Figure 6.9).

We used variance inflation factor to assess the multicollinearity of the predictors. The results guided us to drop BMI (while keeping mass and height) and instantaneous walking speed at heel strike (while keeping the mean walking speed during stance). The final predictors are shown in Table 6.1.

Predictor	Unit
mass of the subject	kilograms
height of the subject	meters
age of the subject	years
sex of the subject	binary with 1=male and 0=female
mean walking speed during stance	meters per second
knee frontal alignment (study **II**)	degrees
knee flexion angle predictor (study **III**)	degrees

Table 6.1: The final predictors after omitting some predictor candidates.

From the predictors, we constructed four predictor sets of different complexities as shown in Table 6.2. These predictor sets were later used as the input while training and evaluating ANNs for predicting KJCF peaks.

Set	Included predictors
A	mass, height, age, sex
B	mass, height, age, sex, walking speed
C	mass, height, age, sex, knee frontal alignment (study **II**) or knee flexion angle predictor (study **III**)
D	mass, height, age, sex, walking speed, knee frontal alignment (study **II**) or knee flexion angle predictor (study **III**)

Table 6.2: Predictor sets constructed from the predictors.

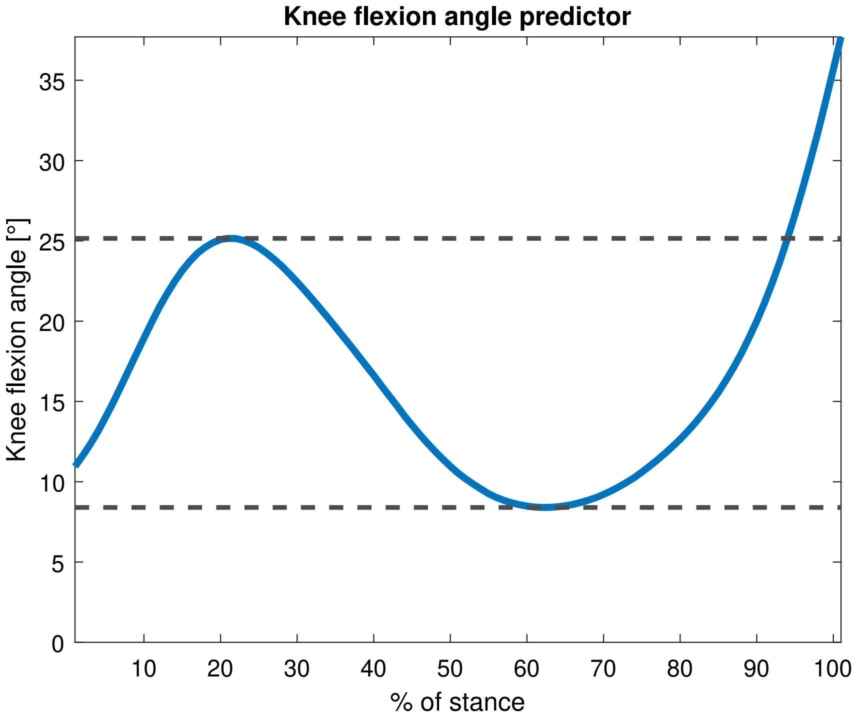

Figure 6.9: In study **III**, the knee flexion angle predictor was calculated as the difference between the maximum of the first 50% and the minimum of the last 50% of the knee flexion angle time series. The vertical distance between the dashed lines represents the knee flexion angle predictor.

6.7.3 Constructing the prediction model

Because we wanted to include the possibility of near-arbitrary nonlinear relationships between the predictors and knee joint loading peaks, we chose feedforward artificial neural networks over least squares regression. We trained an individual network for each unique combination of predictor set and response variable.

We designed the networks as fully connected feedforward shallow neural networks. Nodes in the hidden layer had a hyperbolic tangent activation function, while an identity function was used to activate the single node in the output layer. The input layer consisted of four to six nodes depending on the predictor set used (i.e., the number of nodes in the input layer was the number of predictors in the predictor set). In study **II**, the number of hidden nodes was optimized with grid search with values ranging from one to ten. Study **II** indicated that the number of nodes in the hidden layer was not important for the accuracy of predicting KJCF peaks, and therefore, in study **III** we only used one node in the hidden layer. This decision reduced the ability of the networks to approximate arbitrary nonlinear functions and its effect is discussed later in this dissertation.

The networks were trained using backpropagation (see section 4.5). In study **II**, the algorithm used a validation set to avoid overfitting; the process was implemented as a nested k-fold cross-validation [143, 174] scheme for optimizing the hyperparameters of the model and computing test error. In study **III**, we used a Bayesian regularization backpropagation algorithm that required no validation set and instead used a number of training epochs that minimized the linear combination of squared training errors and weights [134, 136]; there, a separate holdout test set was used instead of cross-validation.

The inputs underwent pre-processing that normalized them to zero mean and a range of [-1, 1]. This helped in preventing the saturation of the activation function and comparing the weights to determine the importance of the inputs (elaborated in section 6.10.3). Consequently, the output had to be post-processed to scale it up to meaningful units of physical loading. This sort of normalization is a standard practice in artificial neural networks [38].

6.7.4 Evaluating the prediction model

For comparing the predictions (i.e., outputs of the networks) against reference values, we used response values from musculoskeletal simulation and modeling corresponding to the inputs of each prediction. The reported accuracy metrics are the root mean square error (RMSE) between the predictions and the reference values, the RMSE normalized to the mean of the reference values (NRMSE), and Pearson linear correlation coefficients; these are common metrics in evaluating the performance of ANN prediction models [20, 75, 143]. The metrics presented in studies **II** and **III** have been calculated against a test set whose data points were not used during training or validation.

In study **II**, dataset overlap but no subject overlap was allowed between the training, validation, and test subsets. This practice is recommended while training generalizable models [38]. In study **III**, we further eliminated the possible dataset-related bias by using the Kuopio gait dataset as the test set, while the combined dataset was used as the training set. Thus, there should be no data leakage between the subsets.

It should be noted that although we calculated Pearson correlation coefficients, the data points were not always independent. Each subject in the data had several walking trials associated with them. Because one data point represents a single walking trial, the assumption of independence does not hold. Thus, the correlation coefficients are not comparable to other studies but instead serve to provide comparison between predicting different responses in studies **II** and **III**.

6.8 ESTIMATING PREDICTORS FROM VIDEO DATA

This section describes how walking speed and the knee flexion angle predictor were estimated from sagittal-plane video data of the walking trials in study **III**. We first used a human pose estimation (HPE) algorithm to detect keypoints in video data. After artefact correction, both walking speed and the KFA predictor could be calculated from the trajectories of the keypoints.

6.8.1 Identifying keypoints from video frames

In study **III**, we used the OpenPose HPE algorithm (introduced in section 4.6.1) because among HPE algorithms, it is relatively easy to use and has been shown to be robust in detecting keypoints during commonly performed exercises outside the motion laboratory [39,41]. OpenPose is a framework with different models that define what keypoints are identified. We used the BODY_25 model, which identifies 25 keypoints as named in Table 6.3 and demonstrated earlier in Figure 4.8.

Index	Name	Description
0	Nose	tip of the nose
1	Neck	manubrium of the sternum
2	RShoulder	right shoulder
3	RElbow	right elbow
4	RWrist	right wrist
5	LShoulder	left shoulder
6	LElbow	left elbow
7	LWrist	left wrist
8	MidHip	middle of the pelvis
9	RHip	right hip joint
10	RKnee	right knee
11	RAnkle	right ankle
12	LHip	left hip joint
13	LKnee	left knee
14	LAnkle	left ankle
15	REye	right eye
16	LEye	left eye
17	REar	right ear
18	LEar	left ear
19	LBigToe	left big toe
20	LSmallToe	left small toe
21	LHeel	left heel
22	RBigToe	right big toe
23	RSmallToe	right small toe
24	RHeel	right heel

Table 6.3: The indices, names, and approximate locations of the 25 keypoints of the BODY_25 model of OpenPose. Note that the positions are not anatomically exact because the model is trained on images labeled by amateurs instead of experts of human anatomy.

OpenPose was run on all sagittal-plane videos recorded during the walking trials in the Kuopio gait dataset. We used a resolution of 480x272 to identify the keypoints because high resolutions often erroneously detected human shapes in the background texture of the image (e.g., the wall). In an ideal case, we could extract the positions of 25 keypoints on the participant in each video frame. However, all keypoints were seldom correctly identified and we had to implement corrections to the data to mitigate the effect of artefacts.

6.8.2 Correcting artefacts in keypoint data

Common issues with OpenPose keypoints are the mistracking of body parts [24] and the flipping of paired extremities, particularly legs, between frames [42]. In the latter, OpenPose can make the left leg and the right leg keypoints switch places suddenly. In study **III**, this appeared to occur more frequently when the left and right leg overlapped in the sagittal camera view and it was difficult to discern which leg shape was in the front and which was in the back, although it could also occur at other times. When this artefact occurs just in single isolated frames, its effect can be minimized by filtering the trajectories, which mitigates the effect of a single outlier point. However, sometimes steps such as interpolation of missing data points precede the filtering and the flipping of paired extremities can make them difficult, possibly extending the issue to interpolated data points and preventing filtering from alleviating it. A case where interpolation and filtering are successfully used for artefact correction is shown in Figure 6.10.

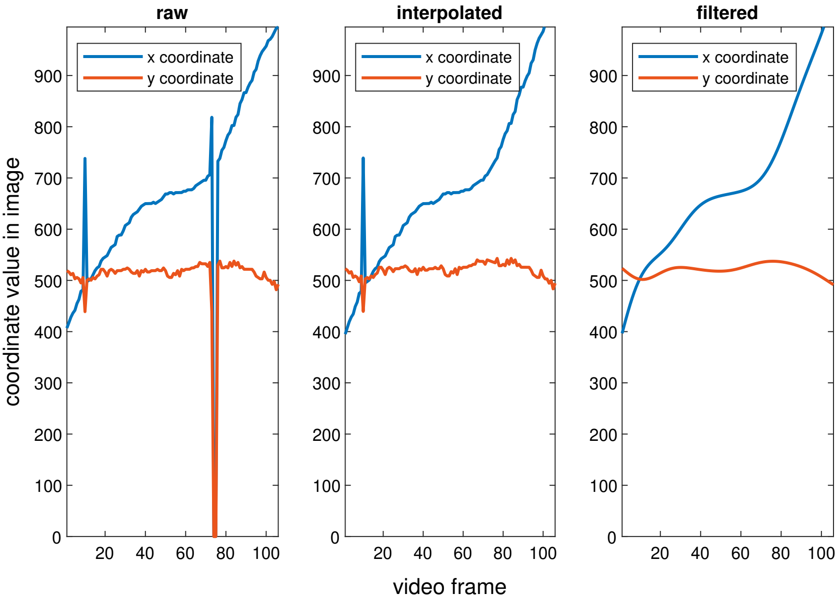

Figure 6.10: Demonstration of keypoint signal processing from a walking trial with isolated leg switching and missing data. Raw keypoint data (left) can contain artefacts from leg switching (around frame 10) and missing data where the coordinate values are set to zero (around frame 75). Single frames of missing data can be interpolated (middle). Low-pass filtering can mitigate the artefacts from single frames of leg switching (right).

However, if the flipped assignment of paired extremities occurs back and forth repeatedly in a short time period or lasts for several consecutive frames, filtering will not be as effective in removing its effect (Figure 6.11).

Additionally, as outlined in study **III**, OpenPose has trouble assigning the correct keypoints to the correct person when two people overlap in the image (Figure 6.12). This can cause burdensome work in the analysis phase.

6.8.3 Estimating walking speed using keypoint trajectories

We defined the linked height of the participants using keypoint trajectories by calculating the keypoint-to-keypoint length from the heel to keypoint to the nose keypoint. In effect, the linked height was the sum of the distance from heel keypoint

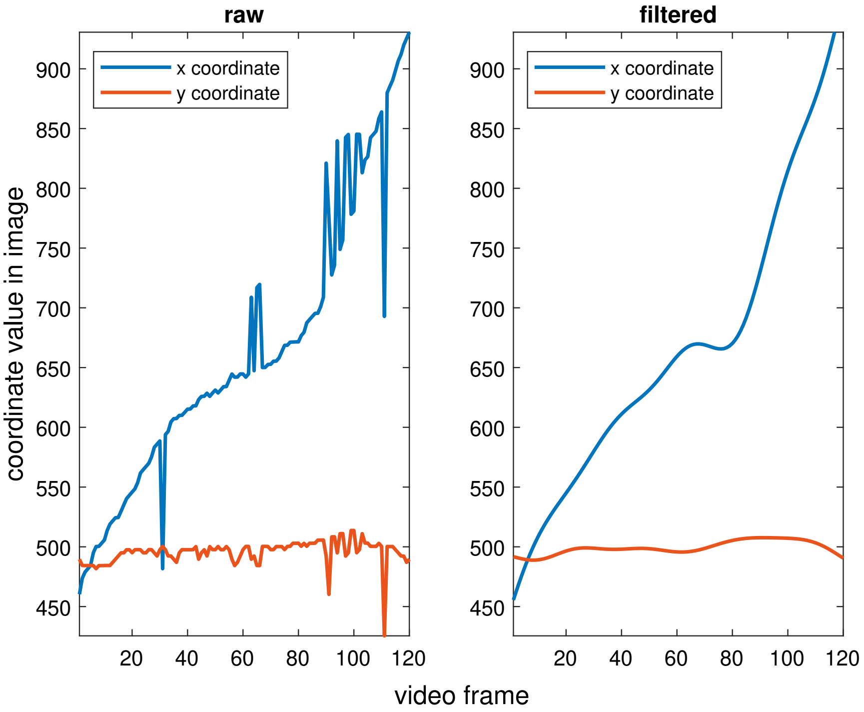

Figure 6.11: Demonstration of keypoint signal processing from a walking trial with repeated leg switching. When raw keypoint data (left) contains several consecutive artefacts from leg switching, low-pass filtering cannot mitigate their effect properly and a visible local maximum remains (right, around frame 65). Trials where this occurred were nonetheless included in the data.

to the ankle keypoint, the ankle keypoint to the knee keypoint, the knee keypoint to the hip keypoint, the hip keypoint to the neck keypoint, and the neck keypoint to the nose keypoint. Because the keypoint data was extracted from walking trials with several video frames (i.e., images), we calculated the linked height as the mean over all frames.

We then made an assumption that the relationship between the participant's linked height and the height during static standing were related according to

$$\text{linked height} = 1.2 \times \text{standing height}. \tag{6.1}$$

The factor of 1.2 between linked height and standing height was found to be reasonably accurate during initial tests. While a more accurate factor could probably have been found by methods like grid search, we opted to keep the factor because it showed good agreement between estimated walking speeds and reference walking speeds calculated from motion capture markers (e.g., Figure 6.13). The above assumption, combined with the height of the participant we measured in centimeters, allowed us to calculate an image-pixels-to-centimeters conversion factor. Using this factor, we could estimate how much any displacement in the image was in centimeters.

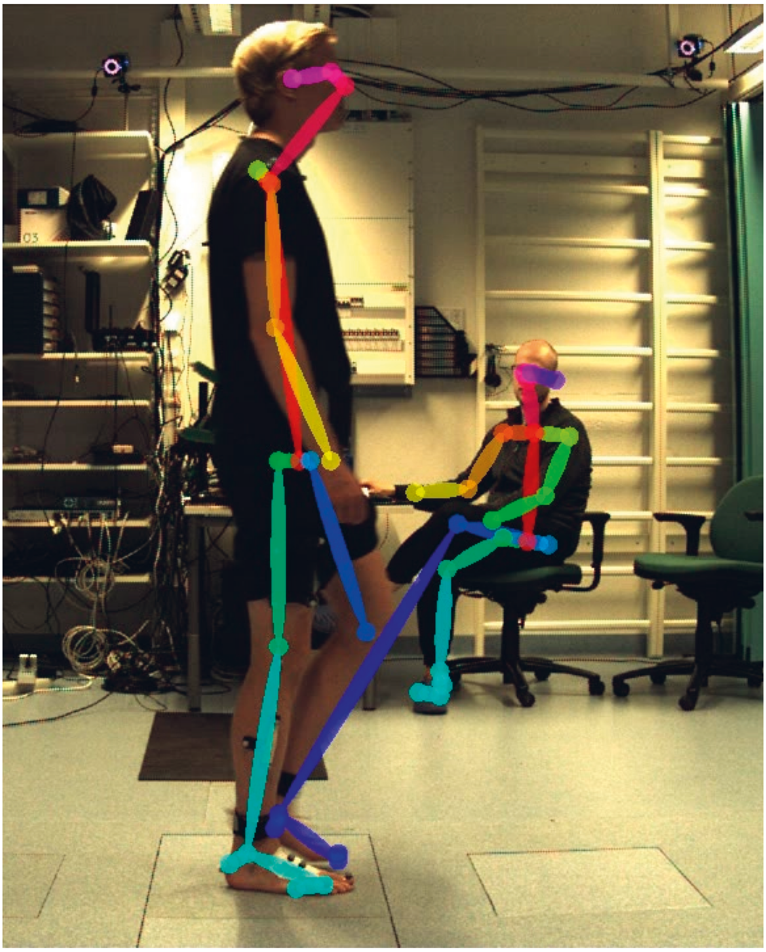

Figure 6.12: When two people overlap in the image, OpenPose sometimes fails to assign body parts to them correctly. Here, the leg assigned to the person in the background extends to the ankle and toes of the study participant in the front. Consequently, OpenPose fails to identify the study participant's left leg past the knee.

Thus, we estimated mean walking speed by tracking the displacement of the MidHip keypoint during the stance phase, using the image-pixels-to-centimeters conversion factor to convert the displacement to centimeters and dividing the result by the duration of the stance phase.

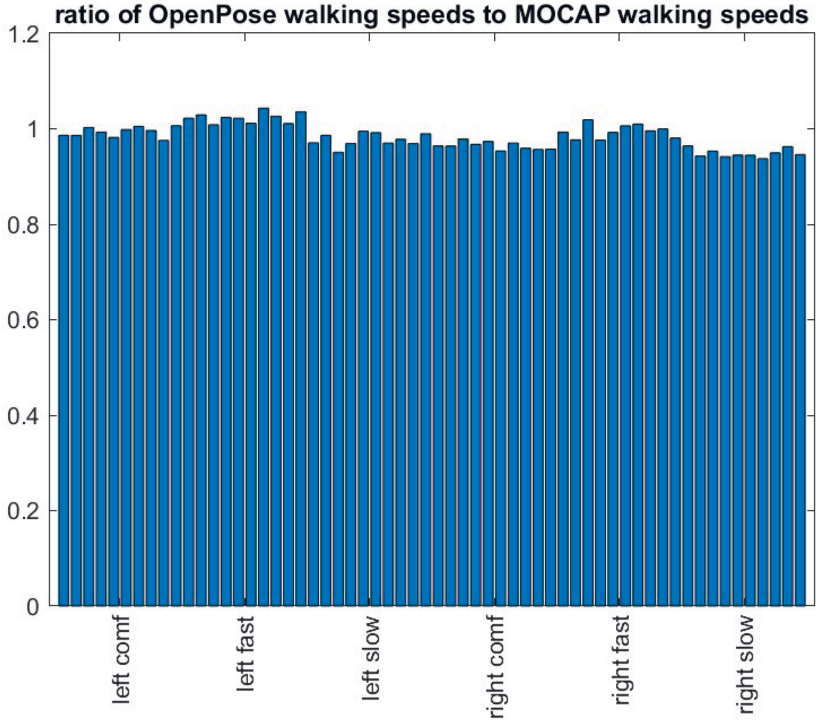

Figure 6.13: An example demonstrating the ratio between walking speeds estimated from the displacement of the MidHip keypoint and reference walking speeds calculated from the displacement of motion capture markers behind the pelvis. The figure shows data from all 60 walking trials of one participant. The labels under the bars indicate the configurations of the walking trials (see section 6.3.2); here, each configuration has 10 walking trials and a single bar represents one walking trial.

6.8.4 Estimating knee flexion angle using keypoint trajectories

We calculated the vectors from the hip keypoint to the knee keypoint and the knee keypoint to the ankle keypoint and calculated the angle between them (Figure 6.14). Because the video data was recorded in the sagittal plane and hip rotation and abduction were relatively small during walking, we assumed that the angle represented motion in the knee flexion-extension direction, i.e., the knee flexion angle. By calculating this angle in all frames of a walking trial, we could extract an estimated time series of the knee flexion angle.

6.9 ESTIMATING PREDICTORS FROM INERTIAL MEASUREMENT UNIT DATA

As stated in section 6.3, each participant was equipped with IMUs on the pelvis, thighs, shanks, and feet. For estimating walking speed and the KFA predictor, we used orientation data from IMUs on the the thighs and the shanks. Although methods utilizing machine learning exist for estimating variables from IMU data [100],

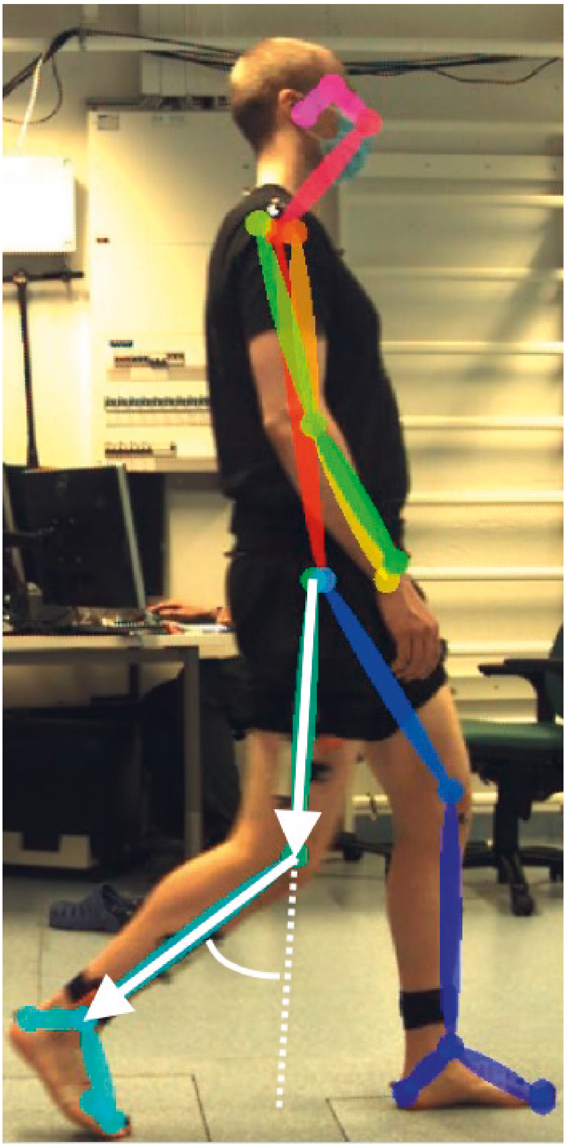

Figure 6.14: Knee flexion angle at each frame can be estimated by calculating the angle between the vector from the hip keypoint to the knee keypoint and the knee keypoint to the ankle keypoint. The vectors are drawn in white.

in study **III**, we opted to keep our methods as simple and intuitive as possible. We chose thusly because we only wanted to demonstrate how predictors could be estimated from IMU data, not to develop the new cutting-edge method for doing so.

6.9.1 Estimating walking speed using the orientation data of sensors

We modeled the thighs and shanks of the participant as rigid bodies whose lengths l are proportional to the participant's height h according to

$$\begin{cases} l_{\text{thigh}} = \frac{h}{3.7} \\ l_{\text{shank}} = \frac{h}{3.5} \,. \end{cases} \tag{6.2}$$

This proportion was optimized with grid search in a range of common proportions of limbs to body height [111]. The thighs were constrained to be linked at their proximal ends, and the distal end of the thigh was constrained to be linked to the proximal end of the shank.

We reconstructed the motion from the walking trials using this simple physical model and the orientation data from thigh and shank IMUs. For an entire walking trial, we calculated the time series of the distance between the distal ends of the shanks, which we assumed to represent the distance between the feet (Figure 6.15). Then, we identified the peaks (i.e., local maximum distances between feet) in the resulting time series; the magnitudes of these peaks represented step lengths. The time difference from one peak to the next then estimated the duration of one step, so time from one peak to the two following peaks estimated the duration of one gait cycle (i.e., stride duration). Therefore, the summed magnitudes of the two latter peaks estimated stride length. Finally, walking speed was estimated as stride length divided by stride duration.

6.9.2 Estimating knee flexion angle using orientation data of sensors

Knee flexion angle was estimated from the orientation data of thigh and shank IMUs of the dominant leg. First, we extracted the orientations of the IMUs as axis-angle representations and conducted principal component analysis on them. This analysis retrieved the rotation angles around the principal axis (i.e., the axis of greatest variance or the axis experiencing the greatest rotation around it), which we assumed to effectuate rotation in the sagittal plane during walking in a straight line. Furthermore, we assumed that at the beginning of the walking trial when the participant stood still, the knee was fully extended, which allowed us to set the initial condition of zero rotation in both IMUs. We then estimated the time series of the knee flexion angle during the stance phase as the difference between the principal rotations of the shank IMU and the thigh IMU.

6.10 POST-STUDY ANALYSES

After studies **II** and **II** were reported, we conducted additional analyses to support discussion about the developed methods. In separate analyses, we 1) investigated the possibility of using other joint angles than knee flexion angle as predictors by analyzing the linear correlation between joint angle-related parameters and knee joint loading-related parameters, 2) compared the prediction capabilities of the ANNs of study **III** against linear least squares regression, and 3) sought a way to explain the ANN-based prediction models of study **III** by analyzing their trained weights.

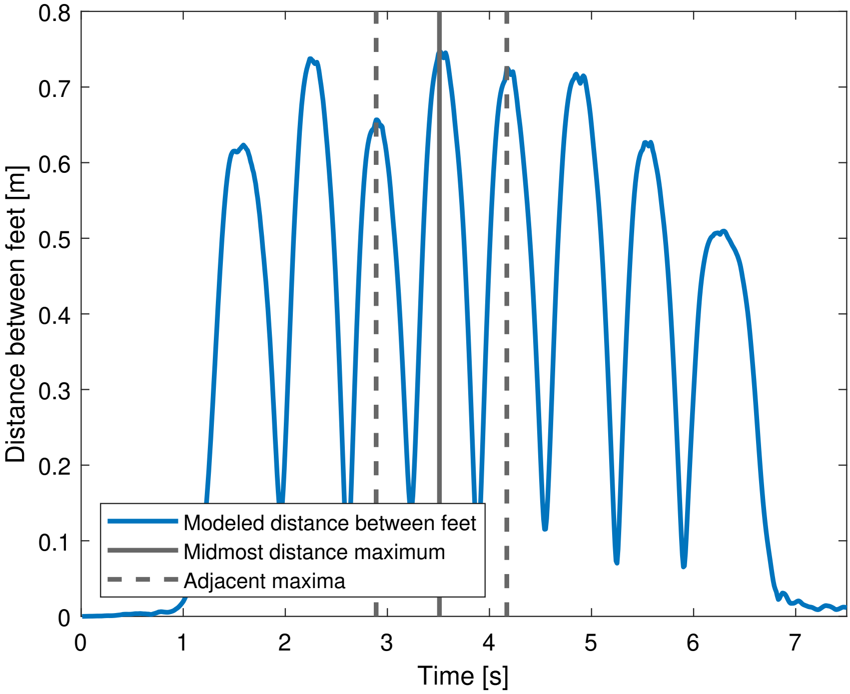

Figure 6.15: Time series of modeled distance between the feet of the participant during a walking trial, showing several gait cycles. The maxima of the curve represent the distances between the feet during gait events where the feet are maximally distant. The duration from one maximum to the next is the time taken by a single step and the value of the maximum is the step length.

6.10.1 Correlation analysis between kinematics and knee joint contact forces

After the Kuopio gait dataset was collected and analyzed, we extracted parameters of motion from the trial of all 46 analyzed subjects. The parameters were extracted from some outputs of the OpenSim pipeline, namely the time series of hip, knee, and ankle joint angles and tibiofemoral compressive knee joint contact forces. The parameters extracted from joint angles are presented in Table 6.4.

From the time series of the knee joint contact forces, the extracted parameters from the nine response variables of studies **II** and **III** are presented in Table 6.5. These parameters were extracted from the time series of the medial compartment, the time series of the lateral compartment, and the time series of both compartments summed. Some of the parameters are visualized in in Figure 6.16.

Our aim was to evaluate the linear correlation between parameters that could be easily extracted as predictors using the methods presented in studies **II** and **III** (i.e., parameters calculated from lower limb joint angles) and parameters that describe the loading of the knee joint. We hoped to highlight potential features that could be used as predictors when predicting knee joint loading in our future studies.

Joint angle time series	Extracted parameters
ankle dorsiflexion angle	(1) minimum of first half (2) maximum of second half (3) difference between maximum of 2nd half and minimum of 1st half
knee flexion angle	(4) maximum of first half (5) minimum of second half (6) difference between maximum of 1st half and minimum of 2nd half
hip adduction angle	(7) maximum of full time series
hip rotation angle	(8) minimum of full time series (9) maximum of full time series (10) difference between maximum and minimum
hip flexion angle	(11) minimum of full time series (12) maximum of full time series (13) slope of line drawn between minimum and maximum

Table 6.4: The parameters of joint angle time series used in the correlation analysis.

There were approximately 60 trials per participant at three instructed walking speeds, meaning that there were also 60 values for each parameter per participant. However, for our correlation analysis, we wanted one value for each parameter and instructed walking speed (i.e., three values of each parameter per participant). Within each participant, we extracted the parameters from the output of the OpenSim pipeline from each walking trial. We then calculated the intra-participant means of the parameters from trials where the instructed walking speed was slow, comfortable and fast, separately. This process resulted three intra-participant mean values for each parameter (i.e., one for each instructed walking speed). Therefore, we had one mean value per parameter and walking speed for each of the 46 analyzed participants. These results allowed us to evaluate the correlation coefficients between the parameters for three different instructed walking speeds and a sample of 46 independent observations.

For each unique permutation of joint angle parameter and knee joint loading parameter, we calculated the linear Pearson correlation coefficient and the corresponding probability of observing the correlation assuming there is no correlation (i.e., p-value). We excluded correlation coefficients where the corresponding p-value was higher than 0.05 or where the absolute value of the correlation coefficient was less than 0.6.

6.10.2 Multiple linear regression of knee joint loading peaks

We observed in study **III** that a shallow feedforward neural network with just one node in the hidden layer is sufficient to estimate KJCF peaks with accuracy that is comparable to that of study **II**, where the hidden layer had several nodes. Using only one hidden node in study **III** was motivated by study **II**, where the hyperparameter optimization algorithm couldn't find a conclusive number of hidden nodes for best prediction accuracy. In this case, it's possible that the relationship between the

Loading parameter	Description
notch	the minimum value between the first (loading response) and the second (terminal extension) peak
loading response ascent	the ascending slope before the loading response peak
loading response descent	the descending slope between the loading response peak and the notch
terminal extension ascent	the ascending slope between the notch and the terminal extension peak
terminal extension descent	the descending slope after the terminal extension peak
area under curve	the total area under the KJCF curve
loading response area	the area of the loading response peak above a horizontal line at the height of the notch
terminal extension area	the area of the terminal extension peak above a horizontal line at the height of the notch
width at notch	the horizontal distance at the height of the notch between the curve during the ascent before the loading response peak and the curve during the descent after the terminal extension peak
peak separation	the horizontal distance between the loading response peak and the terminal extension peak

Table 6.5: The parameters of knee joint contact force time series used in the correlation analysis.

predictors and the output of the networks is so simple that using many hidden nodes will needlessly complicate the network and possibly cause overfitting. If so, we are faced with the question: is the approximation of a nonlinear function necessary at all when predicting KJCF peaks, or could multiple linear regression suffice?

To explore this question, we fitted multiple linear regression models for predicting KJCF peaks using least squares first-order polynomial fitting. Note that while in study **III** we used the data of knee osteoarthritis patients to pre-train the model before continuing training with the remaining data, here we fitted the parameters of multiple linear regression using the entire combined dataset. This difference complicates comparability against trained ANN models slightly because the multiple linear regression likely emphasizes fitting to high-loading data points relatively more than the ANNs did in study **III**.

6.10.3 Nonlinear effect of predictors on knee joint loading peaks

In study **III**, the neural network architecture employed a single hidden layer with just one node. While this reduces the function approximation capability of the network (see section 8.4.4), it also reduces the "black box" characteristics of the model because we can evaluate how different predictors affect the prediction by looking at the weights of the network. Because there is just one node containing a nonlinear activation function (the singular node in the hidden layer) and the activation function is strictly increasing (as the hyperbolic tangent function is), the weight con-

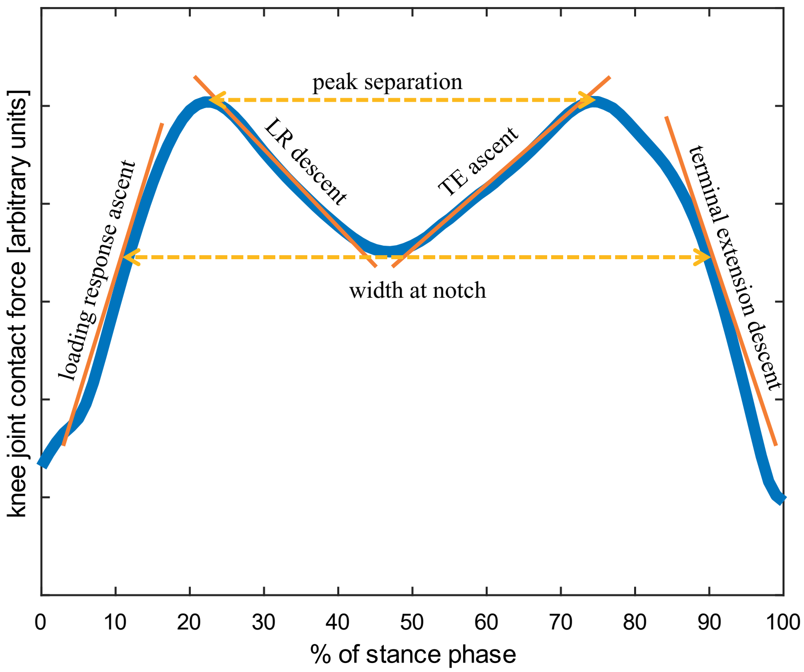

Figure 6.16: Some parameters that were extracted from the time series of knee joint contact forces. Orange dashed lines show the horizontal peak separation and width at notch. Orange solid lines represent the slopes of loading response ascent, loading response descent, terminal extension ascent, and terminal extension descent.

necting each input (i.e., predictor value) to that node describes if that predictor has a positive or a negative correlation with the prediction.

As this train of thought may be difficult to grasp, we will now approach it through the steps involved in a value passing through a model trained in study **III**. The value of a single predictor, as it is given as input to the network, undergoes the following steps: pre-processing to scale it between -1 and 1, multiplication with its weight connecting to the hidden node, activation through the activation function in hidden node as part of the linear combination of other inputs and their weights, multiplication with the weight between the single hidden node and the single output node as part of the output of the activation function, and finally, post-processing to scale the prediction to a physically sensible range (Figure 6.17).

Now imagine we pass two different values of the same predictor, e.g., walking speed, through the network while keeping the values of the other predictors constant. We denote these values by x_1 and x_2 and assume that the inequality $x_1 < x_2$ applies.

The pre-processing and post-processing steps simply scale (by a positive value) and apply an offset to the inputs, meaning that the inequality still applies after the inputs have been put through those steps (note that in the case of post-processing, the original inputs have been transformed to different values). Similarly, between

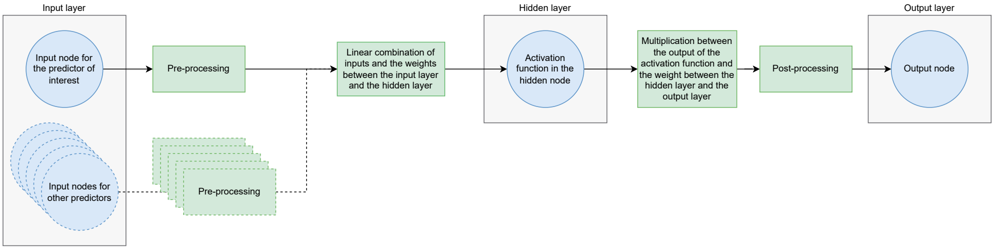

Figure 6.17: As a single input value propagates through a feedforward neural network trained in study **III**, it undergoes pre-processing and partakes in the linear combination of inputs and the weights between the input layer and the hidden layer. The linear combination undergoes activation in the hidden node, the output of the activation function is multiplied by a weight and finally put through a linear activation function (i.e., identity function) in the output node.

the input layer and the hidden layer, the values are multiplied by the same weight $w_{\text{input to hidden}}$, meaning that the inequality applies if the weight is positive and flips if the weight is negative. We can assume the weight is nonzero because otherwise the predictor has no effect on the prediction.

Furthermore, because the activation function $f(x)$ in the hidden node is strictly increasing, for any input values a_1 and a_2 for which $a_1 < a_2$, we have $f(a_1) < f(a_2)$. In plain words, if the input to the activation function (and by extension, the input to the hidden node) increases, the output of the activation function must increase. If the weight between the hidden node and the output node $w_{\text{hidden to output}}$ is nonzero and positive, the output of the network (i.e., the prediction) also increases; if the weight is nonzero and negative, the output of the network decreases.

Now, we have asserted simple rules that define if an input to the network is positively or negatively correlated with the input to the hidden layer (when other predictors are kept constant), and if the input to the hidden node is positively or negatively correlated with the prediction. In other words, a predictor is positively correlated with the prediction if

$$w_{\text{input to hidden}} w_{\text{hidden to output}} > 0 \tag{6.3}$$

and negatively correlated otherwise (assuming nonzero weights).

Because we can read the weights of the network, we can use equation 6.3 to determine if a predictor is positively or negatively nonlinearly correlated with the prediction in any of the networks trained in study **III**. Because of the effect of other predictors, quantifying the strength of the correlation is difficult. However, assuming the pre-processing step scales the predictors to the same scale, we can compare the magnitudes of $w_{\text{input to hidden}}$ between different predictors to assess how strongly a predictor affects the prediction compared to other predictors.

Therefore, for the trained ANN corresponding to each of the nine response variables, we calculated the importance of each predictor (mass, height, age, sex, walking speed, KFA predictor) of study **III** according to

$$\tau = w_{\text{input to hidden}} \operatorname{sgn}(w_{\text{hidden to output}}), \tag{6.4}$$

where τ is the importance of a predictor (i.e., how much changing it affects the KJCF peak prediction), $w_{\text{input to hidden}}$ is the weight between input and hidden layer for that predictor, sgn is the signum function returning the sign of the input, and $w_{\text{hidden to output}}$ is the weight between the hidden layer and the output layer.

We then normalized the importances under each response variable by dividing them by the maximum absolute magnitude of importances of all predictors for that response variable to easily compare importances between networks trained for different response variables.

7 Results

7.1 PERFORMANCE OF COMPUTING REAL-TIME INVERSE KINEMATICS

In study **I**, we showed that full-body inverse kinematics can be calculated in real-time. The throughput (solved kinematics frames per unit time) of the computation could be increased by increasing the number of processing threads (study **I**, Figure 3) and the execution time of computing a single frame of kinematics during gait was nearly instantaneous (<60 ms, Figure 2 in study **I**). The computational performance of the solution was affected by computer hardware, the complexity of the utilized musculoskeletal model, the number of IMUs used to measure orientation data, and the measured movement type. The results applied even with a laptop computer, meaning that IMUs and a laptop for computation are a suitable combination for portable measurement and real-time analysis of joint angles during gait.

7.2 MUSCULOSKELETAL SIMULATION-ESTIMATED KNEE JOINT CONTACT FORCE PEAKS

Representative characteristics of knee joint contact force peaks obtained using musculoskeletal modeling and simulation in study **III** are illustrated in Figure 7.1. During self-selected comfortable walking speeds, the participants of study **III** walked at 1.04 m/s to 1.78 m/s (mean $\pm$ standard deviation was 1.35 ± 0.17 m/s) and the maxima of their full-stance tibiofemoral compressive knee joint contact force in both compartments summed varied between 2.82 BW and 4.92 BW (3.59 ± 0.53 BW). During slow walking speeds, the participants walked at 0.72 m/s to 1.66 m/s (1.11 ± 0.18 m/s) with full-stance maximum KJCF between 2.76 BW and 4.68 BW (3.34 ± 0.44 BW); during fast walking speeds, they walked at 1.27 m/s to 2.33 m/s (1.55 ± 0.23 m/s) with full-stance maximum KJCF between 2.86 BW and 7.18 BW (4.09 ± 0.88 BW). The values are calculated over 46 analyzed participants from their intra-participant means.

7.3 ESTIMATING THE WALKING SPEED AND KNEE FLEXION ANGLE PREDICTORS

In study **III**, walking speed was estimated with similar accuracy using IMU and VC data, although IMU estimates were more precise as indicated by the smaller standard deviation of their intra-participant RMSE over different participants (Table 7.1). However, the KFA predictor could be estimated from IMU data more accurately than from VC data as indicated by the smaller mean of their intra-participant RMSE over different participants (Table 7.1). Comparing to the inter-participant mean reference values of walking speed (133.84 ± 17.71 cm/s) and the KFA predictor ($16.26 \pm 4.09°$), the relative root mean square error of estimating walking speed is below 5% for both modalities, and the relative root mean square error of estimating the KFA predictor is approximately 23% for IMUs and 32% for VCs.

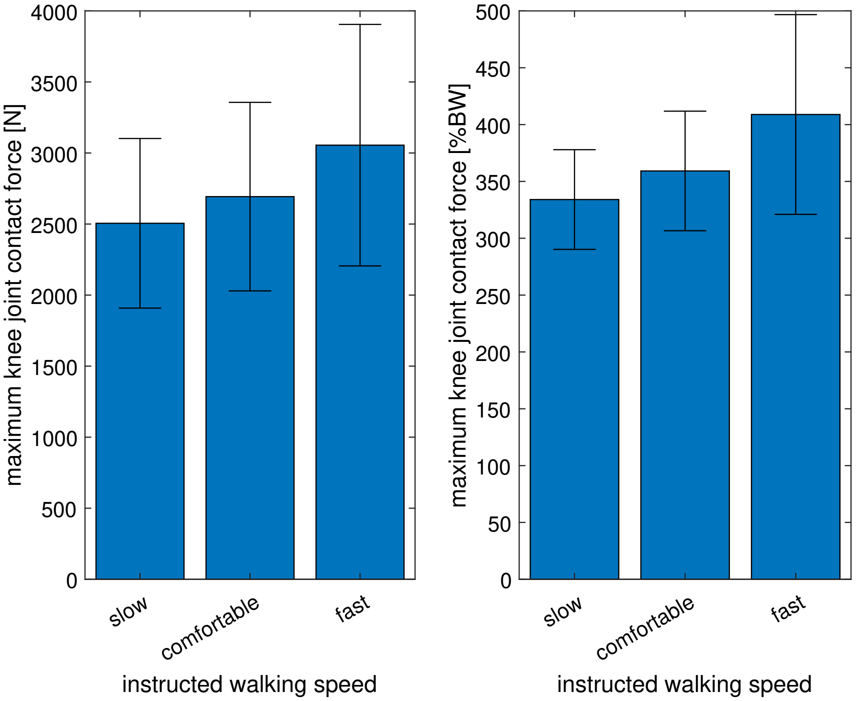

Figure 7.1: The inter-participant mean (bar height) and standard deviation (error bars) of intra-participant knee joint contact force maxima (right) and bodyweight-normalized knee joint contact force maxima (right) when the instructed walking speed was slow, comfortable, and fast. The knee joint contact forces are the outputs of the musculoskeletal modeling and simulation workflow.

	IMU		**VC**	
Predictor	**RMSE ($\mu \pm \sigma$)**	**R ($\mu \pm \sigma$)**	**RMSE ($\mu \pm \sigma$)**	**R ($\mu \pm \sigma$**
walking speed [cm/s]	6.52 ± 4.42	0.95 ± 0.08	6.29 ± 9.16	0.96 ± 0.10
KFA predictor [°]	3.75 ± 1.54	0.91 ± 0.11	5.24 ± 1.83	0.58 ± 0.24

Table 7.1: Inter-participant means (μ) and standard deviations (σ) of intra-participant root mean square errors (RMSE) and Pearson correlation coefficients (R) of walking speed and knee flexion angle (KFA) predictor estimates retrieved from inertial measurement unit (IMU) and video camera (VC) data. The values are computed by comparing against reference walking speed and KFA predictors calculated using MOCAP data.

The knee flexion angle time series, from which the KFA predictor was extracted, was estimated with an overall mean RMSE of 4.1° and 5.1° with IMU and VC data, respectively. IMU-based estimates of knee flexion angle were less accurate during the minimum of the time series, while VC-based estimates had relatively high error throughout the stance phase (Figure 7.2). There is a noticeable phase mismatch between the reference and VC-based estimated knee flexion time series.

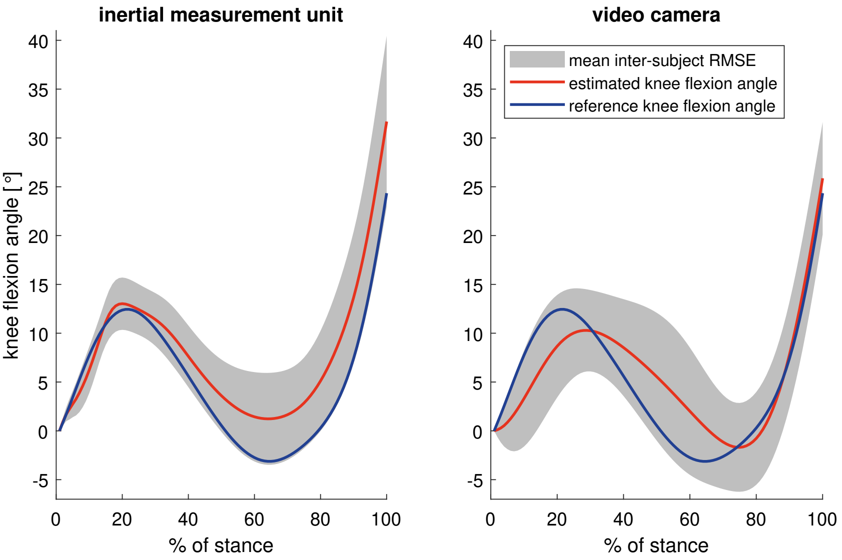

Figure 7.2: Mean inter-subject knee flexion angle time series estimated from inertial measurement unit (left) or video camera (right) data compared with the reference time series (from motion capture data and musculoskeletal modeling and simulation). The shaded area depicts the mean inter-subject root mean square error per data point (all time series were interpolated to 101 data points).

7.4 PREDICTING KNEE JOINT CONTACT FORCE PEAKS

In studies **II** and **III**, we showed that artificial neural networks with simple inputs can be used to predict knee joint loading peaks during the stance phase of gait. Furthermore, in study **III**, we demonstrated that motion capture data is not necessary to obtain the inputs for such neural network models after their training, and therefore, portable modalities can be used to estimate knee joint loading peaks outside the motion laboratory.

The accuracy of predicting KJCF peaks depended on the predictor set (Figure 7.3). The predictor set with the most predictors (mass, height, age, sex, walking speed, and either knee frontal plane alignment or the knee flexion angle predictor) resulted in the smallest prediction error (root mean square error normalized to the mean of the reference values in the test set) for most response variables. Furthermore, using only demographic variables and walking speed resulted in prediction error similar to when using all predictors.

Using the largest predictor set, the KJCF peaks for different response variables were predicted with mostly comparable error between studies **II** and **III** (Figure 7.4).

In addition to the choice of the predictor set, the response variable (i.e., which loading peak was predicted) affected the prediction error (Figures 7.3). Maxima over the full stance phase and terminal extension peaks were predicted with smaller error than loading response peaks were. Response variables in the lateral compartment were predicted with much greater error compared to their counterparts in the medial compartment or the sum of both compartments.

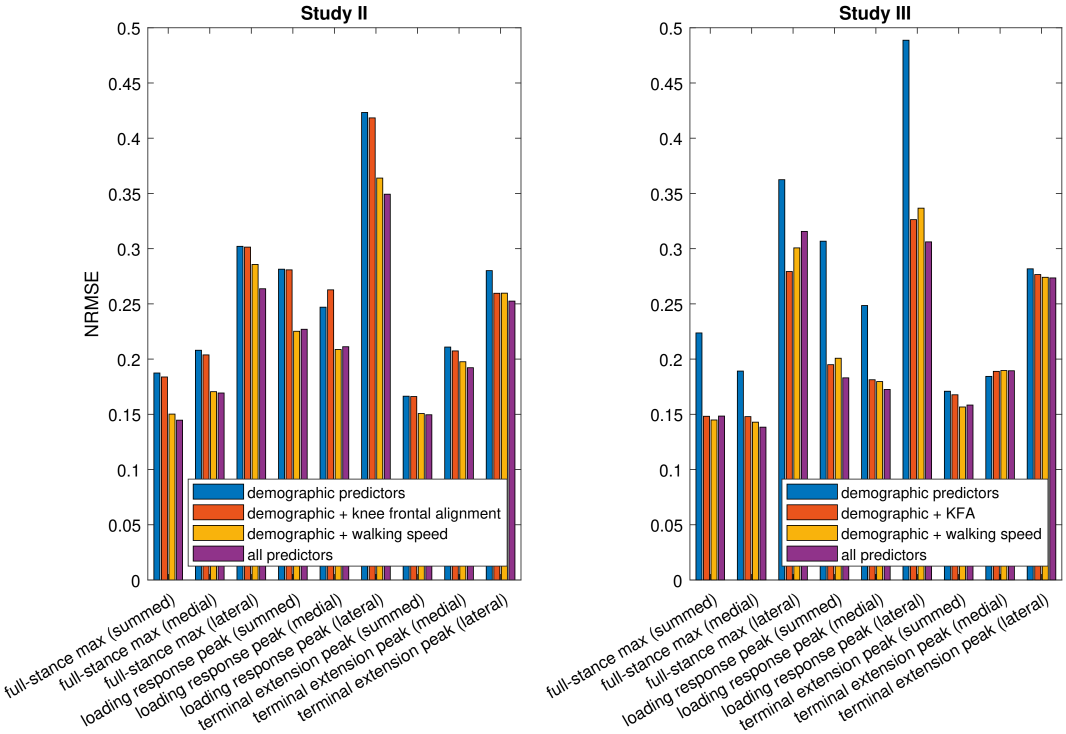

Figure 7.3: The effect of predictor set on the normalized errors of knee joint contact force peaks in studies **II** (left) and **III** (right). The predictor set comprised mass, height, age, sex, walking speed, and either knee frontal alignment (study **II**) or the knee flexion angle predictor (study **III**). Note that "knee frontal alignment" in the left figure is not the same as "KFA" in the right figure, where "KFA" denotes the knee flexion angle predictor.

Finally, to see if increasing the number of nodes from one to two increased prediction accuracy, we compared error for the full predictor set (where assumedly the most complicated relationship between predictors and output exists) for all nine response variables. The results are not presented in detail because prediction accuracy varied very little between ANNs with one and two nodes. Thus, because increasing the number of nodes had no marked effect on prediction accuracy, a single node is likely enough to capture the relationship between the predictors and the output.

7.4.1 Nonlinear effect of predictors on knee joint contact force peaks

Mass was always the most important predictor for predicting the KJCF peaks, no matter the response variable (Table 7.2); hence, mass is not explicitly presented in the table and its value was always 1. It was followed by the KFA predictor and walking speed, with the former being slightly more important than the latter. Height had a small importance, but its importance was still greater than that of sex and lastly, age. For most response variables, only height and age had negative correlation with the

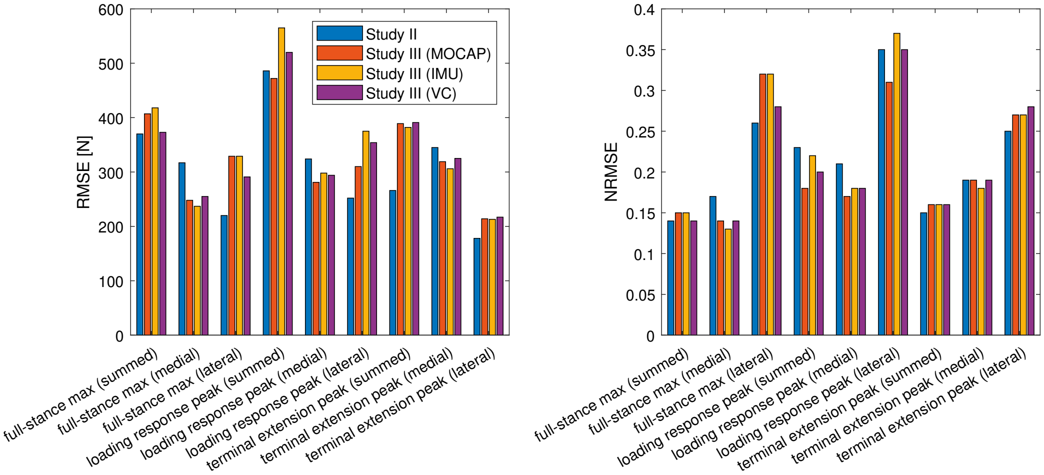

Figure 7.4: Comparison of the prediction errors of knee joint contact force peaks between studies **II** and **III**. The predictor set comprised mass, height, age, sex, walking speed, and either knee frontal alignment (study **II**) or the knee flexion angle predictor (study **III**). The knee flexion angle predictor and walking speed in study **III** were estimated from motion capture (MOCAP), inertial measurement unit (IMU), or video camera (VC) data. Root mean square errors (RMSE) in study **III** are greater than those in study **II** for summed and lateral force peaks, but smaller for medial force peaks.

prediction; otherwise the predictors correlated positively with the predicted KJCF peaks (except for the KFA predictor for lateral terminal extension peaks, where the KFA predictor hardly mattered anyway).

7.5 LINEAR CORRELATION BETWEEN KINEMATICS AND KNEE JOINT CONTACT FORCE PEAKS

The results of the correlation analysis confirmed our choice of using knee flexion angle data to construct a predictor. However, in addition to the KFA predictor that we selected as the magnitude between the maximum of the first half and the minimum of the second half of the time series, the maximum of the time series had comparably high correlation with variables related to knee joint loading. Correlation coefficients between knee flexion-related variables and knee joint loading were generally higher than those between other joint angles and knee joint loading. The knee flexion angle-related correlation coefficients that made it past the exclusion criteria ($p<0.05$ and $R>0.6$) were usually tested against loading response-related knee joint loading variables.

At comfortable walking speed, the maximum of hip flexion angle correlated positively with loading response ascent in the medial and summed compartments ($R=0.71$ and $R=0.68$, respectively), and the maximum of hip internal rotation angle correlated negatively with the lateral terminal extension peak ($R=-0.60$).

At slow walking speed, the slope of hip flexion angle correlated negatively with

	height	**age**	**sex**	**walking speed**	**KFA predictor**
full-stance maximum (summed)	-0.12	-0.04	0.08	0.32	0.39
full-stance maximum (medial)	-0.12	-0.06	0.08	0.29	0.38
full-stance maximum (lateral)	-0.19	-0.02	0.11	0.35	0.52
loading response peak (summed)	-0.10	-0.05	0.06	0.40	0.70
loading response peak (medial)	-0.12	-0.05	0.03	0.34	0.50
loading response peak (lateral)	-0.11	-0.06	0.11	0.47	0.91
terminal extension peak (summed)	-0.13	-0.03	0.07	0.24	0.21
terminal extension peak (medial)	-0.13	-0.07	0.08	0.23	0.36
terminal extension peak (lateral)	-0.16	0.02	0.05	0.27	-0.02

Table 7.2: Normalized importances of predictors (top row) for trained neural networks for predicting various KJCF peaks (leftmost column). Because mass was always the most important predictor, its normalized value was always 1 and it is not presented in the table.

the summed loading response ascent and lateral loading response area (R=-0.61 and R=-0.62, respectively), the maximum of hip internal rotation angle correlated positively with peak separation and width at notch in the lateral compartment (R=0.68 and R=0.63, respectively) and negatively with the lateral terminal extension peak (R=-0.62), and the minimum of hip internal rotation angle correlated positively with peak separation and width at notch in the lateral compartment (R=0.63 and R=0.61).

At fast walking speed, the maximum of the hip flexion angle correlated with summed loading response descent (R=-0.62), summed loading response ascent (R=0.61), summed loading response peak (R=0.60), medial loading response descent (R=-0.60), and medial loading response ascent (R=0.63), and the slope of hip flexion angle correlated negatively with summed loading response area (R=-0.61) and lateral loading response area (R=-0.67).

Ankle dorsiflexion angles did not correlate significantly with |R|>0.6 at any walking speed.

7.6 MULTIPLE LINEAR REGRESSION OF KNEE JOINT LOADING PEAKS

Predicting KJCF peaks with multiple linear regression (Figure 7.5) results in the same observations as predicting KJCF peaks with ANNs, i.e., **1)** the simplest predictor set results in the highest prediction error, **2)** for most response variables the addition of walking speed in the predictor set is more important than the addition of the KFA predictor, and **3)** the addition of both walking speed and the KFA predictor in the predictor set does not further reduce prediction error.

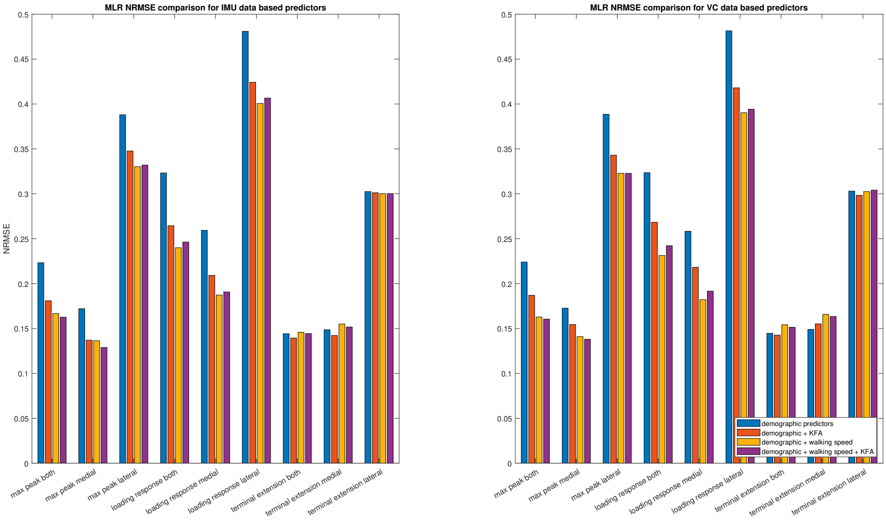

Figure 7.5: NRMSE for different predictor sets when predicting KJCF peaks with multiple linear regression. The parameters are fitted on the data from study **II** and NRMSE calculated against the "ground truth" peaks from the dataset we collected in study **III**.

Figure 7.6 shows that multiple linear regression has smaller NRMSE in predicting KJCF peaks than the ANN models of study **III** has in some cases. For instance, when the predictor set contains only demographic variables, multiple linear regression is more accurate for predicting the peaks of most response variables. However, the ANNs outperform multiple linear regression when the predictor set contains more than just the demographic predictors.

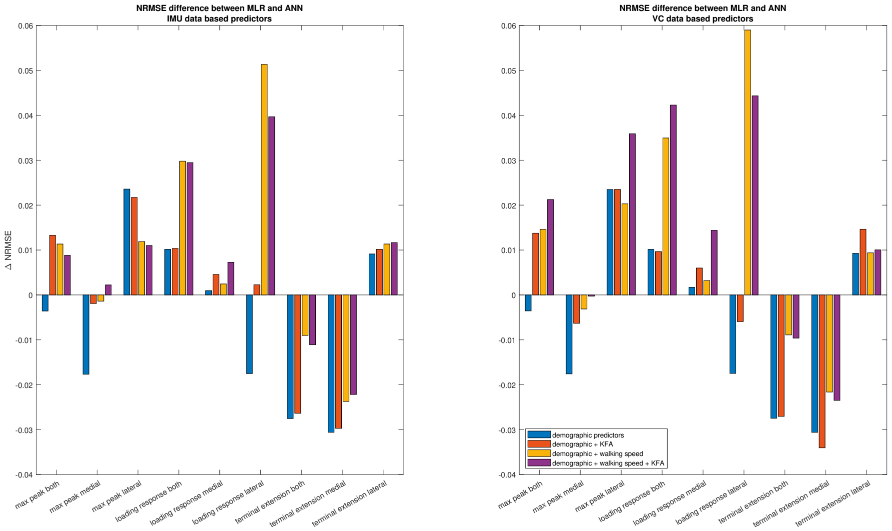

Figure 7.6: Difference in the NRMSE of predicting KJCF peaks between multiple linear regression and ANN regression. The ANNs are from study **III**. Positive values indicate that multiple linear regression had higher NRMSE than an ANN.

8 Discussion

8.1 REAL-TIME INVERSE KINEMATICS

Our solution in study **I** computes inverse kinematics utilizing an user-defined number of threads within real-time criteria of delay between measurement and output of the computations [175]. However, it utilizes a now-outdated version of OpenSim API. Modern versions of OpenSim support a more efficient way to feed data to the IK-solving algorithm (which was not available during development) than the one used by OpenSimLive. Therefore, the performance could be further improved.

Furthermore, while we were reporting our results, two other research groups independently published their own real-time IK solutions utilizing IMU data [29,94]. Unlike our solution, which only enables analysis of inverse kinematics, the solution by Stanev et al. further enables the real-time calculation of inverse dynamics and even muscle forces by estimating ground reaction forces and moments using the solved kinematics [29]. Considering these observations and the fact that our solution is incompatible with the current version of OpenSim, it is unlikely that it will be widely used by others. Nonetheless, it provides any interested parties an example of how to implement real-time biomechanical analysis using the OpenSim API by utilizing an user-defined number of processing threads.

8.2 VALIDITY OF ESTIMATED KNEE JOINT CONTACT FORCE

We can compare the knee joint contact forces (KJCFs) obtained from our musculoskeletal modeling and simulation pipeline in study **III** to those in existing literature to assess the validity of the estimated forces. For self-selected comfortable walking speed (inter-participant mean 1.35 m/s), the summed knee joint contact force maxima estimated with our musculoskeletal modeling and simulation pipeline, the tibiofemoral compressive force peaks varied roughly between 2000 and 3500 Newtons, or 3–4.5 times bodyweight (Figure 7.1).

In literature, Winby et al. (2009) estimated KCJFs for 11 healthy subjects (44 ± 6.9 years of age) and found peak tibiofemoral compressive forces of 3.9 BW (range 3.2–4.9 BW) [3]. These peaks occurred during early stance, i.e., loading response peaks were generally higher than terminal extension peaks. Furthermore, Winby et al. noted that the estimates probably underestimate the forces because ligaments (which have a compressive effect on the knee) were not modeled.

Our results conform to those of Winby et al. [3], although the values of peak KJCFs in study **III** are slightly smaller (3–4.5 BW versus 3.2–4.9 BW). Winby et al. did not report the walking speeds, but considering the participants of study **III** are much younger than those of their study (age 28.7 ± 5.9 versus 44 ± 6.9), age-related effects (in addition to differences in modeling and data collection) may affect the estimated contact forces.

In their review of existing literature, D'Lima et al. (2012) summarized that maximum forces transmitted across the knee joint during normal walking vary between two and three times body weight [5]. However, because of the diversity of computational methods to estimate these forces, the estimates of tibiofemoral compressive forces vary much more: estimated forces from musculoskeletal simulations varied between 1.7 and 7 times body weight [5]. Similarly, another review by Fregly et al. (2012) (with several overlapping authors as in the review by D'Lima et al., so the two reviews cannot be considered entirely independent) summarized that maximum forces ranged from 1.8 to 3.0 BW, while estimates ranged from 1.8 to 8.1 BW during walking [6]. All in all, the outputs of the musculoskeletal modeling and simulation workflow in study **III** fit snugly within existing literature. This observation is no surprise because we already did some validation in study **II** (elaborated in its manuscript and supplement). Therefore, it appears that the minor modifications we did to the musculoskeletal modeling and simulation workflow in study **III** did not meaningfully change the estimated knee joint contact forces between the studies.

8.3 ESTIMATING PREDICTORS FROM PORTABLE MODALITIES

The methods we developed provide a way to estimate walking speed and the knee flexion angle predictor in limited portable use cases (the assumptions of each method are elaborated in the subsections below). Existing literature contains many examples of estimating joint angles or walking speed from video camera or IMU data [32], or even a combination of both [176, 177]. Instead of developing new state-of-the-art methods to compete with existing solutions, our aim was rather to develop simple, modular, and portable methods that provide usable estimates of the predictors and demonstrate their usability in predicting knee joint contact force peaks.

8.3.1 Walking speed from IMU data

Using a physical model-based approach to estimating walking speed from IMU data using one IMU on the shank, Li et al. (2010) estimated walking speed with an RMSE of 4% during overground walking [101]. Mariani et al. (2010) used two IMUs to estimate walking speed with a mean absolute error of 1.4 ± 5.6 cm per second, corresponding to a relative error of 1.5 ± 5.8 % and a correlation coefficient of 0.971 [102]. In study **III**, our walking speed estimates from IMU data had an RMSE of 6.5 ± 4.4 cm/s and a correlation coefficient of R=0.95, corresponding to a relative error of 5%.

8.3.2 Walking speed from video camera data

Estimating walking speed from video camera data was the most straightforward for predictor estimation in study **III**. The MidHip keypoint was easy to track as it couldn't flip between extremities and its trajectory was mostly straight at approximately uniform speed throughout the stance phase. Because we had positioned the video camera such that the participant was in the center of the view during the stance phase and the movement was recorded sagittally, we could get an accurate estimate of walking speed simply by associating pixel displacement in the image to real-world displacement.

The usability of our method assumes that the motion occurs in a straight line and that the camera can be placed to record it sagittally. When former fails to hold, our method is no longer usable. When the latter does not apply, the estimated walking speed should be corrected by multiplying it with a factor corresponding to the angle difference from the sagittal plane; further correction may also be necessary to associate pixel displacement to real-world displacement if the participant's distance from the camera changes significantly during the movement or if the participant moves close to the edge of the camera view and lens distortions become noticeable.

Even if OpenPose couldn't identify the MidHip keypoint (we never encountered such an issue), other keypoints could be used to estimate walking speed. For instance, torso and head keypoints experience minimal non-uniform trajectory changes during the stance phase. Alternatively, even if we had to use keypoints from lower or upper extremities, whose trajectories change non-uniformly during the stance phase, an average position calculated from the keypoint pairs of paired extremities could perhaps be used to estimate the movement of the participant's center of mass with reasonable accuracy. Therefore, although our aim wasn't to develop a new state-of-the-art method for walking speed estimation from video data, our method may be usable also in different conditions with only minor modifications (e.g., using other keypoints when the MidHip keypoint is not available).

A video-based approach like ours may be useful in mitigating measurer-specific accuracy and repeatibility errors present in stopwatch measurements of walking speed (which can be considered an alternative, even more portable way to estimate walking speed) [178]. With a stopwatch, the measurer must react to the subject walking over the start and finish lines or some points in space representing those so that the walking speed can be calculated from the known distance between those points and the duration measured with a stop watch; a human measurer cannot exactly start and stop the stopwatch at the right moment, causing the error [178,179]. For instance, during a 4-meter walking test for healthy old adults, walking speeds estimated with a manual stopwatch were found to be significantly higher than walking speeds measured with an automatic system when the participants had a moving start but not when they had a standing start; however, even then the error was less than 5% of the mean walking speed [179]. Because our method associates image pixels to real world distance from video data, which also contains time stamps (or we can exactly define them by knowing the number of frames and the frequency of the camera), there is no human measurer-induced error. However, a stopwatch could still be used effectively to time a longer distance of walking (e.g., 100 meters), in which case the human error would become very small in comparison to the duration of the measurement and the resulting mean walking speed would be more reliable than a measure from just one gait cycle; such long walking measurements are difficult to capture with a camera, which would either have to move with the subject or be be placed far from the subject (decreasing the resolution keypoint tracking) to capture the entire measurement.

8.3.3 Knee flexion angle from IMU data

Our method assumed that the highest variation in IMU rotation occurred along the knee flexion-extension plane of motion for both thigh and shank IMUs. While this assumption may mostly apply for healthy gait while walking at a straight line, it may not apply for pathologically affected gait [118] or when the walking measurement does not occur in a straight line. We observed the latter during some walking trials, where the participants forgot to stop at the end of the trial and instead started turning to walk back to their starting position in preparation for the following walking trial. In such a case, the rotation around the vertical axis measured by the IMUs had far more variation than the rotation on the sagittal plane of walking. As a result, our PCA-utilizing algorithm used rotation on the horizontal plane for estimating knee flexion angle, which led to nonsensical estimates. Wherever a method utilizing the assumption of most rotational variance in the sagittal plane is used, participants forgetting to follow instructions will lead to having to repeat some walking trials. Therefore, our method is not ideal for many real use cases. However, it still demonstrates that planar knee flexion information can be estimated with a limited number of IMUs and used for predicting KJCF peaks. Finally, analyzing the relative rotation of the two IMUs instead of analyzing the rotations of individuals IMUs could enable analysis of knee flexion angles despite the participants turning around; however, such analysis would assume that the IMU coordinate axes are identically aligned, which is an assumption we could not make due to, e.g., ferromagnetic disturbances affecting the definition of coordinate axes of individual IMUs.

Long et al. (2023) used just a single shank-mounted IMU for predicting knee flexion angle with an RMSE of $3.30 \pm 0.69°$ [32]. This error is smaller than the RMSE we got for predicting the KFA predictor ($3.75 \pm 1.54°$, Table 7.1). Most likely the error we would have for predicting the KFA time series would be greater than the error we got for predicting the KFA predictor (which is just the difference of two points in the time series), because possible systematic error in our time series estimates is eliminated by the KFA predictor extraction process. Therefore, the method of Long et al. is more accurate than ours, even though they used just one IMU. However, they used a more involved machine learning model (random forest regression), so the accuracy advantage comes at a cost of reduced simplicity. However, because our solution uses machine learning (artificial neural networks) anyway, adopting the method of Long et al., or another machine learning method, to improve accuracy could increase the accuracy of predicting KJCF peaks without compromising usability.

8.3.4 Knee flexion angle from video camera data

Out of the four predictor estimates (KFA and walking speed from IMU data and those from VC data), KFA from VC data had the highest error. This result is similar to the results of Pearl et al. (2023), where joint flexion angle estimates from IMU data and VC data (keypoints detected with an HPE algorithm with multiple cameras and reconstructed in 3D) were compared against reference values from marker-based motion capture and the estimates from VC data were less accurate than those from IMU data [177].

In literature, IMU and VC data have also been combined to estimate kinematics with better accuracy than either modality could separately [177]. However, this direction is not feasible in the context of our aim, where we try to develop a simple, modular, and portable method for predicting KJFCs; using both IMUs and video cameras would be more cumbersome than simply using either, even if accuracy could be improved in doing so.

In study **III**, because we conducted walking measurements in two directions, the analyzed leg of the participant was occluded by the other leg in one of the directions. This occlusion had no significant effect on the predicted KJCF peaks, i.e., the effect of occlusion on identified keypoints didn't lead to any observable changes in KJCF prediction error when comparing between walking trials where the analyzed leg was occluded or clearly visible. This indicates that OpenPose is, at least in our measurement conditions, resistant to partial occlusion of limbs. Ino et al. (2023) evaluated the accuracy of OpenPose for estimating lower limb kinematics and found that kinematics of both the camera-side leg and the opposite leg were sufficiently robust for clinical evaluation (i.e., mean absolute errors of lower limb sagittal joint angles with their 95% confidence intervals were below 5 degrees for both the camera-facing leg and the partially occluded leg, indicating that a single camera could be used for clinical gait analysis), although the error on the opposite leg was slightly greater [77]. Wade et al. (2023) also found that tracking occluded limbs with OpenPose was less reliable than tracking the visible side, although tracking the knee joint was more resistant to occlusion than, e.g., tracking the hip joint; nonetheless, they could only support the use of OpenPose in clinical settings for tracking visible limbs [23]. Moreover, for visible hip and knee joints, Wade et al. found that the errors of using single camera-based estimation of sagittal and frontal joint angles were near or within the error ranges of marker-based motion capture. Therefore, while OpenPose's resistance to occlusion can be useful in many out-of-lab applications as well as in motion laboratories, it may not be robust enough for tracking occluded limbs in clinical settings, where no more than a single camera should be required.

When using OpenPose to identify keypoints, we ran into a number of issues including keypoints switching between pairs of extremities (e.g., OpenPose placing left leg keypoints on the right leg for a frame or several consecutive frames). This has been observed in at least one other study and methods exist to correct these artefacts [104], but after initial tests in study **III**, we decided not to use them because they complicated the analysis pipeline without significantly improving the accuracy of predicting KJCF peaks. Nonetheless, other studies may benefit from using such methods. Moreover, proper study preparation, i.e., considering lighting conditions, recording angle, clothing of participants and the environment visible in the background, is a good way to mitigate the chances of these issues from occurring in the first place.

8.4 PREDICTING KNEE JOINT LOADING

8.4.1 Practical feasibility of predicting knee joint contact force peaks

The simplicity of our predictor sets ensure that the demographic predictors (mass, height, age, sex) should be available in most settings, while the others (walking speed and knee flexion or abduction-adduction angle) must be estimated somehow.

The results of study **III** showed that the knee flexion angle predictor is irrelevant if walking speed is included (i.e., prediction accuracy is hardly affected by further including the KFA predictor), which simplifies the use of the method because walking speed is easier to estimate than the knee flexion angle predictor. It was predicted using IMUs or sagittal video feed in study **III**, but if no knee flexion angle estimation is required, we can forgo both modalities and use even simpler ones, e.g., a stopwatch or a photocell. Therefore, predicting KJCF peak with the accuracies reported in study **III** should be feasible in various settings. However, considering the fact that our IMU- or VC-based estimates of walking speed were quite accurate, the accuracy of predicting KJCF peaks may suffer if other modalities cannot accurately estimate walking speed. Because walking speed was observed to be a very important predictor, its erroneous values will bring similar error to predicted KJCF peaks.

Furthermore, if prediction of compartment-specific loading is required, knee abduction-adduction angle may not be informative enough for accurate prediction of compartmental loading peaks. Although it can be estimated, e.g., with Open-

Pose [33], other methods may have to be designed to find a predictor that better corresponds with the loading peaks in specific compartments.

8.4.2 Clinical potential of the methods

Because predicting the joint loading of patients is one of the hypothetical future uses of the developed methods, their practical suitability to clinical settings should be considered. Out of the two portable modalities used (IMUs and video cameras), IMUs are still somewhat uncommon equipment. However, we can expect that each modern office-like environment (e.g., a clinician's office appointment) will have access to a webcam and a computer, thus enabling the video camera-based method. A more likely limitation for a video camera is the space required for walking – even without a motion laboratory, the video camera based method assumes there is enough room for walking several meters in a straight line and placing the webcam distant enough from the walking line to capture the patient in the field of view. This requirement may not be fulfilled in most clinical appointments. Therefore, no matter the portable modality used to estimate predictors, predicting knee joint contact force peaks from measured gait may be unrealistic during a normal clinical appointment. However, a separate larger room to conduct the walking measurements is still much more accessible than sending the patient to the nearest fully equipped motion laboratory. In our studies, we intentionally limited the number of required cameras to one (whereas some alternative studies utilizing HPE algorithms use multiple cameras [22,39]), used a limited number of inertial measurement units (in fact, just one IMU can be used to estimate, e.g., walking speed [101]), and kept the two modalities separate (instead of fusing them, even if that could improve results [177]) to keep our method for predicting KCJF peaks as modular and accessible as possible.

Additionally, OpenPose has hardware requirements that increase with the resolution. The higher the resolution OpenPose uses, the more video memory (VRAM) is required. Running OpenPose at a resolution of 1920x1080 pixels can take more than 30 GB of VRAM, which only specialized graphics computing systems have. This limits the availability of OpenPose if high-resolution analysis is desired. However, we noticed that using a high resolution with images where a person is fairly close to the camera will actually result in false detections in the background and reduce the correct identification of the person's keypoints. Hence, in study **III**, we used a resolution of 480x272 pixels, which detected keypoints on our participants well and only required approximately 4 GB of VRAM. This amount is present in many modern computers, although some older laptops may not have enough VRAM. Therefore, a low-resolution keypoint analysis using OpenPose is possible with modern office-grade computers, making video cameras an available modality during physical appointments also in terms of computer hardware. In addition to downloading and configuring OpenPose, this requires only a webcam, so video camera-based analysis may be a better option than inertial measurement units in routine use.

In addition to physical requirements of room and devices, other factors of practical suitability for clinical settings exist. For instance, Tohka et al. (2021) name usability for end users and ease of integration in existing processes and infrastructures [143] as some criteria for an algorithm to succeed in healthcare. As clinicians cannot be expected to be computer experts, these points are emphasized.

Perhaps the most important point, however, is that the clinical advantage of using the methods should be proven [143]. Because physical loading estimates have no clinical value in isolation, their value in designing better joint implants, improving physical rehabilitation, or predicting the onset or progression of loading-related pathologies should be demonstrated through clinical studies. Otherwise, clinicians have no motivation to adopt the developed methods even if they have the means. In pursuit of this goal, balanced datasets should be used to minimize bias in the trained models [38, 180, 181]. Furthermore, the methods should be able to communicate the recommendations for clinical outcomes clearly (e.g., "lose weight and learn to walk slower to reduce the pain in your joint") instead of providing metrics that have little meaning to a clinician or patient (e.g., "at this walking speed, your medial compartmental loading peaks at 2350 Newtons").

Additionally, explainability and understandability of the results retrieved using these methods is important [38, 137, 143, 180]. This point is emphasized in "black box" deep neural network models where the relationship between the input and the output is unclear [38, 137, 180]. In such cases, the use of other methods simply to explain the models in common terms may be required [137,180,181]. Our models in studies **II** and **III** were designed to utilize shallow neural networks (i.e., networks with just one hidden layer) with a relatively small number of nodes in any layer (e.g., six input nodes, one hidden node, and one output node in study **III**). Furthermore, we formulated the relevant features (demographic variables, walking speed, knee angle) directly in the input predictor sets instead of using difficult-to-interpret input data such as marker trajectories. These design choices makes our models more understandable than many other "black box" solutions. Therefore, the relationship between the input features and the output KJCF peaks is mostly comprehensible even if the exact nonlinear function approximated by our ANN models is not known.

Finally, it should be noted that this doctoral work only aimed to develop simple methods for out-of-laboratory estimation of knee joint loading peaks. Although clinical settings were presented as a potential out-of-laboratory use case, nothing rules out non-clinical use, and actual adoption of the methods to clinical use or convincing clinicians of their value is not within the scope of this doctoral work.

For the purposes of gait retraining, the model should be able to at least tell if a gait modification results in an increase or decrease in knee joint loading; Kaneda et al. (2023) call this ability "directional accuracy" [114]. We can assume that demographic predictors (i.e., mass, height, age, and sex) remain constant during gait retraining, so the model should be sensitive to changes in the other predictors, i.e., walking speed and the knee flexion angle predictor in study **III**. Literature has indicated that walking speed is directly proportional to knee joint loading [128] and the effect of knee flexion angle can be assumed to also be directly proportional, because during most of the stance phase, the more flexion there is, the greater the moment arm of the ground reaction force is and the more muscles must work to keep the moments balanced [50, 51]. The direct proportionality between knee flexion angle and knee joint loading was also supported by our correlation analysis (section 7.5). Therefore, at present, our model incorporating just these two predictors in addition to demographic information offers no new discoveries. However, it can be used to estimate the amount of change in knee joint loading instead of just the direction of change. The model of study **III** could be more useful in this context if it was further developed to utilize more predictors describing gait (e.g., more sagittal-plane joint angles in addition to knee flexion), thus enabling evaluating the effect modifying them has on knee joint loading.

8.4.3 Defining the criteria for prediction error

We have quantified the error of predicting knee joint contact force peaks. However, assessing how much error is acceptable depends on the application [38], which there are several. As this doctoral work was focused on the development of methods rather than their application, we cannot provide any frames of reference for acceptable error in real-world applications.

However, we can try to estimate the theoretically achievable minimum error by looking at the inter-trial variance of knee joint loading peaks in trials with identical instructions for the participant. Achieving generalization accuracy greater than the inherent variability of knee joint loading peaks should be impossible. This variability was reported in study **II** for knee joint loading peaks calculated from the motion capture data of different datasets. For the full-stance maximum peak, the mean variability varied between 170 and 380 Newtons and showed great variability between datasets. This variability between datasets likely constitutes a combination of technical differences and differences in gait of the subjects of different datasets. Four of the datasets comprised healthy subjects, while one comprised diagnosed KOA patients. However, in terms of variability, the KOA patient dataset was in the middle-to-low range of the mean variabilities (216 Newtons), implying that pain-induced variability of steps cannot be the main cause of KJCF peak variability between datasets (otherwise the mean variability of the KOA dataset should be higher than the mean variability of other datasets). Hence, technical differences, such as the marker set and equipment used and the instructions given to subjects may have a significant effect on the variability of the knee joint loading peaks retrieved from musculoskeletal simulation and modeling. Estimating the theoretically achievable minimum error is therefore difficult and depends on the data used. This observation conforms with literature, where the quality of the data is often emphasized in training viable machine learning models [143,182,183].

Our prediction errors are at the same scale as the variabilities, which implies good generalization. Prediction errors much greater than the variabilities would have implied underfitted or overfitted networks.

Finally, if we wish to estimate the *in vivo* contact forces in the knee, our musculoskeletal models should be as accurate as possible. This criterion does not only refer to the anatomical validity of the models' components, but also that simulations utilizing the models should give accurate results that match reality. In studies **II** and **III**, we quantified prediction error by comparing predictions to reference loading peaks from musculoskeletal modeling and simulation, but those reference values may differ from the actual *in vivo* forces. Our studies, like many other biomechanical studies of human motion, scaled generic musculoskeletal models to create subject-specific models, but more involved methods such as generating the model geometry from magnetic resonance images exist and may result in, e.g., different muscle moment arms than generic model scaling [115]. Therefore, if we wish to minimize the error between predicted and *in vivo* forces, we must not only hone our prediction model, but the musculoskeletal modeling workflow as well. However, depending on the application of the method, it may not be necessary (e.g., if we only wish to determine if an intervention decreases or increases contact forces rather than exactly quantifying the change) [114].

8.4.4 Modeling the relationship between knee joint loading and simple predictors

In study **II**, we did not notice the number of nodes having an effect on the prediction accuracy. We used this observation to justify using only one hidden node in study **III**. This may seem somewhat counterintuitive in artificial neural networks, whose strength lies in approximating arbitrary functions [136], i.e., arbitrary relationships between predictors and predicted output. The robustness of modeling different relationships between variables comes from the different connections between the nodes and the nonlinearity in the activation function of the node. If we use only one hidden node, we seriously limit the variety of relationships between variables that the network can be trained to model, i.e., the representational capacity of the model [135].

For example, consider the network architecture we used in study **III**. With six inputs, a single hidden node, and a single output node, we have six weights between the inputs and the hidden layer, and one weight connecting the hidden layer to the output layer. In the hidden layer, the linear combination of the inputs x_i and weights w_i (plus bias b) is fed to the activation function f to produce the output A of the node according to

$$A = f(\sum_{n=1}^{6} w_i x_i + b). \tag{8.1}$$

Depending on the values of the weights, some inputs may be amplified with respect to the others. However, only one kind of weight configuration can be used in the network.

Now consider a network with ten hidden nodes instead of just one. We have ten equations of the above type, each with unique weight configurations. This means that, e.g., while the first node may model the combined effect of the first two inputs, the second node may model the effect of the fourth input alone, the third node may model the effects of all but the first input, and so forth. The ten weights connecting the hidden layer to the output layer then control how strongly each of these modeled relationships affects the final output of the network. In plain terms, this network with ten hidden nodes can model much more complicated relationships than a network with a single hidden node.

Despite this limitation, we still decided to use a single hidden node in study **III**. In part this was because a network with a single hidden node was slightly easier to write in a function form, which we figured would be helpful in understanding the model and consequently adopt it in clinical settings. Moreover, we deduced that the indifference to the number of nodes observed in study **II** meant that there were no complicated relationships to model between variables given our data and network architecture. Note that we cannot deny the presence of complicated variable relationships altogether, but rather state that in our limited dataset and imperfect network architecture, the relationships between our simple predictor set and the response variables were very simple — simple enough to be modeled with a single hidden node without an observable increase in prediction error.

If we keep simplifying the predictor set, e.g., by including only demographics and walking speed and omitting information about knee alignment (as study **III** hints could be one future direction), then our network architecture should remain sufficient in future. However, we should then ask ourselves if instead of artificial neural networks we should use more classical and understandable approaches such as linear regression, which may be able to model our data as well as the simple neural network and is furthermore much easier to explain.

On the other hand, if we leverage the potential of human pose estimation algorithms or IMUs to include more features from joint angle time series in our predictor set, we may have to increase the number of hidden nodes to model the relationships in the data.

8.4.5 Comparison to multiple linear regression

We note that Figure 7.5 is very similar to Figure 7.3. This similarity indicates that our shallow neural networks with a single hidden node approximate a near-linear relationship. The fact that the simple neural networks of study **III** are so close to linear regression indicates that we may have used a needlessly complex method to model a fairly simple relationship between the predictors and the response variables.

On the bright side, the observation shows that multiple linear regression may be suitable to predict KJCF peaks from simple predictors. Because multiple linear regression is simpler and better understood than artificial neural networks, the method could be more approachable to people lacking the technical expertise to understand ANNs. For instance, our "simple" shallow neural networks with a single hidden node have 23 parameters to estimate (three preprocessing factors, six weights from input to hidden layer, one bias term in the hidden layer, one weight from hidden to output layer, one bias term in the output layer, three postprocessing factors); in contrast, multiple linear regression only has seven (six coefficients for the predictors and one intercept term). As an equation, multiple linear regression is also far simpler to express than a neural network prediction model because multiple linear regression can be written as a linear combination.

Before we abandon feedforward neural networks in favour of linear regression, we should note that changing the predictor set may result in relationships between the predictors and the response variables that linear regression is no longer able to model accurately. Hence, unlike multiple linear regression, feedforward networks should have a more stable performance than multiple linear regression if the data, inputs or outputs of the model are changed. Furthermore, nonlinear regression may provide an easily interpretable alternative to feedforward neural networks when the data contains nonlinear relationships.

Because multiple linear regression resulted in smaller error than ANNs with demographic predictor sets (Figure 7.6), it could be used to create prediction models that are more intuitively understandable than ANNs when working with very simple predictors. However, ANNs have the potential to model more complicated relations than linear regression, and have the advantage of not having to assume the underlying function between predictors and the output (unlike, e.g., polynomial regression). Therefore, ANNs can be seen as a more robust solution. However, their adoption to non-expert use may be limited by the fact that they are difficult to understand.

A potential future model could leverage the strength of both machine learning and analytical regression. A validation set could be used to optimize the L2 regularization term of ridge regression so that the generalization accuracy of prediction is maximized. Furthermore, pre-processing such as centering and scaling of predictors could be tried to further reduce prediction error of the analytical regression model. In conclusion, if our research continues to the direction of very simple predictor sets and transparent algorithms, analytical regression is a viable option. Otherwise, if we want to use more complicated predictor sets or instead of single-valued peaks predict entire time series with the help of, e.g., joint angle time series, we may need to continue using artificial neural networks or similar machine learning algorithms to maximize the generalization capability of the prediction models.

8.4.6 Nonlinear effect of predictors on knee joint contact force peaks

The high importance of mass is no surprise because following physical intuition, the more the person weights, the more loading the knee joint experiences whenever the leg is supported on [5]. Therefore, we expect mass to be the most important predictor when the responses are not normalized to body weight.

The KFA predictor and walking speed follow the mass as the next most important predictors.

The KFA predictor is more important than walking speed for many response variables, and the difference in the importances of the two are emphasized when the response variable is the loading response peak. This emphasized difference can be explained by the fact that the magnitude of the KFA predictor is largely defined by the maximum of the knee flexion angle during the first half of its time series, which almost coincides with the loading response peak in time domain (Figure 8.1). The greater the knee flexion at that time instant is, the greater the moment arm between the ground reaction force and the knee, necessitating a greater knee extension moment to balance the forces in the musculoskeletal model. The knee extension moment is generated by muscles that compress the knee, thus leading to an increased compressive tibiofemoral contact force. Hence, the importance of the KFA predictor is highlighted when the loading response peak occurs.

The difference in importances between the KFA predictor and walking speed is small when predicting terminal extension peaks.

Interestingly, Figure 7.3 shows that the error of predicting loading response peaks is smaller when the predictor set includes demographic predictors and walking speed than when the predictor set includes demographic predictors and the KFA predictor. From the importances of the predictors, the opposite could be expected. However, we should note that the importances are calculated from a trained neural network model utilizing all six predictors. Therefore, they only state how sensitive the model is to each predictor when all six predictors are used and the importances do not apply to cases where some predictors are isolated from the predictor set such as in Figure 7.3.

8.4.7 Our studies among the existing corpus of machine learning in biomechanics

In their survey of existing literature, Halilaj et al. (2018) noted the exponentially increasing trend of using machine learning methods in human movement biomechanics [38]. The number of subjects in those studies ranged from 4 to 2956 with a median of 40, putting study **II** (290 subjects) and **III** (336 subjects) well above the median. The minimum acceptable number of subjects (or data points in general) depends on the size of the feature space, i.e., the number of input variables used as features [38]. If the number of features is larger than the number of observations, overfitting becomes likely [38]. In studies **II** and **III**, where one of our priorities was simplicity of the prediction model, this number of input variables was six at most. The number is smaller than the sizes of feature spaces in some other studies, where

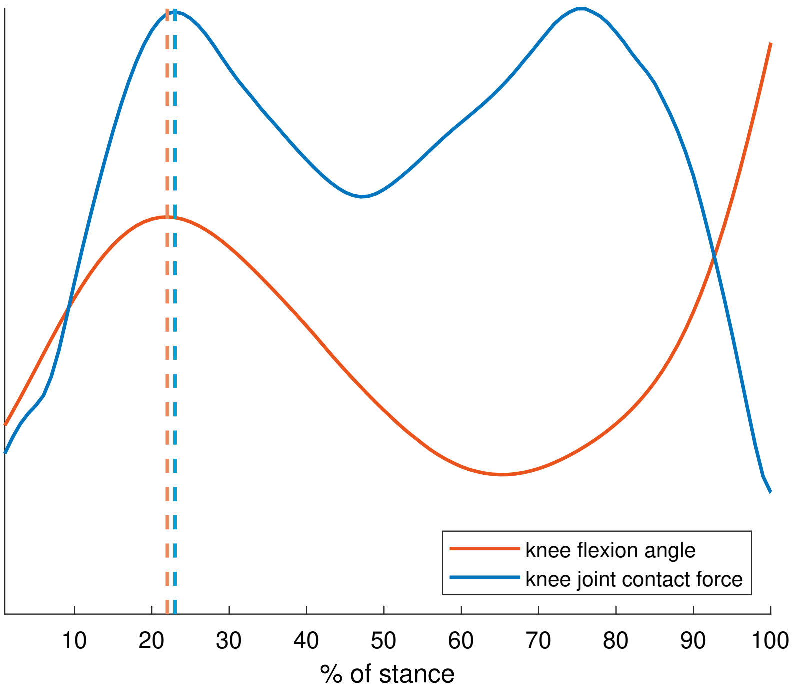

Figure 8.1: The near-coincidence of the first maximum of the knee flexion angle and the loading response peak of tibiofemoral compressive knee joint contact force. Dashed vertical lines indicate the positions of the maxima, showing that they occur very close to one another. The curves are calculated as means over the self-selected comfortable speed walking trials of participants of study **III**. The curves are normalized to fit in the same plot.

the predictor set is not simplified and instead uses, e.g., a large number of marker trajectories [18, 19], joint angles [20], force plate data [19, 20], or electromyographic signals [18]. Therefore, because studies **II** and **III** have a comparably large amount of data, a very simple model (in the context of artificial neural networks), and a very limited number of features, the model should generalize well without meaningful overfitting. The simplicity of the model and its features also reduces the need for dimensionality reduction methods (e.g., principal component analysis), which were not used in studies **II** and **III**.

While planning to minimize bias in collected datasets should always be one of the goals in studies involving machine learning, we should acknowledge that all real data contains some bias.

8.5 LINEAR CORRELATION BETWEEN KINEMATICS AND KNEE JOINT CONTACT FORCES

The relationship between the predictor and the response does not have to be linear when using nonlinear regression algorithms. Therefore, the results of our linear correlation analysis cannot rule out any features in the general case, but instead highlight potentially useful features. However, if we use really simple methods such as multiple linear regression, the results of this analysis can also show us which features have little importance in predicting knee joint loading with linear regression.

For knee flexion angle time series, both the difference between the maximum and the minimum and the simple maximum correlated similarly with knee joint loading-related variables. While the maximum is simpler to calculate, we used the difference in study **III**. Justifications for this include the fact that the difference is not affected by a possible offset in the time series data of knee flexion angle. In knee flexion angle time series from video camera data, there should be no offset, but in IMU-derived time series, an offset could be present if our assumption of fully extended knees does not apply at the beginning of the walking trial. Furthermore, an offset may be present in OpenSim-analyzed motion capture data if any is introduced during the scaling of the muscusloskeletal model. This offset could be important to note because although we do not feed the networks motion capture data when we use them, we used motion capture data to train them.

The maxima of hip flexion angle and knee flexion angle occur during the first half of stance, with the hip flexion maximum occurring at the very beginning of the stance phase and the knee flexion maximum coinciding with the loading response part of stance. Therefore, the correlations between the maxima and knee joint loading variables related to the loading response are expected. Knee and hip joint angle extrema cannot be attributed to the terminal extension part of stance as distinctly, which may explain why the knee joint loading variables that had strong correlation were mostly related to loading response rather than terminal extension. It is hence possible that the extremum information from joint angles is more useful in predicting the loading response peak than the terminal extension peak.

Hip rotation angle correlated with some lateral loading variables and loading variables not exclusively related to loading response. Therefore, including it as a predictor could be useful in reducing prediction error of lateral loading peaks (which had the higher prediction error than medial or summed peaks in studies **II** and **III**) or terminal extension peaks. However, the hip rotation angle is difficult to estimate from video data, especially in the sagittal plane. Even with IMU data, estimating hip rotation angle can prove difficult because of its small range of motion compared to flexion-extension angles, which may result in relatively more noise in the orientation data. Hence, accurate estimates of hip rotation angles may be inaccessible with some portable modalities. However, an IMU on the ankle may provide a substitute measure of hip rotation provided knee rotation and ankle motion outside plantarflexion and dorsiflexion are negligible.

As a closing remark, we expect the musculoskeletal model to affect the correlations strongly. Thus, the results of the correlation analysis may be poorly generalizable to other musculoskeletal models.

8.6 FUTURE DIRECTIONS

We see two main development directions for the methods: 1) keeping the methods modular and simple, and 2) abandoning modularity to increase prediction accuracy.

In the first direction, artificial neural networks could be replaced by simpler, more intuitive methods (e.g., least squares regression) while keeping the predictor set mostly as it is (e.g., consisting of demographic variables and possibly other predictors that are easily estimated, such as walking speed). Alternatively, if we identify more predictors that are simple to estimate while keeping the models modality-agnostic, we can keep using artificial neural networks and use more nodes to leverage larger datasets that allow us to leverage complex nonlinear relationships between the predictors and output.

In the second direction, we could integrate the measurement modality into the prediction model and create a model that derives predictors that are more descriptive than our current simple predictors. This has been done by several studies in the field, e.g., by developing models for predictors specifically from video data [63,149] or inertial measurement unit data [11,30,75,89], or even a combination of both [177]. While becoming modality-specific would probably increase prediction accuracy, it has already been done in existing literature and could limit the usability of the method. Additionally, the results could improve if more data was included [137], as the total amount of trials is still quite small. This addition of real or synthesized data could allow leveraging more complicated neural networks than the ones used in study **III**.

In either future direction, the accuracy of estimating predictors from the measurement modalities (whether they are modular or tied to the prediction model) should be improved. For instance, the accuracy of predicting KJCF peaks would improve if we reduced the error of estimating the KFA predictor with OpenPose, which was notably high in study **III**. Camera-based methods in general seem like a good future investment because they require no physical contact with the subject and most out-of-laboratory environments have access to a video camera (e.g., webcam) or one could at least be used in those environments. Additionally, while OpenPose is well-known and relatively simple to use, top-down approaches that first detect people and then their joints may be more accurate, so OpenPose is not necessarily the ideal HPE algorithm for best-possible keypoint tracking especially for clinical conditions [43]. Furthermore, as most HPE algorithms currently are not specifically made with biomechanical validity in mind, it would be interesting to see how a model trained for detecting precise joint centers as keypoints, or for specific sports [24], would improve the results. Such specialized models could be achieved, e.g., by taking existing HPE models and further training them using a small corpus of application-specific and accurately labeled data.

In our case, the future development choices likely depend on the end use case of the predicted KJCF peaks. Although we framed our models for general out-of-laboratory use, their development was driven by the fact that the loading predictions could be used as more personalized inputs to finite element models for improved out-of-laboratory prognostics and rehabilitation guidance of patients with musculoskeletal disorders [27,184,185].

9 Summary and conclusions

In this doctoral work, we developed methods for real-time computation of joint kinematics, and the prediction of knee joint loading peaks outside the motion laboratory. Our knee joint loading predictions were non-personalized, i.e., data from a single subject could only appear in training, validation, or test subset of data, but not several subsets. Moreover, the predictions are not specifically designed to model intra-subject variation. Our predictions were also subject-specific, i.e., in training the networks, we used data from simulations that utilized subject-specifically scaled generic musculoskeletal models, including subject-specific bony dimensions of the knee. Our predictions had error on the same scale as the natural variability of knee joint loading peaks in repeated walking trials.

Supporting our first hypothesis, study **I** presented a software for real-time inverse kinematics analysis using data from inertial measurement units. It was implemented as an plug-in that extended the application programming interface of the popular musculoskeletal modeling and simulation software OpenSim. Therefore, it is also compatible with the variety of different and customizable musculoskeletal models used by OpenSim. The software developed in this study enables real-time estimation of joint angles outside the motion laboratory using inertial measurement units. Although it does not support the current version of OpenSim and it was not the first tool to provide real-time inverse kinematics using OpenSim, its open-source code is available as an example of how to use the programmatical tools of OpenSim in real-time analysis.

To support our second and third hypotheses, studies **II** and **III** presented a neural network solution for predicting knee joint contact force peaks. The difference from existing machine learning solutions is that we designed our method with simplicity (input variables that are mostly demographic and easily collected), portability (after training, usable without motion capture equipment), and modularity (use demonstrated with IMUs and video data, but other methods to estimate the KFA predictor and walking speed could be used instead) in mind — in the end, this resulted in better interpretability than artificial neural network solutions usually have (the weights of the network describe the importance of the predictors), although the simple architecture does not leverage the full approximation potential of nonlinear functions that typical artificial neural networks have. The solutions developed in studies **II** and **III** enable estimation of knee joint loading peaks using, e.g., inertial measurement units or a video camera.

Future studies should investigate the possibility of including more predictors that allow more accurate estimation of knee joint loading while retaining the portability and modularity of the developed methods. Furthermore, while we developed methods for accessible out-of-laboratory use, we did not test them specifically in clinical settings, e.g., for predicting the structural degeneration of the knee joint or informing gait retraining to exert less loading on the knee. Therefore, the suitability of the developed methods should be tested in one or more of the proposed use cases.

BIBLIOGRAPHY

[1] M. D. Chivers and S. D. Howitt, "Anatomy and physical examination of the knee menisci: a narrative review of the orthopedic literature.," *The Journal of the Canadian Chiropractic Association* **53**, 319–33 (2009).

[2] H. E. Rubash, *The Adult Knee*, 2 ed. (Wolters Kluwer Health, 2021).

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

[4] P. Sritharan, Y. Lin, and M. G. Pandy, "Muscles that do not cross the knee contribute to the knee adduction moment and tibiofemoral compartment loading during gait," *Journal of Orthopaedic Research* **30**, 1586–1595 (2012).

[5] D. D. D'Lima, B. J. Fregly, S. Patil, N. Steklov, and C. W. Colwell, "Knee joint forces: Prediction, measurement, and significance," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* **226**, 95–102 (2012).

[6] B. J. Fregly, T. F. Besier, D. G. Lloyd, S. L. Delp, S. A. Banks, M. G. Pandy, and D. D. D'Lima, "Grand challenge competition to predict in vivo knee loads," (2012).

[7] J. A. Wilson and D. Kobsar, "Osteoarthritis year in review 2020: mechanics," *Osteoarthritis and Cartilage* **29**, 161–169 (2021).

[8] T. Miyazaki, "Dynamic load at baseline can predict radiographic disease progression in medial compartment knee osteoarthritis," *Annals of the Rheumatic Diseases* **61**, 617–622 (2002).

[9] D. T. Felson, "Osteoarthritis as a disease of mechanics," (2013).

[10] A. Litwic, M. H. Edwards, E. M. Dennison, and C. Cooper, "Epidemiology and burden of osteoarthritis," *British Medical Bulletin* **105**, 185–199 (2013).

[11] B. J. Stetter, F. C. Krafft, S. Ringhof, T. Stein, and S. Sell, "A Machine Learning and Wearable Sensor Based Approach to Estimate External Knee Flexion and Adduction Moments During Various Locomotion Tasks," *Frontiers in Bioengineering and Biotechnology* **8** (2020).

[12] A. Mündermann, C. O. Dyrby, and T. P. Andriacchi, "Secondary gait changes in patients with medial compartment knee osteoarthritis: Increased load at the ankle, knee, and hip during walking," *Arthritis & Rheumatism* **52**, 2835–2844 (2005).

[13] I. Kutzner, A. Bender, J. Dymke, G. Duda, P. V. Roth, and G. Bergmann, "Mediolateral force distribution at the knee joint shifts across activities and is driven by tibiofemoral alignment," *Bone and Joint Journal* **99B**, 779–787 (2017).

[14] X. Li, C. Wang, Y. Guo, and W. Chen, "An Approach to Developing Customized Total Knee Replacement Implants," *Journal of Healthcare Engineering* **2017**, 1–8 (2017).

[15] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "OpenSim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering* **54**, 1940–1950 (2007).

[16] A. Seth, J. L. Hicks, T. K. Uchida, A. Habib, C. L. Dembia, J. J. Dunne, C. F. Ong, M. S. DeMers, A. Rajagopal, M. Millard, S. R. Hamner, E. M. Arnold, J. R. Yong, S. K. Lakshmikanth, M. A. Sherman, J. P. Ku, and S. L. Delp, "OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement," *PLoS Computational Biology* **14** (2018).

[17] A. Karatsidis, G. Bellusci, H. M. Schepers, M. de Zee, M. S. Andersen, and P. H. Veltink, "Estimation of ground reaction forces and moments during gait using only inertial motion capture," *Sensors (Switzerland)* **17** (2017).

[18] M. M. Ardestani, Z. Chen, L. Wang, Q. Lian, Y. Liu, J. He, D. Li, and Z. Jin, "Feed forward artificial neural network to predict contact force at medial knee joint: Application to gait modification," *Neurocomputing* **139**, 114–129 (2014).

[19] L. Rane, Z. Ding, A. H. McGregor, and A. M. Bull, "Deep Learning for Musculoskeletal Force Prediction," *Annals of Biomedical Engineering* **47**, 778–789 (2019).

[20] G. Giarmatzis, E. I. Zacharaki, and K. Moustakas, "Real-time prediction of joint forces by motion capture and machine learning," *Sensors (Switzerland)* **20**, 1–19 (2020).

[21] I. Jonkers, E. Beaucage-Gauvreau, B. A. Killen, D. Gupta, L. Scheys, and F. D. Groote, "In Silico Biomarkers of Motor Function to Inform Musculoskeletal Rehabilitation and Orthopedic Treatment," *Journal of Applied Biomechanics* **39**, 284–293 (2023).

[22] L. Needham, M. Evans, D. P. Cosker, L. Wade, P. M. McGuigan, J. L. Bilzon, and S. L. Colyer, "The accuracy of several pose estimation methods for 3D joint centre localisation," *Scientific Reports* **11**, 20673 (2021).

[23] L. Wade, L. Needham, M. Evans, P. McGuigan, S. Colyer, D. Cosker, and J. Bilzon, "Examination of 2D frontal and sagittal markerless motion capture: Implications for markerless applications," *PLOS ONE* **18**, e0293917 (2023).

[24] N. J. Cronin, J. Walker, C. B. Tucker, G. Nicholson, M. Cooke, S. Merlino, and A. Bissas, "Feasibility of OpenPose markerless motion analysis in a real athletics competition," *Frontiers in Sports and Active Living* **5** (2024).

[25] G. D. Raimondo, M. Willems, B. A. Killen, S. Havashinezhadian, K. Turcot, B. Vanwanseele, and I. Jonkers, "Peak Tibiofemoral Contact Forces Estimated Using IMU-Based Approaches Are Not Significantly Different from Motion Capture-Based Estimations in Patients with Knee Osteoarthritis," *Sensors* **23**, 4484 (2023).

[26] A. J. V. D. Bogert, T. Geijtenbeek, O. Even-Zohar, F. Steenbrink, and E. C. Hardin, "A real-time system for biomechanical analysis of human movement and muscle function," *Medical and Biological Engineering and Computing* **51**, 1069–1077 (2013).

[27] M. E. Mononen, M. K. Liukkonen, and R. K. Korhonen, "Utilizing Atlas-Based Modeling to Predict Knee Joint Cartilage Degeneration: Data from the Osteoarthritis Initiative," *Annals of Biomedical Engineering* **47**, 813–825 (2019).

[28] V. Bonnet, C. Mazza, P. Fraisse, and A. Cappozzo, "Real-time estimate of body kinematics during a planar squat task using a single inertial measurement unit," *IEEE Transactions on Biomedical Engineering* **60**, 1920–1926 (2013).

[29] D. Stanev, K. Filip, D. Bitzas, S. Zouras, G. Giarmatzis, D. Tsaopoulos, and K. Moustakas, "Real-Time Musculoskeletal Kinematics and Dynamics Analysis Using Marker- and IMU-Based Solutions in Rehabilitation," *Sensors* **21**, 1804 (2021).

[30] B. J. Stetter, S. Ringhof, F. C. Krafft, S. Sell, and T. Stein, "Estimation of knee joint forces in sport movements using wearable sensors and machine learning," *Sensors (Switzerland)* **19**, 1–12 (2019).

[31] W. Teufl, M. Miezal, B. Taetz, M. Fröhlich, and G. Bleser, "Validity of inertial sensor based 3D joint kinematics of static and dynamic sport and physiotherapy specific movements," *PLOS ONE* **14**, e0213064 (2019).

[32] T. Long, J. Outerleys, T. Yeung, J. Fernandez, M. L. Bouxsein, I. S. Davis, M. A. Bredella, and T. F. Besier, "Predicting ankle and knee sagittal kinematics and kinetics using an ankle-mounted inertial sensor," *Computer Methods in Biomechanics and Biomedical Engineering* 1-14 (2023).

[33] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *arXiv* (2018).

[34] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience* **21**, 1281–1289 (2018).

[35] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**, 7157–7173 (2023).

[36] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv* (2022).

[37] M. Miezal, B. Taetz, and G. Bleser, "Real-time inertial lower body kinematics and ground contact estimation at anatomical foot points for agile human locomotion," *2017 IEEE International Conference on Robotics and Automation (ICRA)* 3256-3263 (2017).

[38] E. Halilaj, A. Rajagopal, M. Fiterau, J. L. Hicks, T. J. Hastie, and S. L. Delp, "Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities," *Journal of Biomechanics* **81**, 1–11 (2018).

[39] C. J. C. McGuirk, N. Baddour, and E. D. Lemaire, “Video-Based Deep Learning Approach for 3D Human Movement Analysis in Institutional Hallways: A Smart Hallway,” *Computation* **9**, 130 (2021).

[40] E. P. Washabaugh, T. A. Shanmugam, R. Ranganathan, and C. Krishnan, “Comparing the accuracy of open-source pose estimation methods for measuring gait kinematics,” *Gait and Posture* **97**, 188–195 (2022).

[41] J. Adolf, J. Dolezal, P. Kutilek, J. Hejda, and L. Lhotska, “Single Camera-Based Remote Physical Therapy: Verification on a Large Video Dataset,” *Applied Sciences* **12**, 799 (2022).

[42] L. Needham, M. Evans, L. Wade, D. P. Cosker, M. P. McGuigan, J. L. Bilzon, and S. L. Colyer, “The development and evaluation of a fully automated markerless motion capture workflow,” *Journal of Biomechanics* **144**, 111338 (2022).

[43] L. C. Büker, F. Zuber, A. Hein, and S. Fudickar, “HRDepthNet: Depth Image-Based Marker-Less Tracking of Body Joints,” *Sensors* **21**, 1356 (2021).

[44] M. Terreran, L. Barcellona, and S. Ghidoni, “A general skeleton-based action and gesture recognition framework for human–robot collaboration,” *Robotics and Autonomous Systems* **170**, 104523 (2023).

[45] R. M. H. McMinn and R. T. Hutchings, *A Colour Atlas of Human Anatomy*, 2 ed. (Wolfe Medical Publications Ltd, 1988).

[46] V. B. Duthon, C. Barea, S. Abrassart, J. H. Fasel, D. Fritschy, and J. Ménétrey, “Anatomy of the anterior cruciate ligament,” *Knee Surgery, Sports Traumatology, Arthroscopy* **14**, 204–213 (2006).

[47] J. A. Aragão, F. P. Reis, D. P. de Vasconcelos, V. L. C. Feitosa, and M. A. P. Nunes, “Metric Measurements and Attachment Levels of the Medial Patellofemoral Ligament: An Anatomical Study in Cadavers,” *Clinics* **63**, 541–544 (2008).

[48] K. N. Shah, S. F. DeFroda, J. K. Ware, S. C. Koruprolu, and B. D. Owens, “Lateral Patellofemoral Ligament: An Anatomic Study,” *Orthopaedic Journal of Sports Medicine* **5**, 232596711774143 (2017).

[49] P. Allard, I. A. F. Stokes, and J.-P. Blanchi, *Three-dimensional analysis of human movement* (Human Kinetics, 1995).

[50] L. T. Gruss, “Limb length and locomotor biomechanics in the genus *Homo*: An experimental study,” *American Journal of Physical Anthropology* **134**, 106–116 (2007).

[51] M. Hora, V. Sládek, L. Soumar, K. stráníková, and T. Michálek, “Influence of body mass and lower limb length on knee flexion angle during walking in humans,” *Folia Zoologica* **61**, 330–339 (2012).

[52] H. A. Gray, S. Guan, L. T. Thomeer, and M. G. Pandy, “Moment arm of the knee-extensor mechanism measured in vivo across a range of daily activities,” *Journal of Biomechanics* **123**, 110484 (2021).

[53] S. Glyn-Jones, A. J. Palmer, R. Agricola, A. J. Price, T. L. Vincent, H. Weinans, and A. J. Carr, "Osteoarthritis," *The Lancet* **386**, 376–387 (2015).

[54] K. D. Allen and Y. M. Golightly, "Epidemiology of osteoarthritis: state of the evidence," *Current Opinion in Rheumatology* **27**, 276–283 (2015).

[55] L. Sharma, "The role of proprioceptive deficits, ligamentous laxity, and malalignment in development and progression of knee osteoarthritis.," *The Journal of rheumatology. Supplement* **70**, 87–92 (2004).

[56] A. Mündermann, C. O. Dyrby, and T. P. Andriacchi, "A comparison of measuring mechanical axis alignment using three-dimensional position capture with skin markers and radiographic measurements in patients with bilateral medial compartment knee osteoarthritis," *Knee* **15**, 480–485 (2008).

[57] S. Terpstra, J. van der Velde, R. de Mutsert, D. Schiphof, M. Reijnierse, F. Rosendaal, L. van de Stadt, M. Kloppenburg, and M. Loef, "The association of clinical and structural knee osteoarthritis with physical activity in the middle-aged population: the NEO study," *Osteoarthritis and Cartilage* **29**, 1507–1514 (2021).

[58] M. Henriksen, M. W. Creaby, H. Lund, C. Juhl, and R. Christensen, "Is there a causal link between knee loading and knee osteoarthritis progression? A systematic review and meta-analysis of cohort studies and randomised trials," *BMJ Open* **4**, e005368–e005368 (2014).

[59] W. S. B. Burton, C. A. Myers, and P. J. Rullkoetter, "Machine Learning for Rapid Estimation of Lower Extremity Muscle and Joint Loading during Activities of Daily Living," *Journal of Biomechanics* 110439 (2021).

[60] R. Altman, E. Asch, D. Bloch, G. Bole, D. Borenstein, K. Brandt, W. Christy, T. D. Cooke, R. Greenwald, M. Hochberg, D. Howell, D. Kaplan, W. Koopman, S. Longley, H. Mankin, D. J. McShane, T. Medsger, R. Meenan, W. Mikkelsen, R. Moskowitz, W. Murphy, B. Rothschild, M. Segal, L. Sokoloff, and F. Wolfe, "Development of criteria for the classification and reporting of osteoarthritis: Classification of osteoarthritis of the knee," *Arthritis & Rheumatism* **29**, 1039–1049 (1986).

[61] P. R. Kornaat, R. Y. T. Ceulemans, H. M. Kroon, N. Riyazi, M. Kloppenburg, W. O. Carter, T. G. Woodworth, and J. L. Bloem, "MRI assessment of knee osteoarthritis: Knee Osteoarthritis Scoring System (KOSS) — inter-observer and intra-observer reproducibility of a compartment-based scoring system," *Skeletal Radiology* **34**, 95–102 (2005).

[62] D. Hunter, N. Arden, P. Conaghan, F. Eckstein, G. Gold, A. Grainger, A. Guermazi, W. Harvey, G. Jones, M. H. L. Graverand, J. Laredo, G. Lo, E. Losina, T. Mosher, F. Roemer, and W. Zhang, "Definition of osteoarthritis on MRI: results of a Delphi exercise," *Osteoarthritis and Cartilage* **19**, 963–969 (2011).

[63] M. A. Boswell, S. D. Uhlrich, Kidziński, K. Thomas, J. A. Kolesar, G. E. Gold, G. S. Beaupre, and S. L. Delp, "A neural network to predict the knee adduction moment in patients with osteoarthritis using anatomical landmarks obtainable from 2D video analysis," *Osteoarthritis and Cartilage* **29**, 346–356 (2021).

[64] A. Baliunas, D. Hurwitz, A. Ryals, A. Karrar, J. Case, J. Block, and T. Andriacchi, "Increased knee joint loads during walking are present in subjects with knee osteoarthritis," *Osteoarthritis and Cartilage* **10**, 573–579 (2002).

[65] S. D. Uhlrich, T. K. Uchida, M. R. Lee, and S. L. Delp, "Ten steps to becoming a musculoskeletal simulation expert: A half-century of progress and outlook for the future," *Journal of Biomechanics* **154**, 111623 (2023).

[66] J. W. Fernandez, M. Akbarshahi, H. J. Kim, and M. G. Pandy, "Integrating modelling, motion capture and x-ray fluoroscopy to investigate patellofemoral function during dynamic activity," *Computer Methods in Biomechanics and Biomedical Engineering* **11**, 41–53 (2008).

[67] D. L. Benoit, D. K. Ramsey, M. Lamontagne, L. Xu, P. Wretenberg, and P. Renström, "Effect of skin movement artifact on knee kinematics during gait and cutting motions measured in vivo," *Gait and Posture* **24**, 152–164 (2006).

[68] D. K. Ramsey and P. F. Wretenberg, "Biomechanics of the knee: methodological considerations in the in vivo kinematic analysis of the tibiofemoral and patellofemoral joint," *Clinical Biomechanics* **14**, 595–611 (1999).

[69] B.-M. You, P. Siy, W. Anderst, and S. Tashman, "In vivo measurement of 3-D skeletal kinematics from sequences of biplane radiographs: Application to knee kinematics," *IEEE Transactions on Medical Imaging* **20**, 514–525 (2001).

[70] W. R. Taylor, P. Schütz, G. Bergmann, R. List, B. Postolka, M. Hitz, J. Dymke, P. Damm, G. Duda, H. Gerber, V. Schwachmeyer, S. H. H. Nasab, A. Trepczynski, and I. Kutzner, "A comprehensive assessment of the musculoskeletal system: The CAMS-Knee data set," *Journal of Biomechanics* **65**, 32–39 (2017).

[71] B. Dariush, "Human motion analysis for biomechanics and biomedicine," *Machine Vision and Applications* **14**, 202–205 (2003).

[72] Y.-C. Lin and M. G. Pandy, "Three-dimensional data-tracking dynamic optimization simulations of human locomotion generated by direct collocation," *Journal of Biomechanics* **59**, 1–8 (2017).

[73] S. L. Colyer, M. Evans, D. P. Cosker, and A. I. T. Salo, "A Review of the Evolution of Vision-Based Motion Analysis and the Integration of Advanced Computer Vision Methods Towards Developing a Markerless System," *Sports Medicine - Open* **4**, 24 (2018).

[74] T. J. Hullfish, F. Qu, B. D. Stoeckl, P. M. Gebhard, R. L. Mauck, and J. R. Baxter, "Measuring clinically relevant knee motion with a self-calibrated wearable sensor," *Journal of Biomechanics* **89**, 105–109 (2019).

[75] M. Mundt, W. Thomsen, T. Witter, A. Koeppe, S. David, F. Bamer, W. Potthast, and B. Markert, "Prediction of lower limb joint angles and moments during gait using artificial neural networks," *Medical & Biological Engineering & Computing* **58**, 211–225 (2020).

[76] N. Alexander, H. Schwameder, R. Baker, and U. Trinler, "Effect of different walking speeds on joint and muscle force estimation using AnyBody and OpenSim," *Gait & Posture* **90**, 197–203 (2021).

[77] T. Ino, M. Samukawa, T. Ishida, N. Wada, Y. Koshino, S. Kasahara, and H. Tohyama, "Validity of AI-Based Gait Analysis for Simultaneous Measurement of Bilateral Lower Limb Kinematics Using a Single Video Camera," *Sensors* **23**, 9799 (2023).

[78] B. M. Nigg and W. Herzog, *Biomechanics of the Musculo-Skeletal System*, 2 ed. (Wiley, 1999).

[79] C. J. Payton and R. M. Bartlett, *Biomechanical Evaluation of Movement in Sport and Exercise* (Routledge, 2007).

[80] R. Baker, L. Finney, and J. Orr, "A new approach to determine the hip rotation profile from clinical gait analysis data," *Human Movement Science* **18**, 655–667 (1999).

[81] W. R. Taylor, R. M. Ehrig, G. N. Duda, H. Schell, P. Seebeck, and M. O. Heller, "On the influence of soft tissue coverage in the determination of bone kinematics using skin markers," *Journal of Orthopaedic Research* **23**, 726–734 (2005).

[82] M. P. Kadaba, H. K. Ramakrishnan, and M. E. Wootten, "Measurement of lower extremity kinematics during level walking," *Journal of Orthopaedic Research* **8**, 383–392 (1990).

[83] M. Damsgaard, J. Rasmussen, S. T. Christensen, E. Surma, and M. de Zee, "Analysis of musculoskeletal systems in the AnyBody Modeling System," *Simulation Modelling Practice and Theory* **14**, 1100–1111 (2006).

[84] D. Bakke, P. Ortega-Auriol, and T. Besier, "Shape-model scaling is more robust than linear scaling to marker placement error," *Journal of Biomechanics* **160**, 111805 (2023).

[85] G. E. Gorton, D. A. Hebert, and M. E. Gannotti, "Assessment of the kinematic variability among 12 motion analysis laboratories," *Gait & Posture* **29**, 398–402 (2009).

[86] Z. Ripic, M. Nienhuis, J. F. Signorile, T. M. Best, K. A. Jacobs, and M. Eltoukhy, "A comparison of three-dimensional kinematics between markerless and marker-based motion capture in overground gait," *Journal of Biomechanics* **159**, 111793 (2023).

[87] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A Study of Vicon System Positioning Performance," *Sensors* **17**, 1591 (2017).

[88] F. C. Anderson and M. G. Pandy, "Dynamic optimization of human walking," *Journal of Biomechanical Engineering* **123**, 381–390 (2001).

[89] F. J. Wouda, M. Giuberti, G. Bellusci, E. Maartens, J. Reenalda, B. J. F. van Beijnum, and P. H. Veltink, "Estimation of vertical ground reaction forces and sagittal knee kinematics during running using three inertial sensors," *Frontiers in Physiology* **9** (2018).

[90] N. Zheng and S. W. Barrentine, "Biomechanics and Motion Analysis Applied to Sports," *Physical Medicine and Rehabilitation Clinics of North America* **11**, 309–322 (2000).

[91] I. Poitras, F. Dupuis, M. Bielmann, A. Campeau-Lecours, C. Mercier, L. Bouyer, and J.-S. Roy, "Validity and Reliability of Wearable Sensors for Joint Angle Estimation: A Systematic Review," *Sensors* **19**, 1555 (2019).

[92] M. Paulich, M. Schepers, N. Rudigkeit, and G. Bellusci, "Xsens MTw Awinda: Miniature Wireless Inertial Motion Tracker for Highly Accurate 3D Kinematic Applications," (2018).

[93] K. Saber-Sheikh, E. C. Bryant, C. Glazzard, A. Hamel, and R. Y. Lee, "Feasibility of using inertial sensors to assess human movement," *Manual Therapy* **15**, 122–125 (2010).

[94] P. Slade, A. Habib, J. L. Hicks, and S. L. Delp, "An Open-Source and Wearable System for Measuring 3D Human Motion in Real-Time," *IEEE Transactions on Biomedical Engineering* **69**, 678–688 (2022).

[95] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering* **82**, 35–45 (1960).

[96] H. J. Luinge and P. H. Veltink, "Measuring orientation of human body segments using miniature gyroscopes and accelerometers," *Medical & Biological Engineering & Computing* **43**, 273–282 (2005).

[97] Q. Li and J. T. Zhang, "Post-trial anatomical frame alignment procedure for comparison of 3D joint angle measurement from magnetic/inertial measurement units and camera-based systems," *Physiological Measurement* **35**, 2255–2268 (2014).

[98] K. B. Friesen, Z. Zhang, P. G. Monaghan, G. D. Oliver, and J. A. Roper, "All eyes on you: how researcher presence changes the way you walk," *Scientific Reports* **10** (2020).

[99] L. Tagliapietra, L. Modenese, E. Ceseracciu, C. Mazzà, and M. Reggiani, "Validation of a model-based inverse kinematics approach based on wearable inertial sensors," *Computer Methods in Biomechanics and Biomedical Engineering* **21**, 834–844 (2018).

[100] P. M. S. Ribeiro, A. C. Matos, P. H. Santos, and J. S. Cardoso, "Machine Learning Improvements to Human Motion Tracking with IMUs," *Sensors* **20**, 6383 (2020).

[101] Q. Li, M. Young, V. Naing, and J. Donelan, "Walking speed estimation using a shank-mounted inertial measurement unit," *Journal of Biomechanics* **43**, 1640–1643 (2010).

[102] B. Mariani, C. Hoskovec, S. Rochat, C. Büla, J. Penders, and K. Aminian, "3D gait assessment in young and elderly subjects using foot-worn inertial sensors," *Journal of Biomechanics* **43**, 2999–3006 (2010).

[103] M. A. Borno, J. O'Day, V. Ibarra, J. Dunne, A. Seth, A. Habib, C. Ong, J. Hicks, S. Uhlrich, and S. Delp, "OpenSense: An open-source toolbox for inertial-measurement-unit-based measurement of lower extremity kinematics over long durations," *Journal of NeuroEngineering and Rehabilitation* **19** (2022).

[104] G. Serrancoli, P. Bogatikov, J. P. Huix, A. F. Barbera, A. J. S. Egea, J. T. Ribe, S. Kanaan-Izquierdo, and A. Susin, "Marker-Less Monitoring Protocol to Analyze Biomechanical Joint Metrics During Pedaling," *IEEE Access* **8**, 122782–122790 (2020).

[105] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, "Vision-based and markerless surgical tool detection and tracking: a review of the literature," *Medical Image Analysis* **35**, 633–654 (2017).

[106] F. Crenna, G. B. Rossi, and M. Berardengo, "Filtering Biomechanical Signals in Movement Analysis," *Sensors* **21**, 4580 (2021).

[107] S. S. Tomescu, R. Bakker, T. A. Beach, and N. Chandrashekar, "The Effects of Filter Cutoff Frequency on Musculoskeletal Simulations of High-Impact Movements," *Journal of Applied Biomechanics* **34**, 336–341 (2018).

[108] C. D. Remy and D. G. Thelen, "Optimal Estimation of Dynamically Consistent Kinematics and Kinetics for Forward Dynamic Simulation of Gait," *Journal of Biomechanical Engineering* **131** (2009).

[109] G. Serrancoli, A. Falisse, C. Dembia, J. Vantilt, K. Tanghe, D. Lefeber, I. Jonkers, J. D. Schutter, and F. D. Groote, "Subject-Exoskeleton Contact Model Calibration Leads to Accurate Interaction Force Predictions," *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society* **27**, 1597–1605 (2019).

[110] M. W. Whittle, *Gait Analysis: An Introduction*, 4 ed. (Elsevier, 2007).

[111] R. Drillis, R. Contini, and M. Bluestein, "Body segment parameters; A survey of measurement techniques," *Artificial limbs* **8**, 44–66 (1964).

[112] G. T. Yamaguchi and F. E. Zajac, "A planar model of the knee joint to characterize the knee extensor mechanism," *Journal of Biomechanics* **22**, 1–10 (1989).

[113] Y. Nagano, K. Naito, Y. Saho, S. Torii, T. Ogata, K. Nakazawa, M. Akai, and T. Fukubayashi, "Association between in vivo knee kinematics during gait and the severity of knee osteoarthritis," *Knee* **19**, 628–632 (2012).

[114] J. M. Kaneda, K. A. Seagers, S. D. Uhlrich, J. A. Kolesar, K. A. Thomas, and S. L. Delp, "Can static optimization detect changes in peak medial knee contact forces induced by gait modifications?," *Journal of Biomechanics* **152**, 111569 (2023).

[115] T. A. Correa, R. Baker, H. K. Graham, and M. G. Pandy, "Accuracy of generic musculoskeletal models in predicting the functional roles of muscles in human gait," *Journal of Biomechanics* **44**, 2096–2105 (2011).

[116] J. L. Hicks, T. K. Uchida, A. Seth, A. Rajagopal, and S. L. Delp, "Is My Model Good Enough? Best Practices for Verification and Validation of Musculoskeletal Models and Simulations of Movement," *Journal of Biomechanical Engineering* **137** (2015).

[117] S. A. Roelker, E. J. Caruthers, R. K. Baker, N. C. Pelz, A. M. W. Chaudhari, and R. A. Siston, "Interpreting Musculoskeletal Models and Dynamic Simulations: Causes and Effects of Differences Between Models," *Annals of Biomedical Engineering* **45**, 2635–2647 (2017).

[118] M. M. V. D. Krogt, L. Bar-On, T. Kindt, K. Desloovere, and J. Harlaar, "Neuromusculoskeletal simulation of instrumented contracture and spasticity assessment in children with cerebral palsy," *Journal of NeuroEngineering and Rehabilitation* **13**, 1–11 (2016).

[119] OpenSim Documentation: How Inverse Kinematics Works, `https://simtk-confluence.stanford.edu:8443/display/OpenSim/How+Inverse+Kinematics+Works` (visited on 2024-01-09).

[120] C. Pizzolato, D. G. Lloyd, M. Sartori, E. Ceseracciu, T. F. Besier, B. J. Fregly, and M. Reggiani, "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," *Journal of Biomechanics* **48**, 3929–3936 (2015).

[121] OpenSim Documentation: How Static Optimization Works, `https://simtk-confluence.stanford.edu:8443/display/OpenSim/How+Static+Optimization+Works` (visited on 2024-02-01).

[122] M. Millard, T. Uchida, A. Seth, and S. L. Delp, "Flexing Computational Muscle: Modeling and Simulation of Musculotendon Dynamics," *Journal of Biomechanical Engineering* **135** (2013).

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

[124] OpenSim Documentation: How CMC Works, `https://opensimconfluence.atlassian.net/wiki/spaces/OpenSim/pages/53089706/How+CMC+Works` (visited on 2024-06-19).

[125] H. J. Kim, J. W. Fernandez, M. Akbarshahi, J. P. Walter, B. J. Fregly, and M. G. Pandy, "Evaluation of predicted knee-joint muscle forces during gait using an instrumented knee implant," *Journal of Orthopaedic Research* **27**, 1326–1331 (2009).

[126] C. Pizzolato, M. Reggiani, L. Modenese, and D. G. Lloyd, "Real-time inverse kinematics and inverse dynamics for lower limb applications using OpenSim," *Computer Methods in Biomechanics and Biomedical Engineering* **20**, 436–445 (2017).

[127] G. Valente, G. Crimi, N. Vanella, E. Schileo, and F. Taddei, "NMSBUILDER: Freeware to create subject-specific musculoskeletal models for OpenSim," *Computer Methods and Programs in Biomedicine* **152**, 85–92 (2017).

[128] N. M. Brisson, A. A. Gatti, P. Damm, G. N. Duda, and M. R. Maly, "Association of machine learning based predictions of medial knee contact force with cartilage loss over 2.5 years in knee osteoarthritis," *Arthritis & Rheumatology* 0-1 (2021).

[129] Y. Saiki, T. Kabata, T. Ojima, Y. Kajino, D. Inoue, T. Ohmori, J. Yoshitani, T. Ueno, Y. Yamamuro, A. Taninaka, T. Kataoka, N. Kubo, S. Hayashi, and H. Tsuchiya, "Reliability and validity of OpenPose for measuring hip-knee-ankle angle in patients with knee osteoarthritis," *Scientific Reports* **13**, 3297 (2023).

[130] S. D. Uhlrich, A. Falisse, Łukasz Kidziński, J. Muccini, M. Ko, A. S. Chaudhari, J. L. Hicks, and S. L. Delp, "OpenCap: Human movement dynamics from smartphone videos," *PLOS Computational Biology* **19**, e1011462 (2023).

[131] D. Floreano and C. Mattiussi, *Bio-inspired artificial intelligence: theories, methods, and technologies* (Massachusetts Institute of Technology, 2008).

[132] A. D. Kulkarni, *Computer Vision and Fuzzy-Neural Systems*, 1 ed. (Prentice Hall, 2001).

[133] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing* **151**, 107398 (2021).

[134] N. Buduma and N. Lacascio, *Fundamentals of Deep Learning*, 1 ed. (O'Reilly Media, 2017).

[135] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).

[136] K.-L. Du and M. N. S. Swamy, *Neural Networks and Statistical Learning* (Springer London, 2019).

[137] T. Qamar and N. Z. Bawany, "Understanding the black-box: towards interpretable and reliable deep learning models," *PeerJ Computer Science* **9**, e1629 (2023).

[138] S. Raschka, Y. H. Liu, and V. Mirjalili, *Machine Learning with PyTorch and Scikit-Learn* (Packt Publishing Ltd, 2022).

[139] P. D. Wasserman, *Neural computing: theory and practice* (Van Nostrand Reinhold, 1989).

[140] M. Sewak, R. Karim, and P. Pujari, *Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python* (Packt Publishing, 2018).

[141] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)* 2980-2988 (2017).

[142] J. Zhang, Y. Zhao, F. Shone, Z. Li, A. F. Frangi, S. Q. Xie, and Z.-Q. Zhang, "Physics-Informed Deep Learning for Musculoskeletal Modeling: Predicting Muscle Forces and Joint Kinematics From Surface EMG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **31**, 484–493 (2023).

[143] J. Tohka and M. van Gils, "Evaluation of machine learning algorithms for health and wellness applications: A tutorial," *Computers in Biology and Medicine* **132**, 104324 (2021).

[144] A. Badiola-Bengoa and A. Mendez-Zorrilla, "A Systematic Review of the Application of Camera-Based Human Pose Estimation in the Field of Sport and Physical Exercise," *Sensors* **21**, 5996 (2021).

[145] D. Pagnon, M. Domalain, and L. Reveret, “Pose2Sim: An End-to-End Workflow for 3D Markerless Sports Kinematics—Part 1: Robustness,” *Sensors* **21**, 6530 (2021).

[146] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” *2009 IEEE Conference on Computer Vision and Pattern Recognition* 248-255 (2009).

[147] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common Objects in Context* (Springer, Cham, 2014), pp. 740-755.

[148] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2D Human Pose Estimation: New Benchmark and State of the Art Analysis,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).

[149] N. J. Cronin, T. Rantalainen, J. P. Ahtiainen, E. Hynynen, and B. Waller, “Markerless 2D kinematic analysis of underwater running: A deep learning approach,” *Journal of Biomechanics* **87**, 75–82 (2019).

[150] J. Yang, Y. Ma, X. Zuo, S. Wang, M. Gong, and L. Cheng, “3D pose estimation and future motion prediction from 2D images,” *Pattern Recognition* **124**, 108439 (2022).

[151] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, “BlazePose: On-device Real-time Body Pose tracking,” *arXiv* (2020).

[152] B. Van Hooren, N. Pecasse, K. Meijer, and J. M. N. Essers, “The accuracy of markerless motion capture combined with computer vision techniques for measuring running kinematics,” *Scandinavian Journal of Medicine and Science in Sports* **33**, 966–978 (2023).

[153] C. A. McGibbon, J. Fowler, S. Chase, K. Steeves, J. Landry, and A. Mohamed, “Evaluation of Anatomical and Functional Hip Joint Center Methods: The Effects of Activity Type, Gender, and Proximal Reference Segment,” *Journal of Biomechanical Engineering* **138** (2016).

[154] J. Camargo, A. Ramanathan, W. Flanagan, and A. Young, “A comprehensive, open-source dataset of lower limb biomechanics in multiple conditions of stairs, ramps, and level-ground ambulation and transitions,” *Journal of Biomechanics* **119**, 110320 (2021).

[155] C. A. Fukuchi, R. K. Fukuchi, and M. Duarte, “A public dataset of overground and treadmill walking kinematics and kinetics in healthy individuals,” *PeerJ* **2018**, 1–17 (2018).

[156] F. Horst, S. Lapuschkin, W. Samek, K.-R. Müller, and W. I. Schöllhorn, “A public dataset of overground walking kinetics and full-body kinematics in healthy adult individuals,” *Mendeley Data* (2019).

[157] C. Schreiber and F. Moissenet, “A multimodal dataset of human gait at different walking speeds established on injury-free adult participants,” *Scientific Data* **6**, 1–7 (2019).

[158] J. Aaboe, H. Bliddal, S. P. Messier, T. Alkjær, and M. Henriksen, "Effects of an intensive weight loss program on knee joint loading in obese adults with knee osteoarthritis," *Osteoarthritis and Cartilage* **19**, 822–828 (2011).

[159] B. J. Borbély and P. Szolgay, "Real-time inverse kinematics for the upper limb: A model-based algorithm using segment orientations," *BioMedical Engineering Online* **16** (2017).

[160] A. Falisse, S. V. Rossom, J. Gijsbers, F. Steenbrink, B. J. V. Basten, I. Jonkers, A. J. V. D. Bogert, and F. D. Groote, "OpenSim versus human body model: A comparison study for the lower limbs during gait," *Journal of Applied Biomechanics* **34**, 496–502 (2018).

[161] C. Yi, F. Jiang, M. Z. A. Bhuiyan, C. Yang, X. Gao, H. Guo, J. Ma, and S. Su, "Smart healthcare-oriented online prediction of lower-limb kinematics and kinetics based on data-driven neural signal decoding," *Future Generation Computer Systems* **114**, 96–105 (2021).

[162] F. C. Anderson and M. G. Pandy, "A dynamic optimization solution for vertical jumping in three dimensions," *Computer Methods in Biomechanics and Biomedical Engineering* **2**, 201–231 (1999).

[163] S. L. Delp, J. P. Loan, M. G. Hoy, F. E. Zajac, E. L. Topp, and J. M. Rosen, "An Interactive Graphics-Based Model of the Lower Extremity to Study Orthopaedic Surgical Procedures," *IEEE Transactions on Biomedical Engineering* **37**, 757–767 (1990).

[164] S. R. Hamner, A. Seth, and S. L. Delp, "Muscle contributions to propulsion and support during running," *Journal of Biomechanics* **43**, 2709–2716 (2010).

[165] A. Rajagopal, C. L. Dembia, M. S. DeMers, D. D. Delp, J. L. Hicks, and S. L. Delp, "Full-Body Musculoskeletal Model for Muscle-Driven Simulation of Human Gait," *IEEE Transactions on Biomedical Engineering* **63**, 2068–2079 (2016).

[166] Z. F. Lerner, M. S. DeMers, S. L. Delp, and R. C. Browning, "How tibiofemoral alignment and contact locations affect predictions of medial and lateral tibiofemoral contact forces," *Journal of Biomechanics* **48**, 644–650 (2015).

[167] J. P. Folland, T. M. M. Cauley, and A. G. Williams, "Allometric scaling of strength measurements to body size," *European Journal of Applied Physiology* **102**, 739–745 (2008).

[168] R. M. Ehrig, W. R. Taylor, G. N. Duda, and M. O. Heller, "A survey of formal methods for determining the centre of rotation of ball joints," *Journal of Biomechanics* **39**, 2798–2809 (2006).

[169] R. M. Ehrig, W. R. Taylor, G. N. Duda, and M. O. Heller, "A survey of formal methods for determining functional joint axes," *Journal of Biomechanics* **40**, 2150–2157 (2007).

[170] W. Taylor, E. Kornaropoulos, G. Duda, S. Kratzenstein, R. Ehrig, A. Arampatzis, and M. Heller, "Repeatability and reproducibility of OSSCA, a functional approach for assessing the kinematics of the lower limb," *Gait & Posture* **32**, 231–236 (2010).

[171] R. J. Boekesteijn, M. P. van de Ven, L. M. Wilders, P. Bisseling, B. E. Groen, and K. Smulders, "The effect of functional calibration methods on gait kinematics in adolescents with idiopathic rotational deformity of the femur," *Clinical Biomechanics* **107**, 106028 (2023).

[172] E. B. Simonsen, K. L. Cap-Pelen, R. I. Skorini, T. Alkjaer, K. L. Cappelen, P. K. Larsen, and P. Dyhre-Poulsen, "Explanations Pertaining to the Hip Joint Flexor Moment During the Stance Phase of Human Walking," (2012).

[173] G. Bergmann, A. Bender, F. Graichen, J. Dymke, A. Rohlmann, A. Trepczynski, M. O. Heller, and I. Kutzner, "Standardized loads acting in knee implants," *PLoS ONE* **9** (2014).

[174] H. Huttunen, T. Manninen, and J. Tohka, "MEG mind reading: Strategies for feature selection," *Researchgate.Net* (2012).

[175] O. A. Kannape and O. Blanke, "Self in motion: Sensorimotor and cognitive mechanisms in gait agency," *Journal of Neurophysiology* **110**, 1837–1847 (2013).

[176] X. Liao, J. Zhuang, Z. Liu, J. Dong, K. Song, and J. Xiao, "Reconstructing 3D human pose and shape from a single image and sparse IMUs," *PeerJ Computer Science* **9**, e1401 (2023).

[177] O. Pearl, S. Shin, A. Godura, S. Bergbreiter, and E. Halilaj, "Fusion of video and inertial sensing data via dynamic optimization of a biomechanical model," *Journal of Biomechanics* **155**, 111617 (2023).

[178] T. Handa, R. Sahara, K. Yoshizaki, T. Endou, M. Utsunomiya, C. Kuroiwa, and I. Yamamoto, "Examination of Reliability and Validity of Walking Speed, Cadence, Stride Length - Comparison of Measurement with Stopwatch and Three-Dimension Motion Analyzer," *Journal of Physical Therapy Science* **19**, 213–222 (2007).

[179] S. L. Oh, D. Y. Kim, J. H. Bae, H. Jung, and J. Y. Lim, "Comparison of the use of a manual stopwatch and an automatic instrument for measuring 4-m gait speed at the usual walking pace with different starting protocols in older adults," *European Geriatric Medicine* **10**, 747–752 (2019).

[180] A. Rai, "Explainable AI: from black box to glass box," *Journal of the Academy of Marketing Science* **48**, 137–141 (2020).

[181] A. Holzinger, A. Saranti, C. Molnar, P. Biecek, and W. Samek, *Explainable AI Methods - A Brief Overview* (Springer, Cham, 2022), pp. 13-38.

[182] J. Amann, A. Blasimme, E. Vayena, D. Frey, and V. I. Madai, "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective," *BMC Medical Informatics and Decision Making* **20**, 310 (2020).

[183] D. Slijepcevic, F. Horst, S. Lapuschkin, B. Horsak, A.-M. Raberger, A. Kranzl, W. Samek, C. Breiteneder, W. I. Schöllhorn, and M. Zeppelzauer, "Explaining Machine Learning Models for Clinical Gait Analysis," *ACM Transactions on Computing for Healthcare* **3**, 1–27 (2022).

[184] A. Mohammadi, K. A. Myller, P. Tanska, J. Hirvasniemi, S. Saarakkala, J. Töyräs, R. K. Korhonen, and M. E. Mononen, "Rapid CT-based Estimation of Articular Cartilage Biomechanics in the Knee Joint Without Cartilage Segmentation," *Annals of Biomedical Engineering* **48**, 2965–2975 (2020).

[185] S. Jahangir, A. Mohammadi, M. E. Mononen, J. Hirvasniemi, J. S. Suomalainen, S. Saarakkala, R. K. Korhonen, and P. Tanska, "Rapid X-Ray-Based 3-D Finite Element Modeling of Medial Knee Joint Cartilage Biomechanics During Walking," *Annals of Biomedical Engineering* (2022).

Paper **I**

J. Lavikainen, P. Vartiainen, L. Stenroth, and P. A.
Karjalainen
"Open-source software library for
real-time inertial measurement data-based
inverse kinematics using OpenSim"
PeerJ **11:e15097**,

2023.

DOI: 10.7717/peerj.15097

Open-source software library for real-time inertial measurement unit data-based inverse kinematics using OpenSim

Jere Lavikainen, Paavo Vartiainen, Lauri Stenroth and
Pasi A. Karjalainen

Department of Technical Physics, University of Eastern Finland, Kuopio, Finland

ABSTRACT

Background: Inertial measurements (IMUs) facilitate the measurement of human motion outside the motion laboratory. A commonly used open-source software for musculoskeletal simulation and analysis of human motion, OpenSim, includes a tool to enable kinematics analysis of IMU data. However, it only enables offline analysis, *i.e.*, analysis after the data has been collected. Extending OpenSim's functionality to allow real-time kinematics analysis would allow real-time feedback for the subject during the measurement session and has uses in *e.g.*, rehabilitation, robotics, and ergonomics.

Methods: We developed an open-source software library for real-time inverse kinematics (IK) analysis of IMU data using OpenSim. The software library reads data from IMUs and uses multithreading for concurrent calculation of IK. Its operation delays and throughputs were measured with a varying number of IMUs and parallel computing IK threads using two different musculoskeletal models, one a lower-body and torso model and the other a full-body model. We published the code under an open-source license on GitHub.

Results: A standard desktop computer calculated full-body inverse kinematics from treadmill walking at 1.5 m/s with data from 12 IMUs in real-time with a mean delay below 55 ms and reached a throughput of more than 90 samples per second. A laptop computer had similar delays and reached a throughput above 60 samples per second with treadmill walking. Minimal walking kinematics, motion of lower extremities and torso, were calculated from treadmill walking data in real-time with a throughput of 130 samples per second on the laptop and 180 samples per second on the desktop computer, with approximately half the delay of full-body kinematics.

Conclusions: The software library enabled real-time inverse kinematical analysis with different numbers of IMUs and customizable musculoskeletal models. The performance results show that subject-specific full-body motion analysis is feasible in real-time, while a laptop computer and IMUs allowed the use of the method outside the motion laboratory.

Subjects Bioinformatics, Anatomy and Physiology, Kinesiology, Biomechanics, Rehabilitation
Keywords OpenSim, Real-time, Inverse kinematics, Inertial measurement unit, Motion analysis

Submitted 25 January 2022
Accepted 28 February 2023
Published 5 April 2023

Corresponding author
Jere Lavikainen,
jere.lavikainen@uef.fi

Academic editor
Antonie van den Bogert

Additional Information and Declarations can be found on page 17

DOI 10.7717/peerj.15097

How to cite this article Lavikainen J, Vartiainen P, Stenroth L, Karjalainen PA. 2023. Open-source software library for real-time inertial measurement unit data-based inverse kinematics using OpenSim. **PeerJ 11:e15097 DOI 10.7717/peerj.15097**

INTRODUCTION

Inertial measurement units (IMUs) are measurement devices that contain triaxial magnetometers, gyroscopes, and accelerometers. IMUs used in biomechanics are usually packed into cases that fit on a human palm. They utilize sensor fusion algorithms such as Kalman filters to estimate the three-dimensional orientation of the IMUs in space (*Paulich et al., 2018*). This information can be used as an alternative to marker-based optical motion tracking systems to perform analysis of human movement. Compared with optical motion tracking systems, IMUs are cheaper, can be attached to the subject without the palpation of anatomical landmarks, do not suffer from line-of-sight issues, are not limited to a specific target volume and can be used in field conditions. These advantages come at a small cost of accuracy compared with optical motion tracking systems (*e.g.*, joint angle errors in the lower limbs are generally between 0 and 15 degrees but vary strongly per joint and motion type) (*Poitras et al., 2019*), and IMU-specific error sources such as drifting (*Saber-Sheikh et al., 2010*). In addition, IMUs can be coupled with electromyography (EMG) electrodes to further enhances the versatility of these sensors for analyzing human movement in sports and clinical applications (*e.g.*, Cometa Srl, Cometa Systems | Wireless EMG and IMU Solutions, https://www.cometasystems.com/; Delsys Incorporated, Trigno® Avanti Platform—Delsys, https://www.delsys.com/trigno/; Noraxon USA, Ultium EMG | Noraxon USA, https://www.noraxon.com/our-products/ultium-emg/).

Analysis of kinematics of motion is typically done offline after measurement and data collection in a process called inverse kinematics (IK). Recent years have seen progress in some real-time IK (RTIK) analysis solutions and systems, but these studies (*Bonnet et al., 2013*; *Borbély & Szolgay, 2017*; *Falisse et al., 2018*; *Miezal, Taetz & Bleser, 2017*; *van den Bogert et al., 2013*; *Yi et al., 2021*) have mostly focused on specific marker sets and models and their generalization to arbitrary measurement setups is difficult. For example, in real-time IMU-based applications, *Bonnet et al. (2013)* estimated the RTIK of the trunk and lower limbs using a single IMU located at the lower back. In another example, sensors containing IMUs and EMG electrodes, which were placed on surface musculature, were used in a recent study (*Yi et al., 2021*) to calculate the real-time kinematics and kinetics of the lower limb. However, these studies rely on complex computational methods, making it difficult for others to repeat and adapt the experiment without knowledge of sensor fusion or deep learning. The IMU-based RTIK solution by *Miezal, Taetz & Bleser (2017)* does not allow easy switching between musculoskeletal models or quantify how real-time the solution is.

It has been shown that a full biomechanical analysis of joint and muscle function can be obtained in real-time with a C/C++ software library (*van den Bogert et al., 2013*). The software library was capable of reading marker data and performing IK and inverse dynamics on a full-body model at more than 120 samples per second. However, the software is commercial and relies on a single predefined model rather than a subject-specific or user-defined musculoskeletal model, limiting its usefulness in research. Additionally, *Falisse et al. (2018)* showed that comparing its outputs with those of another similar software (OpenSim 3.3) resulted in statistically significant differences in joint kinematics, kinetics, and muscle forces, highlighting the dependency of the output on the selected model. Hence, it is invaluable that the user can select or generate a model that best fits to the application or research question. The OpenSim software for analyzing the kinematics and dynamics of musculoskeletal systems (*Delp et al., 2007*) offers a solution to the aforementioned issues because it is free and open source, it has a graphical user interface, and it works with customizable musculoskeletal models. While its IK algorithm originally utilized only marker-based motion capture data, since version 4.1 (*Seth et al., 2018*) it has been possible to utilize IMU orientations as input to the IK algorithm of OpenSim to solve skeletal motion (*i.e.*, joint angles). The IMU orientation-based IK algorithm minimizes the sum of squares of the difference between experimental IMU orientations and corresponding segment orientations of the model. Other software with capabilities similar to OpenSim (*Damsgaard et al., 2006*) exist, but the fact that OpenSim is open-source makes it readily available to anyone, enables a variety of community-made modifications and add-ons, and enables the user to view the source code to better understand and troubleshoot the workings of the software. Its customizable models allow the creation of personalized bony geometry, *e.g.*, from imaging data (*Valente et al., 2017*), and the inclusion of muscles in the models allows retrieval of muscle length and similar data that can be used in further analyses. Therefore, although other software may offer kinematics or kinetics based on biomechanical models, enabling real-time analysis with OpenSim has advantages in verifiable and customizable motion analysis and research. While OpenSim has been used for marker-based data to calculate inverse kinematics and inverse dynamics of human motion in real-time (*Pizzolato et al., 2017*), OpenSim-based real-time calculation of IMU-based kinematics with open-source code would enable others to adapt the solution for customizable motion analysis in portable settings.

IMU-based RTIK solutions that utilize the OpenSim API have been recently developed by *Stanev et al. (2021)* and *Slade et al. (2022)*. *Slade et al. (2022)* developed and tested an open-source IMU-based IK system for a microcontroller that can be carried on the subject. Their implementation uses a simplified musculoskeletal model and relaxed IK error tolerance to enable real-time IK with the limited computational power of the microcontroller. With the full computational capacity of the microcontroller (*i.e.*, four threads), they could calculate full-body IK at a throughput of approximately 20 operations per second and at a delay of approximately 200 ms. *Stanev et al. (2021)* have published an open-source software framework that allows kinematical and dynamical analysis of motion. Their software performs the analysis in real-time and supports both IMU- and marker-based data.

The aim of this study was to develop a freely available software library that reads orientation data from IMUs and calculates the IK on a user-given musculoskeletal model in real-time using OpenSim 4.1 API. The development of this work was done prior to the publication of works by *Slade et al. (2022)* and *Stanev et al. (2021)*. Since release of the aforementioned software, our work is not novel but represents an alternative implementation to solve the same problem. The development occurred independently of these software and may therefore provide the community additional value to these previous implementations. To assess the performance of our software we quantified the software's IK execution time and throughput with different numbers of processor threads calculating the IK and different numbers of IMUs, and determined if lowered input data frequency resulting from live visualization meaningfully affects calculated ranges of motion (ROMs).

MATERIALS AND METHODS

Working principles of the software

A software library for reading real-time IMU orientation data as quaternions and processing the data to calculate the IK of a musculoskeletal model was developed using C++ and published on GitHub (https://github.com/jerela/OpenSimLive). The software utilizes OpenSim 4.1 API to invoke methods that calibrate the musculoskeletal model and perform the IK to solve joint angles using quaternion-based orientation data from live measurements.

For each time point, the IK algorithm of OpenSim uses orientation information from all IMUs to find the poses of individual bodies of the musculoskeletal model that, in the least squares sense, minimize the error between experimental IMU orientations and the orientations of the corresponding bodies. For information about the mathematics behind the IK algorithm, see *Delp et al. (2007)*, where it is explained for markers instead of IMUs; for IMUs, IMU orientation error is minimized instead of marker coordinate error. OpenSim's IK algorithm for IMU data is briefly presented in Supplemental Materials, although we did not modify it.

The software library supports Xsens MTw Awinda$^{\text{TM}}$ (Xsens Technologies B.V., Enschede, Netherlands) and Delsys Trigno® (Delsys Inc., Natick, Massachusetts, USA) IMUs. The open-source nature of the software library allows others to add support for other devices. IMU orientations are received wirelessly as quaternions using the Xsens Device API or individual quaternion elements are read from a byte stream *via* socket communication sent by Trigno Control Utility. Information about which IMU corresponds to which body on the musculoskeletal model is read from an XML file. Instead of reading orientation data from the actual IMUs, an option to generate randomized quaternion orientations for testing purposes without IMUs is available.

The orientation information from IMUs is combined in a time series table that contains only one sample, *i.e.*, time point. The time series table is given to OpenSim's IK solver object, which solves the IK for that time point. The process is repeated for each sample. The resulting time series of joint angles can be saved in a text file in .mot format, which allows the output to be viewed using the OpenSim graphical user interface. The read quaternions, and also EMG time series in the case of Trigno® Avanti sensors, can be saved in a text file for later offline analysis.

The working principle of the software library is straightforward. Producer-consumer thread synchronization is used to get orientation data from IMUs. A producer thread and a consumer thread run concurrently. The producer reads orientation data as quaternions from the IMUs and saves it into a buffer that is shared between the threads. The consumer reads and removes data from the buffer and assigns an IK task to a thread in a thread pool. This way there can be several concurrent IK calculations, improving the throughput of the

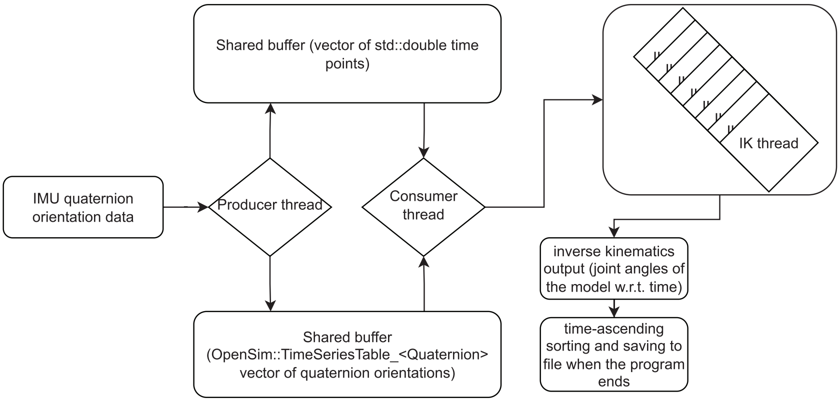

Figure 1 A diagram illustrating the working principle of the inverse kinematics (IK) workflow. Orientation data of inertial measurement units (IMUs) is read as quaternions by the producer thread and saved to a buffer. Time values are saved to another buffer. The consumer thread reads data from both buffers and initiates new threads that calculate IK based on the data. IK threads output joint angle values for the model. Within an IK thread, the IK output can be sent to a visualizer window. The visualization is based on the Simbody (*i.e.*, the physics engine used by OpenSim) API and not a part of our software library and hence not described here in more detail. When the program finishes, the IK output frames can be sorted in a time-ascending order and saved to file. Full-size DOI: 10.7717/peerj.15097/fig-1

program. The maximum number of concurrent IK threads is defined by the user. If that number is already active when the consumer thread starts another IK thread, the consumer thread will wait for one of the IK threads in the thread pool to finish. A diagram illustrating the workflow is shown in Fig. 1.

The software library has been tested to work on 64-bit Windows 7 and Windows 10 operating systems. The source code of the software library is available on GitHub at https://github.com/jerela/OpenSimLive.

Experimental data

Performance tests of the software library and error comparison of joint angles between real-time IK and offline IK were done for walking data of a single subject. The subject was of legal age, had no known musculoskeletal disorders or other conditions that affect gait, and gave their written consent to participate in the study. A total of 12 IMUs (Xsens MTw Awinda, Enschede, Netherlands) were strapped on the subject's upper arms, forearms, chest, pelvis, thighs, shanks, and feet, as shown in Supplemental Information. We recorded 10 trials while the subject walked on an instrumented treadmill (Motek Medical B.V., Amsterdam, Netherlands) at a speed of 1.5 m/s (5.4 km/h) and the IMUs transmitted their orientations at a sample rate of 60 Hz. Each trial contained approximately a minute of gait data. The subject was instructed to take the standard anatomical position at the beginning of each trial to calibrate the IMUs on the musculoskeletal model as per the standard IMU calibration procedure of OpenSim (elaborated in Supplemental Materials). When the subject was in the desired calibration pose, the user of our software library pressed a key to

Table 1 **Conducted performance tests, the parameters that were varied in them, and their purposes.**

Test	Varied parameters	Purpose
Execution times with computer-generated random unit quaternions	Computer, MS model, number of IMUs	Benchmark upper limit of execution times
Execution times with gait data	Computer, MS model, number of IMUs	Quantify execution times in normal use scenario
Throughputs with gait data	Computer, MS model, number of IK threads, number of IMUs	Quantify throughput in normal use scenario

Note:
IMU, inertial measurement unit; MS, musculoskeletal.

calibrate the IMUs. No model scaling was done because we tracked only sensor orientations which are independent of model dimensions.

RTIK was calculated during the measurements. The Simbody visualization of RTIK was enabled, which reduced the throughput of RTIK from the 60 Hz sampling frequency of the IMUs to 45 Hz on average on the desktop computer. This RTIK output was saved to file at the end of each trial. Additionally, the time series of received IMU orientations as quaternions were saved to a file at the full 60 Hz sampling frequency after each trial; the time series were used in calculating execution times and throughputs of the software library.

Performance tests

To quantify the performance of the software library, we conducted performance tests using computer-generated IMU orientations and IMU orientations from recorded human walking (Table 1). Two performance measures were quantified: throughput and execution times of IK. The throughput describes how many IK operations are calculated per second on average when the communication between the producer and the consumer threads is included in the operation. The execution times describe the delays that the calculation of IK for a single time frame takes, *i.e.*, how long it takes after retrieving IMU orientations from one time point to retrieve the corresponding joint angles of the musculoskeletal model. Therefore, throughputs are increased by using multiple IK threads but execution times are not. To retrieve time points, the std::chrono::high_resolution_clock class was used in the C++ implementation.

The tests were conducted with two musculoskeletal models, the Gait2392 lower extremities and torso model (23 degrees of freedom, DOFs; referenced from here on as the lower body model) (*Anderson & Pandy, 1999*, *2001*; *Delp et al., 1990*; *Yamaguchi & Zajac, 1989*) and the Hamner full-body model (29 DOFs, referenced from here on as the full-body model) (*Hamner, Seth & Delp, 2010*). In tests involving the lower body model, data from one and seven IMUs were used; with the full-body model, data from one, seven, and 12 IMUs were used (Table 2). All joint angles that were unlocked in the model by default were solved by the IK algorithm of OpenSim, but meaningful results were obtained only for joint angles defined by the available IMU data, such as joint angles between two segments that both had an attached IMU sensor. Finally, the tests were conducted with two computers, a laptop (HP EliteBook 8570w: Windows 10 Education 64-bit, Intel Core i7-3740QM 2.70 GHz 8-CPU processor, 8192 MB RAM) and a desktop (Fujitsu Celsius

Table 2 The MS model segments whose inverse kinematics were calculated during performance tests, presented by the number of IMUs whose data was utilized.

Number of IMUs	Segments described by IMU orientations
1	Pelvis
7	Pelvis, thighs, shanks, feet
12	Pelvis, thighs, shanks, feet, torso, upper arms, lower arms

Note:
IMU, inertial measurement unit; MS, musculoskeletal.

W550 Power: Windows 10 Education 64-bit, Intel Core i7-6700 3.40 GHz 8-CPU processor, 32768 MB RAM). The laptop computer was used to examine if the software can perform sufficiently using portable devices that allow the measurements to be performed outside a laboratory environment. The desktop computer was used to examine how large difference the increased computing power of a typical desktop compared to a typical laptop makes on the performance of the software.

Execution times with computer-generated IMU orientations were measured with random unit quaternions. The quaternions represented random and sometimes unrealistic poses, which varied greatly between time frames and were time-consuming to calculate. Therefore, the measured execution times represent performance that is worse than in normal human motion, *i.e.*, they were a benchmark of unrealistically poor performance. The execution times were measured with data from different numbers of IMUs (one and seven for the lower body model and one, seven, and 12 for the full-body model). Each execution time measurement lasted until 10,000 IK operations were calculated. We report the mean, standard deviation and 95% confidence interval of the 10,000 execution times.

Finally, IMU orientations from real human walking were used to calculate throughputs and execution times. Performance was measured with pre-recorded quaternion orientations from 10 one-minute walking trials. Although the throughputs and execution times were calculated after the walking trials had been recorded, the performance tests were designed to simulate real-time measurement by feeding the quaternion data into the test environment one data frame at a time. The use of pre-recorded quaternion orientations enabled measuring throughputs above the sampling frequency of the IMUs; otherwise, the throughput would be limited to the sampling rate of the IMUs because IK operations could only be solved at the rate the IMU orientations are received. The tests were conducted with different numbers of IMUs (one and seven for the lower body model and one, seven, and 12 for the full-body model). Furthermore, different numbers of IK threads (one, two, four, six, eight) were used during throughput tests. The performance tests on real walking data allowed us to evaluate performance during a common human motion measurement.

Error comparison of joint angles

Our software library includes an option to visualize motion like in OpenSim GUI by invoking methods from the Simbody API (namely, from the SimTK::Visualizer class), which OpenSim API is built upon. The use of the Simbody visualizer in a real-time IK

thread slows the thread noticeably, and if the throughput drops below the sampling frequency of the IMUs, the IK threads skip some orientation frames from the IMUs in a real-time measurement. This frame drop may negatively affect the accuracy of gait parameters derived from the IK solution. Although not a core feature of our software library, the Simbody visualization during RTIK may be of interest of some. Thus, we evaluated the effect of the frame drops on a kinematic variable that is often of interest and potentially affected by the frame drops, namely the ranges of motion (ROMs) of the joint angles. To this end we compared ROM between visualized RTIK (calculated in this case at approximately 45 Hz) and offline IK at the 60 Hz sample rate of the IMUs. Offline IK was calculated from IMU orientations that were stored after each of the 10 walking trials; RTIK with Simbody visualization was calculated on the desktop computer during each walking trial. Orientations from all 12 IMUs were included, and the full-body model was used in the analysis. The measured motion exerted 26 of the model's DOFs. For determining ROMs, the IK data was divided into periods based on the cyclical nature of the flexion-extension angle of the right knee. For each of the resulting 715 gait cycles (total from 10 walking trials), the difference between the highest and the lowest value of each joint angle was taken as its ROM. The results were reported as the mean absolute error (MAE) and 95% confidence interval (95% CI) between offline IK and RTIK ROM. Note that we calculated errors to evaluate the effect of visualization on ranges of motion and to demonstrate how the software library works, not to validate IMU-based IK.
For IMU-based IK validation, see other studies such as (*Al Borno et al., 2022*; *Tagliapietra et al., 2018*).

RESULTS

Performance tests

Performance test results showed that increasing model complexity and the number of IMUs for orientation tracking increased execution times (Fig. 2) and decreased IK throughput (Fig. 3). Increasing the number of IK calculating threads increased throughput (Fig. 3). The desktop computer always had lower execution times and higher throughput than the laptop computer in the same performance tests (Figs. 2 and 3).

Using computer-generated random unit quaternions, the means and standard deviations of execution times (operation delays) increased with increasing number of IMUs similarly on both the desktop and the laptop (Table 3, Fig. 4). With one IMU, the full-body model was 60–65% slower than the lower body model and had more variation in execution times. The mean execution times were approximately 25% longer on the laptop than on the desktop.

Execution times calculated from human walking (Fig. 2) are shorter than the corresponding execution times from computer-generated data (Table 3, Fig. 4) and remain below 55 ms even with 12 IMUs and the full-body model. Performance with real walking data follows the same patterns as with computer-generated data. Execution times increase (Fig. 2) when the number of IMUs or model complexity (number of DOFs) is increased.

Execution times below 30 ms using the laptop and below 25 ms using the desktop were reached with the lower-body model and seven IMUs (Fig. 2). Furthermore, even with the

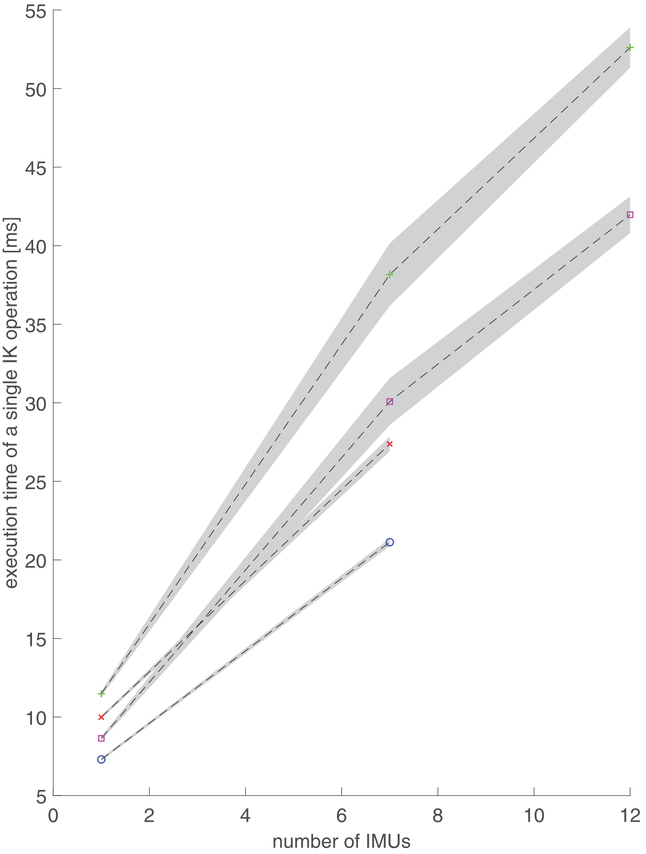

Figure 2 Execution times and standard deviations (shaded area) of a single inverse kinematics operation with respect to the number of inertial measurement units (IMUs). The execution times are presented as mean over 10 trials for two different musculoskeletal models (lower body and torso model and full-body model) using one, seven, or 12 IMUs. IMU quaternions orientations were retrieved from previously recorded walking trials. Red diagonal cross: lower-body model and laptop computer. Green cross: full-body model and laptop computer. Blue circle: lower-body model and desktop computer. Purple square: full-body model and desktop computer. Full-size DOI: 10.7717/peerj.15097/fig-2

laptop, full body kinematics can be calculated with execution times below 60 ms, and provided at least four IK threads are used, with throughputs above 50 Hz (Fig. 3).

Throughput tests show that on the laptop, human walking can be solved at more than 60 IK operations per second when using eight IK threads, the full-body model and 12 IMUs (Fig. 3). The minimal IMU setup to record motion of all lower-body DOFs, seven IMUs with the lower-body model, reached a throughput of 130 with eight IK threads. On the desktop, the corresponding throughputs were 90 and 180, respectively.

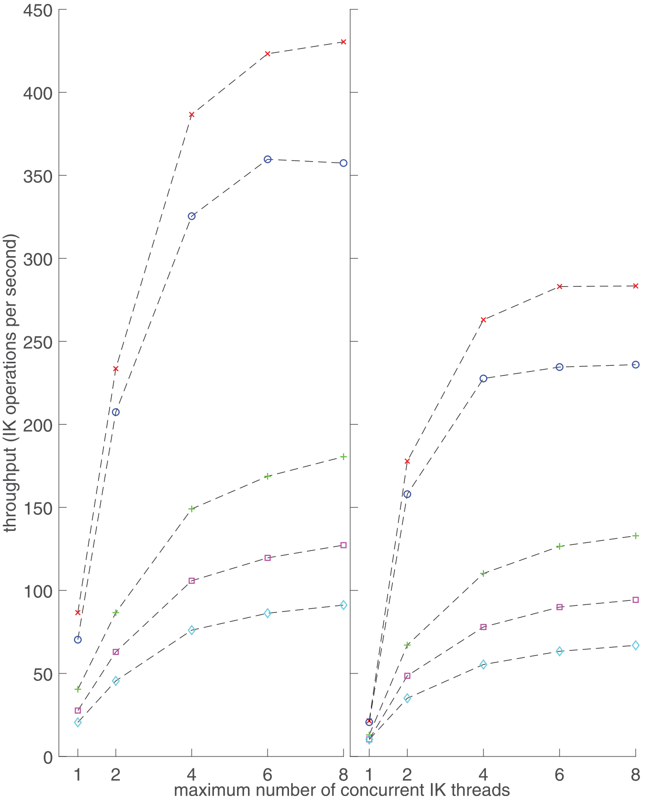

Figure 3 Inverse kinematics (IK) throughput with respect to the number of IK threads used, measured on a desktop computer (left) and a laptop computer (right). The throughputs are presented as mean over 10 trials. The measurements were repeated with two different musculoskeletal models, and using one, seven or 12 inertial measurement units (IMUs). IMU quaternion orientations were retrieved from previously recorded walking trials. Red diagonal cross: lower-body model and one IMU. Green cross: lower-body model and seven IMUs. Blue circle: full-body model and one IMU. Purple square: full-body model and seven IMUs. Cyan diamond: full-body model and 12 IMUs.

Full-size DOI: 10.7717/peerj.15097/fig-3

Table 3 Mean, standard deviation (STD) and 95% confidence interval (CI) of execution times of a single inverse kinematics (IK) operation.

	Lower body, 1 IMU		**Lower body, 7 IMUs**		**Full body, 1 IMU**		**Full body, 7 IMUs**		**Full body, 12 IMUs**	
	Desktop	Laptop	Desktop	Laptop	Desktop	Laptop	Desktop	Laptop	Desktop	Laptop
Mean time (ms)	6.97	8.52	43.55	56.96	11.52	13.77	48.12	64.01	80.42	95.24
STD (ms)	2.31	3.62	22.39	27.19	6.08	7.05	34.03	46.36	29.49	42.00
95% CI (ms)	0.05	0.07	0.44	0.53	0.12	0.14	0.67	0.91	0.58	0.82

Notes:

The values are calculated over 10,000 IK operations for two different musculoskeletal models, two different computers and one, seven, or 12 inertial measurement units (IMUs). Randomly selected unit quaternions were used as IMU orientations.

CI, confidence interval; IK, inverse kinematics; IMU, inertial measurement unit; STD, standard deviation.

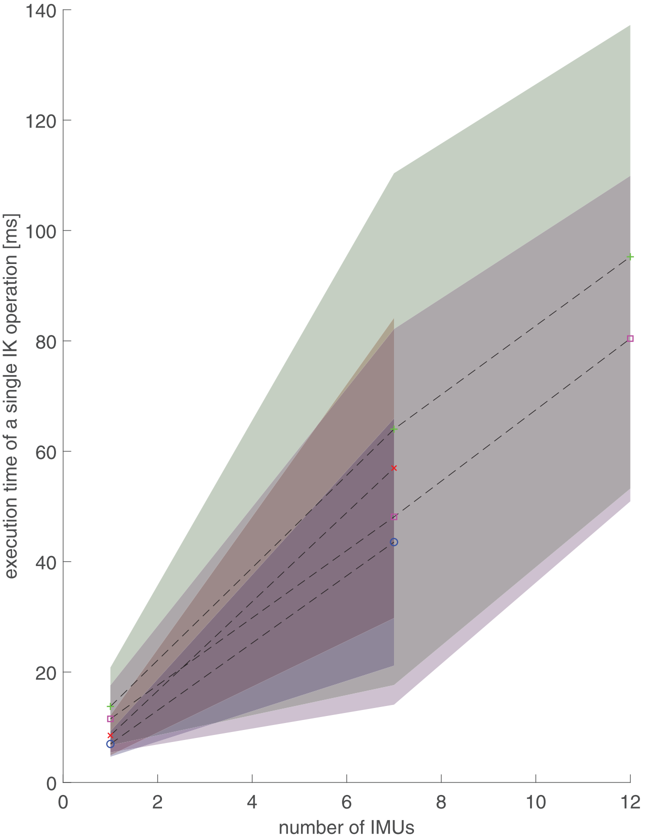

Figure 4 Execution times and standard deviations (shaded area) of a single inverse kinematics operation with respect to the number of inertial measurement units (IMUs), *i.e.*, number of segments with random orientation data. The values are calculated over 10,000 IK operations for two different musculoskeletal models, two different computers and one, seven or 12 inertial measurement units (IMUs). Randomly selected unit quaternions were used as IMU orientations. Red diagonal cross: lower-body model and laptop computer. Green cross: full-body model and laptop computer. Blue circle: lower-body model and desktop computer. Purple square: full-body model and desktop computer.

Full-size DOI: 10.7717/peerj.15097/fig-4

Error comparison of joint angles

Enabling Simbody visualization during the measurement session reduced IK throughput but, compared to IK at full 60 Hz sampling frequency of the IMUs, caused only minimal differences in calculated ranges of motion (Fig. 5). The mean ROM error for all DOFs was 0.0675 degrees. The greatest MAE in ROM was observed in ankle joints (up to 360% of the mean for all joints), followed by the left hip joint (Fig. 5). Pelvis and upper extremities had the smallest ROM error. All MAEs remained below 0.3 degrees.

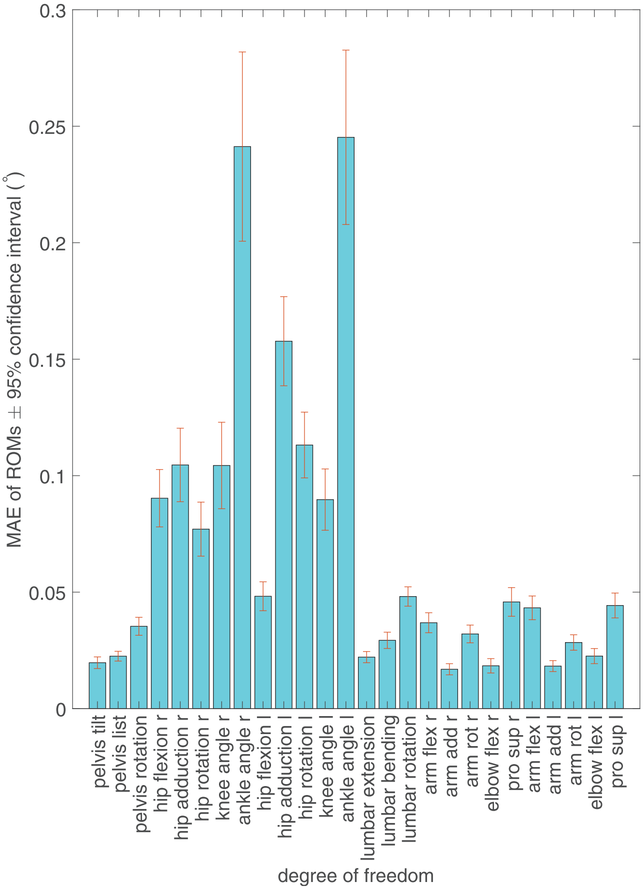

Figure 5 Mean absolute error (MAE) between real-time inverse kinematics and offline inverse kinematics ranges of motion (ROMs) of the exerted degrees of freedom of the musculoskeletal model and the 95% confidence interval of the error. Solid bars show MAEs of ROMs for each exerted degree of freedom. Confidence intervals are shown as error bars centered on the top of the MAE bars.

Full-size DOI: 10.7717/peerj.15097/fig-5

DISCUSSION

We present an open-source software library for the real-time inverse kinematical analysis of IMU data with user-defined musculoskeletal models using OpenSim 4.1. Full-body IK can be calculated for random orientations in less than 100 ms; using real walking data, it can be done in less than 60 ms. On a desktop computer, the software library can solve RTIK at 180 samples per second while tracking the pelvis and lower extremities and at 90 samples per second while tracking the full-body kinematics. On a laptop computer, the corresponding throughputs were 130 and 60 samples per second, respectively. Using 12 IMUs to track walking and visualizing the results on a full-body running model, RTIK was solved at 45 samples per second on a desktop computer. The drop from the IMU output sampling rate of 60 Hz resulted in a minimal difference in calculated joint ROMs (<0.3 degrees). The software library allows the use of RTIK virtually without limitations due to location or environment, which opens possibilities for a variety of applications including rehabilitation, ergonomics, and human-machine interfaces for controlling collaborative robots. Observing the movement of interest in a natural environment is important because a laboratory setting may affect how a person moves (*Friesen et al., 2020*).

Performance tests

We investigated the execution times and throughputs of the IMU-based IK to determine if the output can be considered real-time. *Pizzolato et al. (2017)* used an execution time of 75 ms as the threshold for a real-time system. It was based on a study by *Kannape & Blanke (2013)* in which the subjects were able to identify the displayed motion as self-generated in real-time in over 80% of the cases if the delay in motion display was less than 75 ms. Even with a delay of 210 ms, subjects identified the visualized motion as self-generated in real-time in 50% of the cases. *Borbély & Szolgay (2017)* noted that the IK algorithm of OpenSim 3.3 had an execution time of about 145 ms, thus calculating IK at about 7 Hz and "falling behind the generally accepted practice in human movement recording of at least 50 Hz". Therefore, a real-time application should achieve IK throughput of 50 operations per second with an execution time below 75 ms for any single operation. With our software library, we aimed to achieve this target by using multithreading and the IK algorithm of OpenSim 4.1.

Another interesting finding by *Kannape & Blanke (2013)* was that subjects modulated their stride based on the delay between the motion and its visualization. Therefore, it is important to minimize the delay when preparing a real-time measurement setup to prevent subjects from altering their gait characteristics based on delayed visual feedback.

Live visualization is unnecessary in applications where IK is an intermediate output that is used to estimate contact forces, instruct a robot arm in rehabilitation applications or calculate gait parameters, to name a few examples. Thus, the performance tests were designed so that they evaluate only the performance of IK, which is the core feature of the software library. Although we adopt the visualization-based 75 ms criterion for real-time motion from Kannape and Blanke, our performance tests were conducted without visualization. Our software library relies on Simbody visualization and lacks an elegant visualization solution of its own, which is a limitation that should be acknowledged.

The performance tests show that IK throughput is more sensitive to the number of IMUs than the DOFs of the model, although model complexity also increases computational load because joint angles with no experimental data to solve them are still considered in the IK algorithm. Execution times clearly increased with the number of IMUs, although model complexity also affected them noticeably. Therefore, minimally enough IMUs and DOFs should be chosen to enable high throughput and short execution times in real-time.

The performance of the software library would benefit from improvements particularly in solving IK and visualizing the results. Because a new quaternion time series table is created for a single IMU data point every time an IK frame needs to be solved, the software library is computationally much heavier than it would be if a single time series table could act as a permanent buffer that is updated with new IMU data points. This implementation was not possible using the OpenSim API during the development stage of our software library but appears to be supported in the latest version of the API. Furthermore, the data points in real-time IK may be solved in inconsistent order depending on how quickly each IK thread finishes. Expired IK frames may be omitted if the real-time IK is used as input in further real-time analyses but the missing data points may require interpolation or other consideration, *e.g.*, if filtering the IK is required. Thus, future development of the software library could implement existing interpolation functionalities from the OpenSim API. Additionally, if the visualizer would be implemented in its own thread, it could be used without blocking the IK solving threads and it would enable greater IK throughput while visualization is enabled. This improvement would also enable the solving of joint angles at a high frame rate and visualizing them at a reduced frame rate for performance reasons.

Finally, it should be noted that the software library was only tested on the Windows operating system, and the performance tests were conducted on its operation as a whole. Thus, no proper profiling analysis was done to discover which parts of the software library have potential for performance optimization.

Execution times of the IK operation

Real motion, such as walking, contains a combination of different orientations, most of which are within a typical model's joint angle boundaries. Randomly generated unit quaternions used in execution time tests often result in unrealistic poses. As a result, the IK based on randomized unit quaternions is heavier to calculate than the average orientations during walking, or any typical human motion. Therefore, the execution times from computer-generated data can be interpreted as the worst performance when analyzing human motion without live visualization. Consequently, if the execution times with computer-generated data are sufficiently small for real-time analysis, then any realistic motion should be processed with smaller or equal execution times. On the other hand, we can assume that most human motions can be analyzed with execution times like those from real walking data because joints angles are likely to change at a similar rate and exert constraints similarly.

For both models using computer-generated random unit quaternions, the standard deviations of the execution times are on the same scale as the mean execution times, implying that there is great variation in the execution time. The randomized nature of the used quaternion orientations is a likely contributor to the high standard deviation, because randomized orientations occasionally lead to strange segment orientation combinations that do not reflect valid human motion and take the IK algorithm a long time to solve. This is further supported by the much smaller standard deviation in real walking data (Fig. 2).

Before conclusions are drawn from the computer-generated random unit quaternion-based execution times, it should be acknowledged that the results varied somewhat on repeated runs with the same parameters. This implies that running the tests more than 10,000 iterations could improve the precision of the results. However, because the purpose of the computer-generated random unit quaternions was to estimate worst-case performance for the hardware of the tested computers and the effects of differing the number of DOFs of the model and segments with IMU orientations, which they show well, we chose not to repeat the tests with more iterations.

For both computer-generated (Table 3) and real walking data (Fig. 2), with one and seven IMUs, the execution times are shorter and vary less for the lower body model than for the full-body model. Both the mean execution times and the standard deviations are smaller on the desktop than on the laptop. However, the execution times vary less with 12 than with seven IMUs on the full-body model.

Because execution times calculated on real walking data remained below 55 ms (Fig. 2), the software library is capable of real-time inverse kinematics analysis of the full body even on a laptop. Using computer-generated random unit quaternions, the 95% confidence intervals of execution times are roughly 1% of the mean execution time in all cases, meaning that the execution times stay consistently below 75 ms except when 12 IMUs are used. In that case, the execution times stay consistently below 100 ms, which is still less than half of the 210 ms delay that marks 50% confidence in perceiving motion as real-time (*Kannape & Blanke, 2013*). Therefore, while RTIK is clearly possible with normal walking, some complicated motions may result in longer execution times but could nonetheless be analyzed practically in real-time. Because the execution times represent the minimum delay from the orientation data retrieval to the moment we can visualize or further analyze the IK output, the number of IMUs in a real-time measurement should be chosen considering the delays that are acceptable for the application.

Throughputs

Figure 3 shows that increasing the number of concurrent processor threads increased the throughput until about eight threads, which was the maximum CPU core number for both computers. Increasing the number of IK threads further had no meaningful effect on the throughput, which was also observed in an earlier study on RTIK (*Pizzolato et al., 2017*). The throughput plateau resulted from CPU utilization of the computer reaching 100% and is thus hardware dependent.

The increase in throughput by multithreading is especially large with a small number of threads and one IMU. For example, throughput increases from less than 25 to more than 150 when the number of IK threads increases from one to two on the laptop. Doubled computational capacity alone cannot explain the increase in throughput. The effect is also present on the desktop. Furthermore, the relationship between the throughput and the number of IK threads is clearly nonlinear whereas an earlier RTIK study found it almost linear (*Pizzolato et al., 2017*). No explanation for this phenomenon was found, but it should be addressed in the future development of the software library.

For one IMU and four or more concurrent threads, the lower body model with 23 DOFs had approximately 20% higher throughput than the full-body model with 29 DOFs. For seven IMUs, the lower body model throughput was 40% higher than that of the full-body model. Therefore, model selection has a noticeable effect on the performance of RTIK and the model with the smallest sufficient number of DOFs should be chosen to reach maximal RTIK performance.

The software library is clearly capable of calculating IK at a higher rate than the lower limit of 50 Hz named by *Borbély & Szolgay (2017)*, but requires multithreading to reach it with complex musculoskeletal models. For portable real-time gait measurements, a laptop should be able to achieve sufficiently high IK throughput when seven IMUs are used. We reached a throughput of 130 samples per second in such a scenario, although others' results will vary depending on the hardware of the laptop. Nonetheless, a throughput of 130 samples per second should be sufficient for most gait analysis applications.

Error comparison

Because loss of frequency may lead to reduced accuracy in measuring sharp peaks in joint angles, joints where motion direction changes fast are likely to have high ROM error (Fig. 5). During walking, ankle flexion (ankle_angle_r and ankle_angle_l) undergoes fast changes, which explains why its ROM error stands out. However, because all ROM errors remain consistently below 0.3 degrees, the effect of the drop in visualized IK from 60 to 45 Hz on ROM is very small.

The ROM error of left hip adduction stands out because it is visibly higher than that of the right hip. The error is caused by an artifact in IMU signal that caused the left leg to be violently jerked to the right after the left toe-off phase. The artifact is probably caused by the distortion of magnetic fields near the ferromagnetic laboratory hardware, which the left leg was closer to.

Comparison to other solutions for IMU-based RTIK using OpenSim

In the introduction, we briefly presented some existing solutions for real-time IMU-based IK. Two of them, OpenSenseRT (*Slade et al., 2022*) and the solution by *Stanev et al. (2021)*, utilize the OpenSim API like our software library. OpenSenseRT had comparable execution times and throughputs but to optimize performance, they loosened IK solver tolerance and simplified their musculoskeletal model by removing muscles and locking unused joints. Therefore, the performance results are not easily comparable to ours, where many joints remained unlocked despite not having experimental IMU data to solve them uniquely and muscles of the model were left as they are (although muscles are not part of IK calculations, we made preliminary and unreported observations that OpenSim's standard IMU-based IK throughput is slightly lower on a model with muscles compared with a model without muscles). OpenSenseRT is a good solution for IMU-based IK where the Raspberry Pi computing unit is carried with the subject, while our solution relies on a computer separate from the subject, limiting its applicability slightly but allowing higher performance due to numerous options of laptop hardware and requiring less work to use different musculoskeletal models. The solution by *Stanev et al. (2021)* allows the use of different musculoskeletal models easily. They support both marker-based and IMU-based IK and use lower-level API functions to calculate IK quickly. Furthermore, their solution goes beyond IK, enabling even real-time inverse dynamics and joint reaction force analysis. Their software architecture relies on two threads: one to collect orientation from IMUs or marker positions and then perform IK, and another to perform preprocessing and further musculoskeletal analysis in real-time. Our solution has one thread for collecting orientation from IMUs and a user-defined number of threads for IK. Therefore, although the solution by *Stanev et al. (2021)* is superior to ours in terms of number of features, our solution may allow higher throughputs due to the variable number of IK threads that is only limited by computer hardware.

CONCLUSIONS

An open-source software library that builds upon the widely used OpenSim software was developed and published for IMU-based RTIK. It allows the joint angles of any OpenSim-compatible musculoskeletal model to be analyzed in real-time. While another real-time solution was concurrently and independently developed by *Stanev et al. (2021)*, its IK calculation does not utilize multithreading, which may limit its throughput, although its IK calculation relies on lower-level API classes that are faster than those used by the software library developed in this study when a single thread is used. The authors encourage others to contribute to the open-source project. The development of the software library will closely follow the development of OpenSim to utilize its built-in functionality for processing live data. The software library could be utilized in real-time estimation of joint moments, muscle forces, and joint contact forces based only on IMU data. Ground reaction forces and moments and kinematics are required for solving the equations of motion for the musculoskeletal model using inverse dynamics. It has been shown that ground reaction forces and moments can be predicted from IMU-derived kinematics (*Karatsidis et al., 2017*; *Stanev et al., 2021*). Moreover, estimation of muscle forces using optimization techniques uses kinematics and inverse dynamics estimates of joint moments as inputs and estimates of joint contact forces can be derived based on kinematics, inverse dynamics, and muscle forces. Hence, IMUs could be potentially used for the real-time estimation of musculoskeletal dynamics outside the laboratory and implemented in the software library in the future. Another interesting future application is the use of RTIK output together with EMG. Thus, combining IK output with EMG in real-time may provide interesting possibilities for estimating muscle forces and musculoskeletal loading using EMG driven musculoskeletal simulations (*Sartori et al., 2011*), for biofeedback to optimize rehabilitation or ergonomics or for biosignal-based operating systems.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by the European Union (European Regional Development Fund) and the University of Eastern Finland under the projects Human measurement and analysis research and innovation laboratories (HUMEA, project IDs A73200 and A73241) and Digital Technology RDI Environment (Digi Center, project IDs A74338 and A74340). LS received support from the Academy of Finland (#332915). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
European Union.
University of Eastern Finland (HUMEA): A73200 and A73241.
Digital Technology RDI Environment: A74338 and A74340.
Academy of Finland: #332915.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Jere Lavikainen conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Paavo Vartiainen conceived and designed the experiments, performed the experiments, authored or reviewed drafts of the article, and approved the final draft.
- Lauri Stenroth conceived and designed the experiments, authored or reviewed drafts of the article, and approved the final draft.
- Pasi A. Karjalainen conceived and designed the experiments, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The source code of the software is available at GitHub: https://github.com/jerela/OpenSimLive.

The raw measurement data is available at Zenodo: Lavikainen, Jere, Vartiainen, Paavo, Stenroth, Lauri, & Karjalainen, Pasi. (2022). IMU data from walking trials on a treadmill (1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5901448.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj.15097#supplemental-information.

REFERENCES

Al Borno M, O'Day J, Ibarra V, Dunne J, Seth A, Habib A, Ong C, Hicks J, Uhlrich S, Delp S. 2022. OpenSense: an open-source toolbox for inertial-measurement-unit-based measurement of lower extremity kinematics over long durations. *Journal of NeuroEngineering and Rehabilitation* **19(1)**:361 DOI 10.1186/s12984-022-01001-x.

Anderson FC, Pandy MG. 1999. A dynamic optimization solution for vertical jumping in three dimensions. *Computer Methods in Biomechanics and Biomedical Engineering* **2(3)**:201–231 DOI 10.1080/10255849908907988.

Anderson FC, Pandy MG. 2001. Dynamic optimization of human walking. *Journal of Biomechanical Engineering* **123(5)**:381–390 DOI 10.1115/1.1392310.

Bonnet V, Mazza C, Fraisse P, Cappozzo A. 2013. Real-time estimate of body kinematics during a planar squat task using a single inertial measurement unit. *IEEE Transactions on Biomedical Engineering* **60(7)**:1920–1926 DOI 10.1109/TBME.2013.2245131.

Borbély BJ, Szolgay P. 2017. Real-time inverse kinematics for the upper limb: a model-based algorithm using segment orientations. *BioMedical Engineering Online* **16(1)**:413 DOI 10.1186/s12938-016-0291-x.

Damsgaard M, Rasmussen J, Christensen ST, Surma E, de Zee M. 2006. Analysis of musculoskeletal systems in the AnyBody Modeling System. *Simulation Modelling Practice and Theory* **14(8)**:1100–1111 DOI 10.1016/j.simpat.2006.09.001.

Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG. 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering* **54(11)**:1940–1950 DOI 10.1109/TBME.2007.901024.

Delp SL, Loan JP, Hoy MG, Zajac FE, Topp EL, Rosen JM. 1990. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering* **37(8)**:757–767 DOI 10.1109/10.102791.

Falisse A, van Rossom S, Gijsbers J, Steenbrink F, van Basten BJH, Jonkers I, van den Bogert AJ, de Groote F. 2018. OpenSim versus human body model: a comparison study for the lower limbs during gait. *Journal of Applied Biomechanics* **34(6)**:496–502 DOI 10.1123/jab.2017-0156.

Friesen KB, Zhang Z, Monaghan PG, Oliver GD, Roper JA. 2020. All eyes on you: how researcher presence changes the way you walk. *Scientific Reports* **10(1)**:50 DOI 10.1038/s41598-020-73734-5.

Hamner SR, Seth A, Delp SL. 2010. Muscle contributions to propulsion and support during running. *Journal of Biomechanics* **43(14)**:2709–2716 DOI 10.1016/j.jbiomech.2010.06.025.

Kannape OA, Blanke O. 2013. Self in motion: sensorimotor and cognitive mechanisms in gait agency. *Journal of Neurophysiology* **110(8)**:1837–1847 DOI 10.1152/jn.01042.2012.

Karatsidis A, Bellusci G, Schepers HM, de Zee M, Andersen MS, Veltink PH. 2017. Estimation of ground reaction forces and moments during gait using only inertial motion capture. *Sensors* **17(1)**:75 DOI 10.3390/s17010075.

Miezal M, Taetz B, Bleser G. 2017. Real-time inertial lower body kinematics and ground contact estimation at anatomical foot points for agile human locomotion. In: *Proceedings—IEEE International Conference on Robotics and Automation*. Singapore: IEEE, 3256–3263.

Paulich M, Schepers M, Rudigkeit N, Bellusci G. 2018. Xsens MTw Awinda: miniature wireless inertial motion tracker for highly accurate 3D kinematic applications. In Xsens technologies (Issue May).

Pizzolato C, Reggiani M, Modenese L, Lloyd DG. 2017. Real-time inverse kinematics and inverse dynamics for lower limb applications using OpenSim. *Computer Methods in Biomechanics and Biomedical Engineering* **20(4)**:436–445 DOI 10.1080/10255842.2016.1240789.

Poitras I, Dupuis F, Bielmann M, Campeau-Lecours A, Mercier C, Bouyer L, Roy J-S. 2019. Validity and reliability of wearable sensors for joint angle estimation: a systematic review. *Sensors* **19(7)**:1555 DOI 10.3390/s19071555.

Saber-Sheikh K, Bryant EC, Glazzard C, Hamel A, Lee RYW. 2010. Feasibility of using inertial sensors to assess human movement. *Manual Therapy* **15(1)**:122–125 DOI 10.1016/j.math.2009.05.009.

Sartori M, Reggiani M, Lloyd DG, Pagello E. 2011. A neuromusculoskeletal model of the human lower limb: towards EMG-driven actuation of multiple joints in powered orthoses. In: *IEEE International Conference on Rehabilitation Robotics*. Zurich, Switzerland, Piscataway: IEEE.

Seth A, Hicks JL, Uchida TK, Habib A, Dembia CL, Dunne JJ, Ong CF, DeMers MS, Rajagopal A, Millard M, Hamner SR, Arnold EM, Yong JR, Lakshmikanth SK, Sherman MA, Ku JP, Delp SL. 2018. OpenSim: simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLOS Computational Biology* **14(7)**:e1006223 DOI 10.1371/journal.pcbi.1006223.

Slade P, Habib A, Hicks JL, Delp SL. 2022. An open-source and wearable system for measuring 3D human motion in real-time. *IEEE Transactions on Biomedical Engineering* **69(2)**:678–688 DOI 10.1109/TBME.2021.3103201.

Stanev D, Filip K, Bitzas D, Zouras S, Giarmatzis G, Tsaopoulos D, Moustakas K. 2021. Real-time musculoskeletal kinematics and dynamics analysis using marker-and imu-based solutions in rehabilitation. *Sensors* **21(5)**:1–20 DOI 10.3390/s21051804.

Tagliapietra L, Modenese L, Ceseracciu E, Mazzà C, Reggiani M. 2018. Validation of a model-based inverse kinematics approach based on wearable inertial sensors. *Computer Methods in Biomechanics and Biomedical Engineering* **21(16)**:834–844 DOI 10.1080/10255842.2018.1522532.

Valente G, Crimi G, Vanella N, Schileo E, Taddei F. 2017. NMSBUILDER: freeware to create subject-specific musculoskeletal models for OpenSim. *Computer Methods and Programs in Biomedicine* **152**:85–92 DOI 10.1016/j.cmpb.2017.09.012.

van den Bogert AJ, Geijtenbeek T, Even-Zohar O, Steenbrink F, Hardin EC. 2013. A real-time system for biomechanical analysis of human movement and muscle function. *Medical and Biological Engineering and Computing* **51(10)**:1069–1077 DOI 10.1007/s11517-013-1076-z.

Yamaguchi GT, Zajac FE. 1989. A planar model of the knee joint to characterize the knee extensor mechanism. *Journal of Biomechanics* **22(1)**:1–10 DOI 10.1016/0021-9290(89)90179-6.

Yi C, Jiang F, Bhuiyan MZA, Yang C, Gao X, Guo H, Ma J, Su S. 2021. Smart healthcare-oriented online prediction of lower-limb kinematics and kinetics based on data-driven neural signal decoding. *Future Generation Computer Systems* **114(1)**:96–105 DOI 10.1016/j.future.2020.06.015.

Paper **II**

J. Lavikainen, L. Stenroth, T. Alkjær, P. A. Karjalainen, R.
K. Korhonen, and M. E. Mononen
"Prediction of Knee Joint Compartmental
Loading Maxima Utilizing Simple Subject
Characteristics and Neural Networks"
Ann Biomed Eng **51**, 2479–2489

2023.

DOI: 10.1007/s10439-023-03278-y

Annals of Biomedical Engineering
https://doi.org/10.1007/s10439-023-03278-y

BMES BIOMEDICAL ENGINEERING SOCIETY

ORIGINAL ARTICLE

Prediction of Knee Joint Compartmental Loading Maxima Utilizing Simple Subject Characteristics and Neural Networks

Jere Lavikainen[1] · **Lauri Stenroth**[1] · **Tine Alkjær**[2,3] · **Pasi A. Karjalainen**[1] · **Rami K. Korhonen**[1] · **Mika E. Mononen**[1]

Received: 1 November 2022 / Accepted: 7 June 2023

Abstract

Joint loading may affect the development of osteoarthritis, but patient-specific load estimation requires cumbersome motion laboratory equipment. This reliance could be eliminated using artificial neural networks (ANNs) to predict loading from simple input predictors. We used subject-specific musculoskeletal simulations to estimate knee joint contact forces for 290 subjects during over 5000 stance phases of walking and then extracted compartmental and total joint loading maxima from the first and second peaks of the stance phase. We then trained ANN models to predict the loading maxima from predictors that can be measured without motion laboratory equipment (subject mass, height, age, gender, knee abduction-adduction angle, and walking speed). When compared to the target data, our trained models had NRMSEs (RMSEs normalized to the mean of the response variable) between 0.14 and 0.42 and Pearson correlation coefficients between 0.42 and 0.84. The loading maxima were predicted most accurately using the models trained with all predictors. We demonstrated that prediction of knee joint loading maxima may be possible without laboratory-measured motion capture data. This is a promising step in facilitating knee joint loading predictions in simple environments, such as a physician's appointment. In future, the rapid measurement and analysis setup could be utilized to guide patients in rehabilitation to slow development of joint disorders, such as osteoarthritis.

Keywords Knee joint · Contact force · Compartmental loading · Knee osteoarthritis · Gait analysis · Artificial neural networks · OpenSim

Introduction

Walking and other forms of human locomotion put joints, particularly the knee joint, under stress. Excess loading of the knee joint may have adverse effects on the health and pathology of the joint [13]. For instance, loading of the knee joint may affect the development of knee osteoarthritis (KOA), a disease that affects both the tibiofemoral and the patellofemoral knee joints, causes pain and immobilization, and has major healthcare costs worldwide [18, 32]. Although a causal link between knee joint loading (estimated by knee adduction moments and derived measures as well as tibiofemoral compression force) and structural disease progression of KOA cannot be plausibly established

---

Associate Editor Thurmon E. Lockhart oversaw the review of this article.

✉ Jere Lavikainen
jere.lavikainen@uef.fi

Lauri Stenroth
lauri.stenroth@uef.fi

Tine Alkjær
talkjaer@sund.ku.dk

Pasi A. Karjalainen
pasi.karjalainen@uef.fi

Rami K. Korhonen
rami.korhonen@uef.fi

Mika E. Mononen
mika.mononen@uef.fi

[1] Department of Technical Physics, University of Eastern Finland, Kuopio, Finland

[2] Department of Biomedical Sciences, University of Copenhagen, Copenhagen, Denmark

[3] Parker Institute, Bispebjerg and Frederiksberg Hospital, Copenhagen, Denmark

[17], knee joint loading is believed to have an important role in the development and progression of KOA particularly in the context of frontal plane malalignment and excess loading in either compartment of the joint [9, 13]. Personalized knee joint loading distributions can be estimated by knee joint contact forces, which in turn can be solved with musculoskeletal (MS) modeling utilizing various motion capture (MOCAP) setups [10]. Because information about knee joint loading can be utilized to instruct better movement strategies, capabilities of providing personalized estimations for the joint loading distributions could enable teaching patients movements that reduce loading on vulnerable parts of the joint. However, the biggest limitation why this is not possible in clinics is due to availability of MOCAP setups. This indicates unmet needs for simpler solutions to be applied in the clinics.

MS modeling requires measuring experimental MOCAP data and several computational analysis steps to retrieve knee joint contact forces (KJCFs). Experimentally measured data includes trajectories of markers placed on the subject's skin and ground reaction forces (GRFs) measured with force plates; it may also include other data, such as electromyographic (EMG) signals [2] or data from inertial measurement units [30]. The placing of the markers requires experience and time. The measurement technologies are generally not portable and thus, necessitate measurements in a motion laboratory. Additionally, the analysis steps required to calculate KJCFs are time-consuming. This cumbersomeness may be acceptable in a research setting but can exclude the calculation of KJCFs from clinical use where patient comfortability and time are limiting factors. Therefore, less cumbersome methods to retrieve KJCFs are required. Machine learning methods such as artificial neural networks (ANNs) present a potential solution.

Predicting KJCFs with effortlessly measurable input data could make the estimation of knee joint loading portable. While some previous studies have used raw MOCAP data such as marker trajectories and GRF signals as the input to predict KJCFs in the medial compartment of the knee joint using ANNs [2, 28], the use of predictors with more intuitive biomechanical connection to knee joint loading could be justified and simplify the experimental measurements. For example, the subject's mass, height, BMI, and walking speed have been used as predictors to predict medial KJCF peaks using ANNs [4, 20]. However, those studies still included joint moments [4] or joint angles [12, 20] among the predictors, meaning that MOCAP data were still required in the prediction.

Clearly, the inconvenience of MOCAP data has been noticed by biomechanics researchers and existing studies show a trend toward dynamics estimation while skipping the time-consuming MS analysis steps [2, 12, 20, 28] and using light measurement setups (e.g., without full marker data) instead of laboratory-grade MOCAP [20, 30]. Eliminating reliance on MOCAP-based input data would be a significant step forward in the prediction of the peaks or entire time series of compartmental KJCFs. If KJCFs could be predicted with sufficient accuracy using demographic and anthropometric data, the biomechanical joint loading-based risk assessment of KOA or similar conditions could be done much faster than is possible when MOCAP is required. However, it should be noted that this would require very large training data sets, which would have to consider all possible variations in subject characteristics (such as age, height, weight, type of knee injury).

In this study, we trained feedforward ANNs to predict the total and compartmental KJCFs peaks using mostly anthropometric and demographic data as the input. The input data comprised subject mass, height, age, gender, knee abduction-adduction angle during static standing, and walking speed. Using the same input data, we also investigated the accuracy of predicting medial force ratios (MFRs), which describe the compartmental distribution of joint loading. We aimed to answer the following research questions:

- How accurately can medial, lateral, and total KJCF loading response, terminal extension, and full-stance peaks be predicted using ANNs without motion capture data?
- How accurately can loading response, terminal extension, and full-stance MFRs be predicted using ANNs without motion capture data?
- What is the inter-trial variability of KJCF peaks, i.e., what is the theoretical upper limit of prediction accuracy?

Materials and Methods

The study workflow is presented in Fig. 1.

Included Datasets

The combined dataset for the study was constructed from five separate original datasets. All original datasets contained information about the height, weight, gender, and age of the participants in addition to lower body marker trajectories and ground reaction forces (GRFs) during level walking.

The first original dataset was the CAROT [1] dataset, which contained participants of both genders who had been diagnosed with osteoarthritis in at least one knee. The pathological leg was used in the analysis. Most participants in the dataset had trials from several measurement sessions, in which case we utilized all viable sessions. The CAROT study was approved by the Scientific Ethical Committees for the Capital Region, Denmark (H-B-2007-088). Additionally,

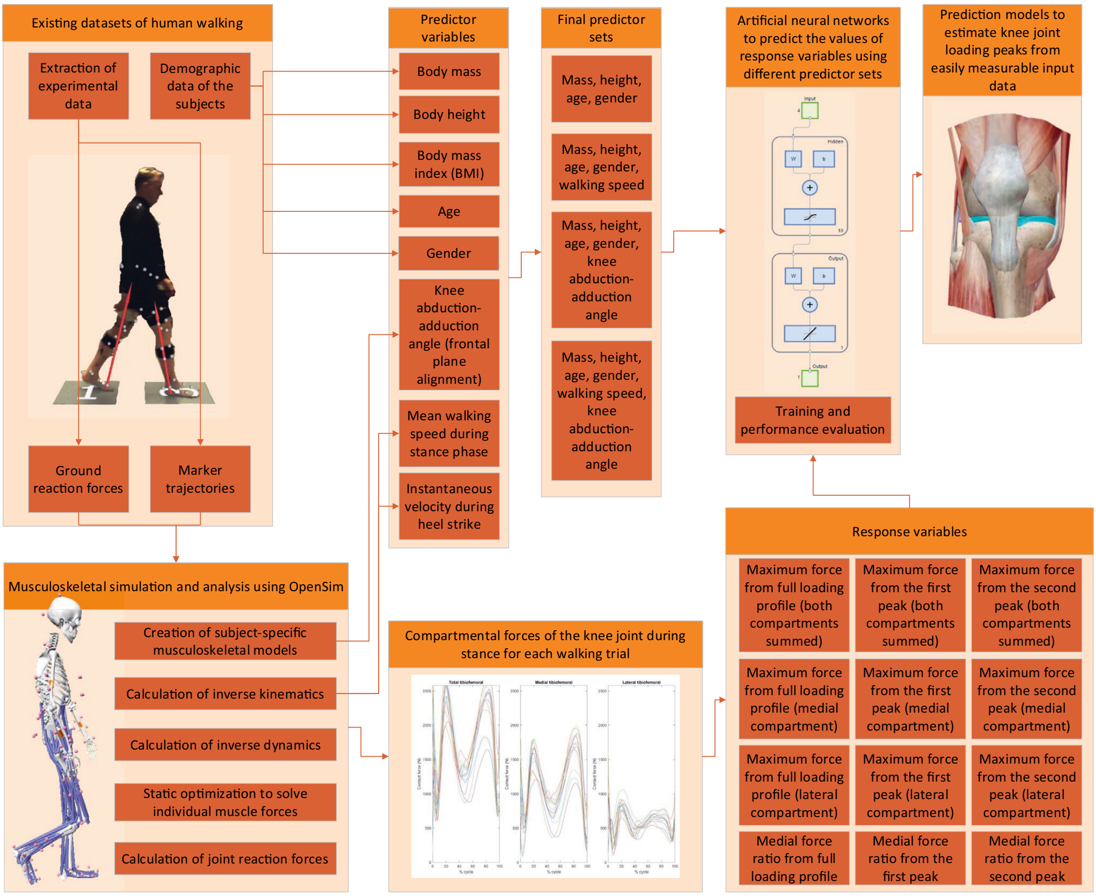

Fig. 1 The workflow consists of extracting data from existing datasets, conducting musculoskeletal simulation and analysis, and constructing artificial neural networks. The data from existing datasets include demographic data and motion capture data. Demographic data are used to formulate most of the predictor variables, while motion capture data are analyzed to retrieve the remaining predictor variables and all response variables. Finally, artificial neural networks are trained for each combination of predictor sets and response variables to predict the value of the response variable from the predictor variables in the predictor set. Detailed information about training the artificial neural network can be found in the Supplementary material

we used data from four open datasets: datasets by Schreiber and Moissenet [29]; Fukuchi et al. [11]; Horst et al. [19]; and Camargo et al. [5]. They comprised healthy participants of both genders and different ages. All open datasets reported informed consent from the participants and ethics approval from an appropriate body.

The combined dataset consisted of 124, 49, 42, 54, and 21 subjects from the CAROT, Schreiber, Fukuchi, Horst, and Camargo datasets, respectively. The combined dataset covers most but not all of the subjects in the original datasets because some subjects had too little valid data (elaborated below in "Estimation Of Joint Loading" section) to be included. Table 1 shows demographic information about the combined dataset and the individual datasets that constitute it. Visual illustrations of the response variables and their dispersion are presented in the Supplementary material.

Estimation of Joint Loading

We used the open-source musculoskeletal (MS) simulation and analysis software, OpenSim [7], to estimate

Table 1 Demographic information about the subjects in the combined dataset, presented per original dataset and in the combined dataset as a whole

Dataset	Age (years)			Mass (kg)			Height (m)			BMI (kg/m^2)			Gender	
	Min	Median	Max	Min	Median	Max	Min	Median	Max	Min	Median	Max	m	f
Camargo	19	21	33	52.2	68.0	96.2	1.52	1.73	1.80	18.6	23.6	30.3	13	8
CAROT	50	62	78	74.0	99.3	145	1.49	1.65	1.91	28.2	35.9	51.0	25	99
Fukuchi	21	33	84	44.9	68.5	95.4	1.47	1.68	1.92	18.0	24.0	33.1	24	18
Horst	19	22	30	47.3	69.0	94.2	1.55	1.77	1.99	18.5	22.2	26.7	28	26
Schreiber	19	38	67	50.0	68.0	98.0	1.55	1.73	1.92	17.2	23.5	29.6	26	23
Combined	19	53	84	44.9	80.8	145	1.47	1.69	1.99	17.2	26.7	51.0	116	174

The presented mass and BMI values for CAROT dataset, where there were several measurement sessions and intra-subject mass and BMI changed between them, represent the values from the first measurement session

knee joint contact forces (KJCF) using the experimental marker trajectories and ground reaction forces (GRF) of the datasets (Fig. 1). The analysis pipeline included the following steps: experimental data extraction, creation of subject-specific musculoskeletal model by scaling the generic model, calculation of inverse kinematics (IK) and inverse dynamics (ID), static optimization (SO) to solve individual muscle forces, and finally the calculation of joint reaction forces. We implemented this pipeline as MATLAB scripts that invoked the OpenSim 4.1 application programming interface. Details of musculoskeletal modeling and simulation, including validation of the selected analysis pipeline, can be found in the Supplementary material.

Before constructing the combined dataset, the estimated KJCF curves were visually checked to exclude trials with potential measurement, modeling, and simulation-related error to ensure validity of the data entering further analyses. Trials were excluded based on non-physiological reasons, such as simulation-related artifacts or technical issues. Exclusion was done on a per-trial basis rather than on a per-subject basis, i.e., presence of excluded trials did not prevent the inclusion of valid trials under the same subject in the combined dataset. Common criteria for exclusion were the presence of only half of the stance phase and sharp distortions that would have been detected as peak values in the following analysis step. Reasons for them included the stance phase-detecting algorithm mistaking the mid-stance local minimum as the end-of-stance minimum in force plate data, jumping marker labels that caused sudden changes in IK, and distortions in force plate data. Overall, 19% of analyzed trials were excluded based on visual inspection of KJCF curves. Although some of these errors could have been corrected by trial-specific manual changes in the data and analysis pipeline, those trials were excluded instead of corrected because fixing them would have taken a lot of time and because of the large amount of valid data we received with the automatic analysis pipeline. Therefore, only trials with visually validated KJCF curves constituted the combined dataset. Extraction of KJCF peak values, calculation of medial force ratios, and dataset-specific MS modeling notes are presented in the Supplementary material.

Predictor Selection

Predictors (i.e., input variables for the prediction model) in the combined dataset were the mass, height, BMI, age, gender, knee angle, walking speed, and heel strike velocity of the subject (Fig. 1). They were selected due to availability and the possibility of simple acquisition, e.g., in a clinical setting. Knee angle was defined as the abduction-adduction angle of the knee during static standing where positive values indicated adduction and negative values indicated abduction. It was determined for each subject during the scaling of the musculoskeletal model. The walking speed was calculated as the mean velocity of MOCAP markers on the pelvis along the walking direction during the stance phase. Similarly, heel strike velocity was the momentary velocity of the pelvis in three-dimensional space between the frame in the marker data where heel strike was detected and the immediately following frame.

To assess predictor collinearity, we calculated variance inflation factors (VIFs) [14] for the predictors. VIF describes the multicollinearity of predictors in a multiple regression model and is a measure of how much explained variance a predictor shares with other predictors. Mass, height, and BMI had VIFs above 20, with walking speed and heel strike velocity both having VIFs above 3. When BMI and heel strike velocity were omitted from the predictor set, the remaining variables had VIFs below 3. This observation confirmed that having BMI, mass, and height or walking speed and heel strike velocity in the predictor

Table 2 Chosen predictor sets and the variables they included

Predictor set	Included predictors
P_A	Mass, height, age, gender
P_B	Mass, height, age, gender, walking speed
P_C	Mass, height, age, gender, knee angle
P_D	Mass, height, age, gender, walking speed, knee angle

Measurement simplicity decreases gradually from P_A to P_D; P_A comprises only demographic and anthropometric parameters, while P_D contains predictors that are possible to measure outside the motion laboratory with a goniometer, measuring tape, and a stopwatch

set resulted in high collinearity. Therefore, we removed BMI and heel strike velocity from the predictor set; we chose the removal of heel strike velocity over walking speed because walking speed is easier to measure. Based on the simplicity of measuring the predictors, we formed four predictor sets P_A, P_B, P_C, and P_D with varying number of predictors (Table 2).

Artificial Neural Network Analysis

MATLAB R2020b (MathWorks, Natick, MA) with deep learning toolbox for the ANN analysis was utilized (Fig. 1). For the architecture of the neural network, we chose a fully connected feedforward network. The ANNs consisted of an input layer, a single hidden layer, and an output layer. In the hidden layer, the activation function, the number of nodes, and the training algorithm were chosen with a hyperparameter optimization algorithm that is described in Supplementary materials. The activation function in the output layer was a linear function.

We used backpropagation training to train ANNs to predict loading response, terminal extension, and full-stance peak on the medial and lateral compartments and the aforementioned peaks identified from the total sum of the time series of both compartments with different predictor sets. Additionally, we trained ANNs to predict MFRs during loading response, terminal extension, and the entire stance phase. Individual trials, and not their ensemble averages, were used in training the ANNs; information about weighting the trials can be found in the Supplementary material. In summary, an ANN was trained for each predictor-output combination, totaling 48 ANNs (four predictor sets times twelve response variables).

During training, all data points in the combined dataset were distributed into three subsets: a training set, a validation set, and a test set. Recommended distributions of data in the sets vary per source but at least half of the whole dataset is usually allocated to the training set [15, 22]; we used fivefold cross-validation to allocate the subsets, explained in detail in the Supplementary material. The data in the training set were used to iteratively modify the weights between the input layer and the hidden layer and the hidden layer and the output layer. MATLAB's predefined training algorithm was used to update the weights iteratively. Formulation of the weights of the training samples is explained in the Supplementary material.

Finally, according to the cross-validation scheme described in the Supplementary material, we determined test error by evaluating the independent test set in the ANN and calculating the RMSE by comparing the predicted output to the true (i.e., MS model estimated) output. To ensure comparability of ANN prediction accuracy between different predictor sets, we used the same random seed of one to initialize the Mersenne Twister pseudorandom number generator of MATLAB for subset division and initial network weight generation for each predictor set.

Analysis of Response Variability

Because of inherent variability in repeated movements, knee loading profiles always differ between trials within a single subject. Therefore, no matter how well-trained and well-structured ANNs are used, response variables in an independent dataset are impossible to predict with perfect accuracy. To determine what the highest attainable accuracy is, within each subject and trial configuration, the inter-trial variability of the response variables was evaluated. For each response variable, we grouped the response values under their respective subjects and, if applicable, further under different trial configurations. The trial configurations included instructed walking speed, measurement session, and other parameters so that each group would ideally result in the same knee loading if there was no measurement noise and the subject could replicate their movement exactly. For each set of responses in a group excluding groups with less than four trials, we calculated the standard deviation. We then calculated the mean and standard deviation of the intra-group standard deviations for each original dataset.

We chose standard deviation as the measure of dispersion (i.e., variability) over other options such as range, interquartile range, and mean absolute deviation because standard deviation is more intuitive to compare against RMSE due to the similarities in their equations. RMSE is calculated as

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left(y_i - t_i\right)^2}{N}},$$

where y_i is the i-th predicted response, t_i is the i-th target value of the response, and N is the number of data points. Standard deviation is calculated as

$$\mathrm{STD} = \sqrt{\frac{\sum_{i=1}^{M} \left(t_i - \mu\right)^2}{M - 1}},$$

where t_i is the i-th true value (i.e., target value) of the response, μ is the sample mean of all t_i, and M is the number of t_is. If we assume that y_i equals μ for all i and the number of data points N in the test subset that is used to calculate RMSE equals $M-1$, RMSE and standard deviation are equal. This assumption fails to hold, but we acknowledged this connection between these two measures and treated standard deviation as an approximation of the minimum attainable RMSE for the ANN.

With standard deviation, we had approximate information of how much the response values vary when measurement conditions remain unchanged. This information provided the context necessary for interpreting the prediction accuracy of the ANNs.

Results

The KJCF peaks resulting from the musculoskeletal (MS) analysis pipeline are available online on Zenodo (https://zenodo.org/record/7253458) [23]. The online dataset also contains the predictors corresponding to the KJCF peaks so that others may use the data in their prediction algorithms. The dataset contains data only from the four open datasets.

Analysis of Response Variability

In Table 3, for all 12 response variables, we present the means and standard deviations of the standard deviations of KJCF peaks over similar trial conditions with four or more trials a group (i.e., measures that describe how much MS modeling estimated loading peaks vary between trials that are identical in terms of subject characteristics and instructed walking speed). The results vary per original dataset and response variable, but overall the standard deviations of the measures are in the same order of magnitude as the measures themselves.

Prediction Accuracy of the Artificial Neural Networks

The best Pearson correlation coefficient (R = 0.84) and root mean square error normalized to the mean of the response variable (NRMSE = 0.14) were reached for the summed peak of both compartments over the full-stance phase with predictor set P_D (Table 4). With predictor sets P_B and P_D, the most accurately predicted response variable was the summed peak of both compartments over the full-stance phase ($R = 0.82$ for P_B and $R = 0.84$ for P_D), whereas with predictors sets P_A and P_C, the summed terminal extension peak was predicted the most accurately (R = 0.74 for $\mathrm{P_A}$ and $R = 0.74$ for P_C).

The worst attained correlation between predicted and MS model estimated responses was for the medial loading response peak with predictor set P_C ($R = 0.42$), and the worst NRMSE was for the lateral loading response peak with predictor set $\mathrm{P_A}$ (NRMSE = 0.42). In both predictor sets, the loading response peaks always performed worse than terminal extension peaks or full-stance peaks.

When walking speed was included in the predictor set, prediction accuracy improved without exception compared with a similar predictor set without walking speed. While compartmental KJCF peaks can be predicted with $R > 0.4$ with just the mass, height, age, and gender of the subject, including the walking speed in the predictors is required for $R > 0.7$.

Although the inclusion of knee abduction-adduction angle in the predictors generally improved the prediction of KJCF peaks in the lateral compartment, its effects on the prediction of medial and summed (non-compartmental)

Table 3 Mean and standard deviation (STD) of the STDs of musculoskeletal model estimated KJCF peaks of all similar trial conditions with at least four trials, presented for each response variable and original dataset

Mean ± STD	Camargo	CAROT	Fukuchi	Horst	Schreiber
Full (summed)	378.5 ± 296.0	216.0 ± 149.6	256.4 ± 153.6	270.1 ± 111.4	171.8 ± 147.6
Full (medial)	271.1 ± 192.3	163.5 ± 94.61	208.5 ± 133.4	213.6 ± 82.5	126.4 ± 95.06
Full (lateral)	272.9 ± 185.7	120.7 ± 91.77	130.6 ± 87.61	90.47 ± 55.66	92.65 ± 65.14
LR (summed)	420.2 ± 272.0	299.1 ± 213.9	317.0 ± 193.7	199.6 ± 112.6	213.2 ± 203.3
LR (medial)	307.2 ± 176.5	200.5 ± 139.4	217.4 ± 139.9	118.2 ± 75.63	144.6 ± 125.7
LR (lateral)	282.4 ± 187.3	131.5 ± 92.55	170.4 ± 88.40	105.3 ± 57.79	131.9 ± 65.85
TE (summed)	340.1 ± 273.5	195.2 ± 124.3	238.9 ± 149.5	280.5 ± 84.39	139.4 ± 90.16
TE (medial)	260.2 ± 205.6	162.3 ± 93.24	212.2 ± 133.5	225.5 ± 69.11	120.1 ± 68.0
TE (lateral)	204.5 ± 120.2	93.48 ± 62.21	100.0 ± 71.2	76.73 ± 25.20	81.27 ± 58.07

“Full” refers to the peaks from the full stance, while LR and TE refer to the loading response (first peak) and terminal extension (second peak) phases of stance, respectively. “Summed” refers to the sum of medial and lateral compartment loading. All values are in newtons

Table 4 Pearson correlation coefficients (R), root mean square errors (RMSE) between the MS model estimated “ground truth” and artificial neural network-predicted response values, and RMSE normalized to the mean of the response variable (NRMSE) ± the standard deviations of R, RMSE, and NRMSE

Response variable	*R*	*RMSE*	*NRMSE*
full (summed)	0.71 ± 0.04	479.4 ± 53.92	0.1874 ± 0.0211
full (medial)	0.62 ± 0.06	389.1 ± 39.84	0.2080 ± 0.0213
full (lateral)	0.58 ± 0.04	252.5 ± 28.89	0.3021 ± 0.0346
LR (summed)	0.54 ± 0.04	602.1 ± 32.25	0.2814 ± 0.0151
LR (medial)	0.55 ± 0.09	379.3 ± 32.91	0.2470 ± 0.0214
LR (lateral)	0.47 ± 0.05	305.7 ± 29.18	0.4233 ± 0.0404
TE (summed)	0.74 ± 0.06	406.7 ± 57.97	0.1664 ± 0.0237
TE (medial)	0.63 ± 0.11	379.0 ± 34.58	0.2109 ± 0.0192
TE (lateral)	0.54 ± 0.11	198.1 ± 37.90	0.2801 ± 0.0536

Response variable	R	RMSE	NRMSE
full (summed)	0.82 ± 0.03	384.2 ± 54.52	0.1502 ± 0.0213
full (medial)	0.75 ± 0.04	319.0 ± 24.18	0.1705 ± 0.0129
full (lateral)	0.67 ± 0.08	238.7 ± 50.25	0.2857 ± 0.0601
LR (summed)	0.75 ± 0.03	481.9 ± 39.37	0.2252 ± 0.0184
LR (medial)	0.70 ± 0.06	320.5 ± 29.71	0.2087 ± 0.0193
LR (lateral)	0.67 ± 0.04	262.8 ± 20.66	0.3639 ± 0.0286
TE (summed)	0.80 ± 0.04	368.4 ± 52.32	0.1507 ± 0.0214
TE (medial)	0.68 ± 0.04	355.1 ± 21.08	0.1976 ± 0.0117
TE (lateral)	0.63 ± 0.07	183.6 ± 25.49	0.2597 ± 0.0361

Response variable	R	RMSE	NRMSE
full (summed)	0.71 ± 0.04	470.2 ± 56.07	0.1838 ± 0.0219
full (medial)	0.60 ± 0.22	381.2 ± 59.06	0.2038 ± 0.0316
full (lateral)	0.59 ± 0.02	251.9 ± 26.74	0.3014 ± 0.0320
LR (summed)	0.54 ± 0.06	600.7 ± 36.71	0.2808 ± 0.0172
LR (medial)	0.42 ± 0.21	403.4 ± 32.18	0.2627 ± 0.0210
LR (lateral)	0.50 ± 0.07	302.2 ± 27.18	0.4184 ± 0.0376
TE (summed)	0.74 ± 0.06	405.9 ± 60.92	0.1661 ± 0.0249
TE (medial)	0.65 ± 0.07	372.7 ± 27.35	0.2074 ± 0.0152
TE (lateral)	0.62 ± 0.11	183.5 ± 26.62	0.2596 ± 0.0376

Response variable	R	RMSE	NRMSE
full (summed)	0.84 ± 0.02	370.3 ± 43.71	0.1447 ± 0.0171
full (medial)	0.76 ± 0.04	316.7 ± 24.63	0.1693 ± 0.0132
full (lateral)	0.71 ± 0.02	220.4 ± 23.77	0.2637 ± 0.0284
LR (summed)	0.74 ± 0.04	485.7 ± 51.97	0.2270 ± 0.0243
LR (medial)	0.70 ± 0.02	324.3 ± 12.99	0.2112 ± 0.0085
LR (lateral)	0.69 ± 0.04	252.3 ± 27.79	0.3493 ± 0.0385
TE (summed)	0.80 ± 0.04	365.5 ± 48.58	0.1496 ± 0.0199
TE (medial)	0.71 ± 0.05	345.4 ± 27.09	0.1922 ± 0.0151
TE (lateral)	0.65 ± 0.07	178.4 ± 20.60	0.2524 ± 0.0291

RMSE is in newtons. The results are calculated with predictor set P_A (mass, height, age, gender), P_B (mass, height, age, gender, walking speed), P_C (mass, height, age, gender, knee angle), and P_D (mass, height, age, gender, walking speed, knee angle) in the top left corner (blue), top right corner (green), bottom left corner (orange), and bottom right corner (red), respectively

KJCF peaks were inconclusive, sometimes increasing prediction accuracy and sometimes reducing it. When comparing predictor sets P_B and P_D, the summed loading response peak was the only response variable that failed to improve by the inclusion of knee abduction-adduction angle.

In summary, Table 4 shows that summed (non-compartmental) KJCF peaks were predicted with better accuracy than compartmental peaks. Full-stance peaks and terminal extension peaks were predicted with $R > 0.5$ to $R > 0.7$ and more accurately than loading response peaks. Walking speed was an important predictor of all KJCF peaks, while knee abduction-adduction angle was beneficial in predicting lateral peaks.

No prediction algorithms with meaningful prediction accuracy could be trained to predict MFRs using the predictors in this study, which is why MFR responses are not included in the tables.

Discussion

Using musculoskeletal (MS) modeling, we analyzed motion capture data from 290 subjects and over 5000 walking trials to retrieve the time series of knee joint contact forces (KJCFs) during different parts of the stance phase and separately in the medial and the lateral compartment. We then trained feedforward artificial neural networks (ANNs) to predict KJCF peaks from input data that can be collected without laboratory-grade motion capture (MOCAP). Our ANN models, which have only one hidden layer with relatively few nodes, do not require MOCAP data, and predict only the maxima of joint loading, can in best cases reach Pearson correlation coefficients above 0.8 (Table 4, top left and bottom left). We achieved such results when walking speed was included in the predictors in addition to demographic and anthropometric predictors. While previous studies have also used ANNs or other machine learning algorithms to predict KJCFs [2, 4, 12, 20, 28] and eliminate the time-consuming MS modeling steps, they still utilized varying amounts of MOCAP data. Our results are a promising step toward predicting KJCFs with simple subject characteristics.

During MS analysis, we observed that the typical tibiofemoral KJCF loading curve over a stance phase had two distinct maxima: the loading response peak and the terminal extension peak. The terminal extension peak was often higher than the loading response peak. When identifying response values in such trials, the terminal extension peak was also the full-stance peak, which explains why the full-stance peaks and the terminal extension peaks resulted in similar prediction accuracies (Table 4). When the subject walked slowly, the typical loading curve distorted as especially the loading response peak became flatter, in some cases merging into the ascent toward the terminal extension peak. Consequently, the loading response peaks either varied a lot or could not be identified in many low-speed trials. Therefore, the data available for predicting loading response peaks were less comprehensive and noisier than that for terminal extension and full-stance peaks, and the prediction accuracy of loading response peaks was smaller. The high prediction accuracy of terminal extension and full-stance peaks can be at least partially attributed to this effect, as the terminal extension peaks of the loading profile underwent more predictable changes with changing walking speed. In the context of KOA studies, difficulties in measuring loading response peaks at low walking speeds are undesirable because pain may cause KOA patients to walk slower than healthy subjects. Alternative derived measures of KJCF, such as area under the curve or mean loading, could perhaps be predicted more accurately than peaks for the loading response phase but were outside the scope of our study.

In general, the summed peaks were predicted with better accuracy than the compartmental peaks (Table 4) and this was expected. Because the summed peaks have a greater magnitude than compartmental peaks, absolute prediction errors in summed peaks do not affect the Pearson correlation coefficient as strongly as the same errors would for compartmental peaks.

Because the inclusion of knee abduction-adduction angle had only small impact on prediction accuracy and the changes were often within the standard deviation of the accuracy measure (Table 4), we cannot draw final conclusions about its effect. The angle was measured during static standing, so it seems reasonable to assume it is also present during the stance phase and affects the KJCF distribution between the medial and lateral compartments. Our choice to lock the knee abduction-adduction angle of the subject-specific model during MS analysis to zero (rather than setting it to whatever was estimated for the subject based on static standing trials) reduces the importance of the angle as a predictor of mediolateral load distribution. Our choice was based on validating the method against in vivo data (Fig. S1). However, because the difference between locking the angle in the model to zero or to its estimated value is small, allowing nonzero abduction-adduction angles during MS analysis may be reasonable in future studies.

The inclusion of walking speed in the predictor set improves prediction accuracy for all response variables (Table 4). Intuitively, walking speed should modulate the force impulses the knee joint experiences, so its accuracy-improving effect was expected. The direct proportionality between walking speed and knee joint loading has also been documented in literature; in 2020, Giarmatzis et al. showed that, with increasing walking speed, the loading response and terminal extension peaks increase in both joint compartments [12]. Additionally, Brisson et al. [4] found that walking speed correlated with medial loading response peaks. Furthermore, Bergmann et al. have shown with total knee replacement patients that in vivo joint loading is greater during jogging than walking [3], although in their study there are likely several factors instead of only locomotion speed involved.

Because the RMSEs of our ANNs (Table 4) are in the same scale as the mean ± standard deviation of the standard deviations of response variabilities (Table 3), the ANNs generalize well and without substantial underfitting or overfitting. Therefore, the ANN architecture we used could also be viable in similar prediction studies of biomechanical functions. This observation is important because although the universal approximation theorem states that an ANN can approximate KJCF peaks, there is no guarantee that the hyperparameters we selected can facilitate such a network. In this study, we showed that they can.

Comparison of accuracy measures with previous studies is difficult because of differences in, e.g., predictor variable selection and validation schemes. To our knowledge, no other study has predicted KJCF peaks or KJCFs using our predictors without including GRFs or marker trajectories in the predictor variables. In the context of our study, it is important to note that we trained and evaluated our ANN so that there was no subject overlap between the training, validation, and test subsets (however, overlap of original datasets was allowed). The ANN prediction accuracies of previous studies can reach Pearson correlation coefficients above 0.9 when MOCAP data are included in the predictors [2, 20]. Our ANN models achieved Pearson correlation coefficients above 0.8 without MOCAP data.

Even though our prediction results were promising including time series information of gait in the predictor set could enable the prediction of the entire stance phase KJCF time series, as has been done with many previous studies [2, 20, 28, 30]. Including time series does not necessarily mean returning to a cumbersome motion laboratory because with existing solutions, such as OpenPose [6], a video camera is sufficient to obtain some gait data [26]. An interesting topic of future research is how computer vision can be integrated into our method to include time series data from, for instance, knee flexion-extension angles to enable KJCF time series prediction while retaining the simplicity of the method. Information about the geometry of the knee joint could also be a valuable addition to the predictor set. Although our musculoskeletal analysis pipeline involved subject-specific scaling of the intercondylar distance based on subject height (see supplementary material), no direct information of knee geometry was included in the predictor set. Although such information could improve predictions of compartmental KJCF maxima and subject specificity of the ANNs, collecting the information would require MRI or X-ray scans, which require specialized equipment and operators to obtain. Such equipment is often found in clinical settings and including knee geometry in the predictors is a potential future research direction. We acknowledge that our method is highly simplified because it predicts peaks of KJCF curves rather than the entire curves and relies on predictors that are easily collected rather than being the best variables to predict KJCF.

Assessing the validity of OpenSim-derived KJCF peaks is difficult because different sources have been shown to have greatly varying results with respect to the bodyweight (BW) of the subject [8] and because available in vivo data is limited. After gathering existing studies, Fregly et al. summarized that maximum KJCF peak ranged from 1.8 to 3.0 BW for forces measured in vivo and from 1.8 to 8.1 BW for MS modeling estimates [10], while D'Lima et al. summarized KJCF peaks from 2 to 3 BW for in vivo forces and 1.7 to 7 BW for MS modeling estimates [8]. Additionally, even though we validated our analysis pipeline against in vivo measurements, the in vivo dataset had only six subjects [31]. Nonetheless, Figure S8 shows that with a median of 3.15 BW, 25th percentile of 2.79 BW, and 75th percentile of 3.59 BW, our MS modeling estimates of summed full-stance peaks fit within the range for MS modeling estimates presented by Fregly et al. Therefore, our MS analysis outputs mostly conform to previous literature.

This study was subjected to limitations that need to be addressed. First, the weights for training samples can be formulated in different ways. However, in preliminary training runs of the ANNs, compared to using no training weights (i.e., equal weights for all data samples), our weights set had little effect on the prediction accuracy of the ANNs. We did not report the difference in prediction accuracy with and without weights, so we cannot quantify the role of weights in loading prediction and whether differently formulated weights have a significant effect on knee loading maxima or MFR prediction should be investigated in later studies.

Second, because subjects with masses above 100 kg came only from the CAROT dataset, the prediction results for heavy subjects contain bias from just a single dataset and, therefore, should be evaluated critically. Additionally, it should be noted that subjects in the CAROT dataset were diagnosed with KOA and we ignored the severity of the KOA, which could be important to consider in formulating the training weights.

Third, in addition to mass, limited overlap between original datasets exists in other predictors. These predictors include walking speed (where both the lowest and the highest values are represented by subjects in the Schreiber dataset), knee abduction-adduction angle (CAROT dataset has the largest range but proportionally the least number of values close to zero), and age (particularly in the 40 to 50 range, which is represented only by subjects in the Schreiber dataset). To improve generalization of the prediction models and mitigate dataset-specific bias, we require a lot of overlap in predictor values between the original datasets. In fact, although we had many subjects and samples, due to differences in subject preparation and marker placement, a greater number of different datasets are required to improve generalization.

Fourth, the original datasets analyzed in this study contained barefoot trials, but in daily life much of walking is done while wearing shoes or similar footwear. The influence of footwear on foot biomechanics has been previously studied [33] and GRFs, joint kinematics and joint kinetics have been found to vary between barefoot and shod walking. Therefore, it would be ideal to have datasets with gait data from shod subjects. However, wearing footwear can make it difficult to perform biomechanical analyses, and additionally, footwear type has been shown to affect the biomechanics of gait [33], which would make it difficult to standardize shod gait trials. Thus, barefoot gait trials provide a standardized way to analyze gait biomechanics and we must assume that the same relationships between predictors and knee loading exist in both barefoot and shod walking.

Fifth, if predictors include gait speed and knee alignment that are calculated from markers, then the models using those predictors require motion capture (MOCAP). However, while in this study we calculated those parameters from MOCAP data, MOCAP is not required to obtain them, as gait speed can be approximated with a stopwatch and a tape measure and knee alignment can be measured with a goniometer, to name simple examples (although, e.g., with goniometer accurately locating the hip joint may be difficult). The predictors obtained that way will slightly differ from the predictors that were used to train the models due to the different methods to measure the predictors. For example, Handa et al. [16] validated stopwatch measurements of walking speed to marker-based optical motion capture and found high correlation ($R > 0.9$) between speeds measured with stopwatch and motion capture, although they also noted that the observed speeds were influenced by the operator of the stopwatch. Furthermore, Oh et al. [27] compared stopwatch-based walking speed to an automatic ultrasound-based timing system and found that the stopwatch overestimated walking speed when the subject was already in motion in the beginning of the measurement. Quantifying the error and the correlation between different measurement methods of the same predictor is an interesting topic for future studies. Therefore, we cannot yet quantify how well knee alignment and walking speed measured with simple methods in a clinical setting correlate to our predictors. Additionally, because including frontal plane knee angle in the predictor set had inconclusive effects on prediction accuracy, omitting it from the predictors in future studies is a possibility if it cannot be effortlessly measured.

Sixth, the ANN prediction accuracy results may be biased by the fact that we visually checked MS-estimated KJCF curves and excluded some trials before constructing the combined dataset and training and validating the ANNs. While the trials we excluded did not represent successfully captured or natural walking (because of, e.g., the presence of only one half of stance phase, marker artifacts resulting in unrealistic kinematics), such artifacts sometimes occur during motion capture sessions; therefore, the ANNs are trained on the features present in successful rather than all motion capture trials. Furthermore, the fact that a single person did the visual validation may introduce small bias into the ANNs. However, the reason we excluded erroneous trials in the first place was to avoid bias causing prediction of biomechanically invalid peak values.

In conclusion, we took promising steps toward predicting knee joint loading peaks during gait without requiring measurements in a motion laboratory. This could enable knee joint loading prediction in environments, such as during clinical examination, eliminate time-consuming analysis steps, and enable the operator to immediately view the results. In future, this method may offer a significant improvement for the clinically applicable prediction models of knee osteoarthritis as those models currently rely on generic loading inputs based on the body weights of the subjects [21, 24, 25]. However, it should be noted that optimal ANN models should be trained on larger training datasets, which would consider all possible variations in subject characteristics (such as age, height, weight, type of knee injury).

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10439-023-03278-y.

Acknowledgments The authors would like to thank Academy of Finland Grant no. 332915 (LS), Academy of Finland Grant no. 334773 under the frame of ERA PerMed (RKK), Innovation Fund Denmark Grant no. 9088-00006B under the frame of ERA PerMed (TA), Sigrid Juselius Foundation (MEM), and Academy of Finland Grant nos. 324994, 328920, and 352666 (MEM) for financial support.

Funding Open access funding provided by University of Eastern Finland (UEF) including Kuopio University Hospital.

Declarations

Conflict of interest The author RKK is a founding partner in Aikoa Technologies Oy. The author MEM has stocks or stock options in Aikoa Technologies Oy.

References

1. Aaboe, J., H. Bliddal, S. P. Messier, T. Alkjær, and M. Henriksen. Effects of an intensive weight loss program on knee joint loading in obese adults with knee osteoarthritis. *Osteoarthr. Cartil.* 19:822–828, 2011.

2. Ardestani, M. M., Z. Chen, L. Wang, Q. Lian, Y. Liu, J. He, D. Li, and Z. Jin. Feed forward artificial neural network to predict contact force at medial knee joint: application to gait modification. *Neurocomputing*. 139:114–129, 2014.
3. Bergmann, G., A. Bender, F. Graichen, J. Dymke, A. Rohlmann, A. Trepczynski, M. O. Heller, and I. Kutzner. Standardized loads acting in knee implants. *PLoS ONE*. 9:e86035, 2014.
4. Brisson, N. M., A. A. Gatti, P. Damm, G. N. Duda, and M. R. Maly. Association of machine learning based predictions of medial knee contact force with cartilage loss over 25 years in knee osteoarthritis. *Arthr. Rheumatol.* 2021. https://doi.org/10.1002/art.41735.
5. Camargo, J., A. Ramanathan, W. Flanagan, and A. Young. A comprehensive, open-source dataset of lower limb biomechanics in multiple conditions of stairs, ramps, and level-ground ambulation and transitions. *J. Biomech.* 119:110320, 2021.
6. Cao, Z., G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, 2018. http://arxiv.org/abs/1812.08008
7. Delp, S. L., F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* 54:1940–1950, 2007.
8. D'Lima, D. D., B. J. Fregly, S. Patil, N. Steklov, and C. W. Colwell. Knee joint forces: Prediction, measurement, and significance. *Proc. Inst. Mech. Eng. Part H*. 226:95–102, 2012.
9. Felson, D. T. Osteoarthritis as a disease of mechanics. *Osteoarthr. Cartil.* 21(1):10–15, 2013.
10. Fregly, B. J., T. F. Besier, D. G. Lloyd, S. L. Delp, S. A. Banks, M. G. Pandy, and D. D. D'Lima. Grand challenge competition to predict in vivo knee loads. *J. Orthop. Res.* 30(4):503–513, 2012.
11. Fukuchi, C. A., R. K. Fukuchi, and M. Duarte. A public dataset of overground and treadmill walking kinematics and kinetics in healthy individuals.*PeerJ* 2018:1–17, 2018.
12. Giarmatzis, G., E. I. Zacharaki, and K. Moustakas. Real-time prediction of joint forces by motion capture and machine learning. *Sensors*. 20:1–19, 2020.
13. Glyn-Jones, S., A. J. R. Palmer, R. Agricola, A. J. Price, T. L. Vincent, and H. A. J. Weinans Carr. Osteoarthritis. *The Lancet*. 386:376–387, 2015.
14. Good, P. I., and J. W. Hardin. Common Errors in Statistics (And How to Avoid Them), 4th ed. Hoboken: Wiley, 2012. https://doi.org/10.1002/9781118360125.
15. Goodfellow, I., Y. Bengio, and A. Courville. Deep Learning. New York: MIT Press, 2016.
16. Handa Rpt, T., R. Sahara Rpt, K. Yoshizaki Rpt, T. Endou Rpt, M. Utsunomiya Rpt, C. Kuroiwa Rpt, I. Yamamoto, K. Kyouritsu Hospital, and R. Professor. Examination of Reliability and Validity of Walking Speed, Cadence, Stride Length-Comparison of Measurement with Stopwatch and Three-Dimension Motion Analyzer. 2007.
17. Henriksen, M., M. W. Creaby, H. Lund, C. Juhl, and R. Christensen. Is there a causal link between knee loading and knee osteoarthritis progression? A systematic review and meta-analysis of cohort studies and randomised trials. *BMJ Open*. 4:e005368–e005368, 2014.
18. Hochberg, M. C., M. G. Cisternas, and S. I. Watkins-Castillo. United States Bone and Joint Initiative: The Burden of Musculoskeletal Diseases in the United States (BMUS)at The Burden of Musculoskeletal Diseases in the United States.
19. Horst, F., S. Lapuschkin, W. Samek, K.-R. Müller, and W. I. Schöllhorn. A public dataset of overground walking kinetics and full-body kinematics in healthy adult individuals. *Mendeley Data*. 2019. https://doi.org/10.17632/svx74xcrjr.3.
20. Ii, W. S. B., C. A. Myers, and P. J. Rullkoetter. Machine learning for rapid estimation of lower extremity muscle and joint loading during activities of daily living. *J. Biomech.* 123:110439, 2021. https://doi.org/10.1016/j.jbiomech.2021.110439.
21. Jahangir, S., A. Mohammadi, M. E. Mononen, J. Hirvasniemi, J. S. Suomalainen, S. Saarakkala, R. K. Korhonen, and P. Tanska. Rapid X-ray-based 3-D finite element modeling of medial knee joint cartilage biomechanics during walking. *Ann. Biomed. Eng.* 2022. https://doi.org/10.1007/s10439-022-02941-0.
22. Kwon, S. J. Artificial Neural Networks. Hauppauge: Nova Science Publishers Inc, pp. 1–426, 2011.
23. Lavikainen, J. J., and L. Stenroth. Dataset of knee joint contact force peaks and corresponding subject characteristics from 4 open datasets, 2022. https://doi.org/10.5281/ZENODO.7253458.
24. Mohammadi, A., K. A. H. Myller, P. Tanska, J. Hirvasniemi, S. Saarakkala, J. Töyräs, R. K. Korhonen, and M. E. Mononen. Rapid CT-based estimation of articular cartilage biomechanics in the knee joint without cartilage segmentation. *Ann. Biomed. Eng.* 48:2965–2975, 2020.
25. Mononen, M. E., M. K. Liukkonen, and R. K. Korhonen. Utilizing atlas-based modeling to predict knee joint cartilage degeneration: data from the osteoarthritis initiative. *Ann. Biomed. Eng.* 47:813–825, 2019.
26. Needham, L., M. Evans, D. P. Cosker, L. Wade, P. M. McGuigan, J. L. Bilzon, and S. L. Colyer. The accuracy of several pose estimation methods for 3D joint centre localisation. *Sci. Rep.* 11:20673, 2021.
27. Oh, S. L., D. Y. Kim, J. H. Bae, H. Jung, and J. Y. Lim. Comparison of the use of a manual stopwatch and an automatic instrument for measuring 4-m gait speed at the usual walking pace with different starting protocols in older adults. *Eur. Geriatr. Med.* 10:747–752, 2019.
28. Rane, L., Z. Ding, A. H. McGregor, and A. M. J. Bull. Deep learning for musculoskeletal force prediction. *Ann. Biomed. Eng.* 47:778–789, 2019.
29. Schreiber, C., and F. Moissenet. A multimodal dataset of human gait at different walking speeds established on injury-free adult participants. *Sci. Data*. 6:1–7, 2019.
30. Stetter, B. J., S. Ringhof, F. C. Krafft, S. Sell, and T. Stein. Estimation of knee joint forces in sport movements using wearable sensors and machine learning. *Sensors*. 19:1–12, 2019.
31. Taylor, W. R., P. Schütz, G. Bergmann, R. List, B. Postolka, M. Hitz, J. Dymke, P. Damm, G. Duda, H. Gerber, V. Schwachmeyer, S. H. Hosseini Nasab, A. Trepczynski, and I. Kutzner. A comprehensive assessment of the musculoskeletal system: the CAMS-Knee data set. *J. Biomech.* 65:32–39, 2017.
32. Woolf, A. D., and B. Pfleger. Burden of major musculoskeletal conditions. *Bull. World Health Organ*. 81:646–656, 2003.
33. Zhang, X., M. R. Paquette, and S. Zhang. A comparison of gait biomechanics of flip-flops, sandals, barefoot and shoes. *J. Foot Ankle Res.* 6:1–9, 2013.

JERE LAVIKAINEN

Thorough evaluation of human musculoskeletal movement requires measurements in a motion laboratory, which involves expensive equipment and trained personnel. In this thesis, methods were developed to utilize portable measurement modalities for estimating musculoskeletal modeling outcomes. Artificial neural networks were used to predict knee joint loading directly from simple input variables. These methods could be used for routine estimation of movement and loading in field conditions.

UNIVERSITY OF
EASTERN FINLAND

uef.fi

**PUBLICATIONS OF
THE UNIVERSITY OF EASTERN FINLAND**
Dissertations in Science, Forestry and Technology

ISBN 978-952-61-5275-2
ISSN 2954-131X