VDI

REIHE 10
INFORMATIK/
KOMMUNIKATION

Fortschritt-Berichte VDI

Petrissa Zell, M. Sc.,
Hannover

NR. 877

Learning-Based Inverse Dynamics for Human Motion Analysis

BAND
1|1

VOLUME
1|1

Institut für Informationsverarbeitung
www.tnt.uni-hannover.de

LEARNING-BASED INVERSE DYNAMICS FOR HUMAN MOTION ANALYSIS

Von der Fakultät für Elektrotechnik und Informatik

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

(abgekürzt: Dr.-Ing.)

genehmigte

Dissertation

von Frau

Petrissa Zell, M. Sc.

geboren am 20. März 1989 in Langenhagen

2022

Hauptreferent:	Prof. Dr.-Ing. B. Rosenhahn
Korreferent:	Prof. Dr.-Ing. G. Pons-Moll
Vorsitzender:	Prof. Dr.-Ing. J. Ostermann
Tag der Promotion:	15. Dezember 2021

VDI

REIHE 10
INFORMATIK/
KOMMUNIKATION

Fortschritt-Berichte VDI

Zell, Petrissa, M. Sc.,
Hannover

NR. 877

Learning-Based Inverse Dynamics for Human Motion Analysis

BAND
1|1

VOLUME
1|1

Institut für Informationsverarbeitung
www.tnt.uni-hannover.de

VDI verlag

Zell, Petrissa

Learning-Based Inverse Dynamics for Human Motion Analysis

Fortschritt-Berichte VDI, Reihe 10, Nr. 877. Düsseldorf: VDI Verlag 2022.

160 Seiten, 35 Bilder, 15 Tabellen.

ISBN 987-3-18-387710-2, E-ISBN 978-3-18-687710-9, ISSN 0178-9627,

57,00 EUR/VDI-Mitgliederpreis: 51,30

Für die Dokumentation: inverse Dynamik – maschinelles Lernen – menschliche Bewegung – Gelenkmomente – Ganganalyse – künstliche neuronale Netze – selbstüberwachtes Lernen

Keywords: inverse dynamics – machine learning – human motion – joint moments – gait analysis – artificial neural networks – self-supervised learning

This dissertation deals with machine learning techniques for inverse dynamics of human motion. Inverse dynamics refers to the derivation of acting forces and moments from the motion of a kinematic model. More precisely, the objective is to estimate joint torques, ground reaction forces and ground reaction moments at both feet based on the three-dimensional input motion of a skeletal model. The problem is solved using a data-driven machine learning approach, proposing several regression models that are particularly suitable with respect to limited data availability. The goal is to exploit the inherent strengths of machine learning, such as fast and noise-resistant data analysis. The described methods are able to predict underlying joint torques and exterior forces with high precision (on gait sequences: relative root mean squared errors of 7.0 %, 16.1 % and 11.9 % for reaction forces, reaction moments and joint moments which correspond to Pearson's correlation coefficients of 0.91, 0.83 and 0.82), while reducing computation times by two orders of magnitude compared to traditional optimization.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter www.dnb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek (German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at www.dnb.de.

ISBN 987-3-18-387710-2, E-ISBN 978-3-18-687710-9, ISSN 0178-9627

ACKNOWLEDGEMENTS

This thesis was written during my time at the Institut für Informationsverarbeitung (TNT) of the Gottfried Wilhelm Leibniz Universität Hannover.

First and foremost, I would like to thank my thesis advisor, Prof. Dr.-Ing. Bodo Rosenhahn, for giving me the opportunity to do my doctoral studies under his supervision and for his scientific guidance and support during the entire time. I am very grateful for his motivational and open-minded style of mentoring and of course for the evaluation of my thesis as first examiner. I would also like to thank Prof. Dr.-Ing. Gerard Pons-Moll for the examination of my thesis and for bringing the research field of motion capture to my attention in the first place. I also cordially thank Prof. Dr.-Ing. Jörn Ostermann, who chaired the examination committee and whose assessment I learned to value very much.

I would like to express my sincere thanks to all of my colleagues for numerous constructive discussions and the productive working atmosphere. They made my time at TNT really memorable to me. I especially thank Bastian Wandt, who always supported me with goal-oriented discussions and advice and with whom I enjoyed sharing an office. I would also like to say a special thank you to Thorsten Laude, Felix Kuhnke, the TNT Alpine team and the entire eNIFE for the great time we spent during and outside of work. Moreover, I thank all participants in my data set recordings for their active contribution despite the physically demanding exercises and Matthias Schuh for setting up the measurement equipment. I also like to thank Martin Pahl, Marco Munderloh and Thomas Wehberg for their constant administrative support and Doris Jaspers-Göhring, Melanie Huch and Ellen Sylla for their help in organizational tasks.

I sincerely thank Carolin Scheler for the joint working meetings during the COVID-19 pandemic, which made the lengthy writing process much more pleasant, and for proofreading parts of the thesis. Finally, my deepest gratitude goes to my family and my partner Hendrik Hachmann for their tireless encouragement and motivation and for always having an open ear for my thoughts and concerns.

Dedicated to the loving memory of Jürgen Zell.
1944 – 2017

CONTENTS

ACRONYMS

EOM	Equation Of Motion
GRF	Ground Reaction Force
GRM	Ground Reaction Moment
GRF/M	Ground Reaction Force and Moment
COP	Center Of Pressure
IMU	Inertial Measurement Unit
DOF	Degree Of Freedom
SVM	Support Vector Machine
RF	Random Forest
CART	Classification And Regression Trees
NN	artificial Neural Network
SGD	Stochastic Gradient Descent
ReLU	Rectified Linear Unit
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
rRMSE	relative Root Mean Squared Error
MAE	Mean Absolute Error
CE	Cross Entropy
PDO	Predictive Dynamics Optimization
PD-set	Predictive Dynamics Set
F-net	neural network trained with forward loss
cFI-net	neural network trained with contact, forward and inverse loss
CMU	Carnegie Mellon University

$\boldsymbol{p}$	joint positions
$\boldsymbol{q}$	generalized coordinates
$\boldsymbol{\xi}$	linear and angular body velocities
$\boldsymbol{v}$	linear body velocity
$\boldsymbol{\omega}$	angular body velocity
$\boldsymbol{x}$	kinematic state
$\hat{\boldsymbol{x}}$	kinematic state without global coordinates
$\boldsymbol{x}_s$	segment center of mass positions and velocities
$\boldsymbol{x}_j$	joint positions and velocities
$\boldsymbol{M}$	inertia matrix
$\boldsymbol{\mathcal{M}}$	reduced inertia matrix
$\boldsymbol{\mathcal{F}}$	reduced force
m_i	mass of segment i
$\boldsymbol{I}_i$	tensor of inertia of segment i
$\boldsymbol{f}_c$	concatenation of contact forces and moments
$\boldsymbol{\tau}$	joint torques
$\boldsymbol{u}$	controls (concatenation of $\boldsymbol{f}_c$ and $\boldsymbol{\tau}$)
$\boldsymbol{T}$	Jacobian of kinematic coordinate transformations
$\boldsymbol{T}_{v_i}$	rows of Jacobian transforming to linear velocity of segment i
$\boldsymbol{T}_{\omega_i}$	rows of Jacobian transforming to rotational velocity of segment i
$\boldsymbol{z}_{j-1}$	z-axis of link j
$\boldsymbol{t}_{j-1}$	origin of link j
$\kappa(j)$	support set of link j
$\mu(j)$	subtree set starting at link j
$\nu(i,j)$	subchain set between links i and j
$\boldsymbol{R}_n^0$	rotation matrix from frame n to the global frame
$\boldsymbol{\zeta}$	convective acceleration
$\boldsymbol{g}$	gravitational acceleration

$\boldsymbol{f}_r$	ground reaction force
$\boldsymbol{m}_r$	ground reaction moment
$\boldsymbol{r}_{(\cdot)}$	position vector of point $(\cdot)$
$\boldsymbol{l}$	segment lengths
T	window size (number of frames)
$\boldsymbol{\alpha}_{(\cdot)}$	polynomial coefficients of variable $(\cdot)$
$\boldsymbol{v}_c$	contact point velocity vector
$\boldsymbol{\theta}$	feature vector
l_c	class label of gait phase
c_i	contact state of foot i
$\boldsymbol{d}$	damping in forward layer
$f(), \boldsymbol{f}()$	functions implemented by neural networks
L	loss function
$\epsilon_{(\cdot)}$	relative root mean squared error of variable $(\cdot)$
$\rho_{(\cdot)}$	Pearson's correlation coefficient related to variable $(\cdot)$
e_{EOM}	equation of motion error

ABSTRACT

This dissertation deals with machine learning techniques for inverse dynamics of human motion. Inverse dynamics refers to the derivation of acting forces and moments from the motion of a kinematic model. More precisely, the objective is to estimate joint torques, ground reaction forces and ground reaction moments at both feet based on the three-dimensional input motion of a skeletal model. Of particular interest are the joint torques, also specified as net joint moments, since they correspond to the total effect of all forces on the joints. In the context of biomechanical investigations, they represent a common measure of the load on joints.

Traditional approaches formulate the problem as an optimization that incorporates the equation of motion (EOM) of a physical model of the human body. The EOM is either used in a forward or an inverse sense which implies either integration or differentiation of kinematics. Both processes are prone to error propagation and complicate the convergence of the optimization algorithms built on the formulation. Furthermore, the EOM belonging to a multi-body system, such as the modeled human body, gives rise to a highly non-linear and non-convex objective function which is notoriously hard to optimize. Last but not least, conventional methods generally rely on measured external reaction forces and moments, which severely limits the motions that can be analyzed due to the laboratory environment required.

Given these limitations, data-driven machine learning techniques open up tremendous opportunities by enabling fast and noise-resistant data analysis. This thesis investigates the applicability of such methods to inverse dynamics of human motion and addresses the design of suitable regression models. The proposed methods are able to predict underlying joint torques and exterior forces with high precision (on gait sequences: relative root mean squared errors of 7.0 %, 16.1 % and 11.9 % for reaction forces, reaction moments and joint moments which correspond to Pearson's correlation coefficients of 0.91, 0.83 and 0.82), while reducing computation times by two orders of magnitude compared to traditional optimization.

A general feature of human motion data is the discontinuity at contact phase transitions, e. g. at the moment the foot touches the ground. By changing the number of contact points of the human model to its environment, the set of dynamic equations is fundamentally altered to the extent that external influences are allowed or forbidden at the corresponding points. Motivated by this property, a multi-stage regression approach is presented. The method initially identifies the current gait phase and limits the inference of joint torques

as well as contact forces to the resulting sub-space. This way, the regression of unrealistic non-zero forces during swing phases is significantly reduced compared to a model that estimates the forces without knowledge of the contact state.

Current problems of machine learning methods for solving inverse dynamics are a lack of suitable datasets and that the compliance with the EOM is not guaranteed for the predictions. Both issues are addressed by a self-supervised learning method presented in this thesis. The approach allows cycle consistent training of an artificial neural network with pure motion data, i. e. without any ground truth forces and moments. Instead of minimizing a direct loss on the target forces, the model solves an initial value problem based on predicted forces and minimizes the distance between the resulting simulation and the input motion. This is realized by implementing a differentiable forward dynamics loss layer that allows backward flow of gradients and can be integrated into the training of the neural network. In addition, the model includes a corresponding inverse dynamics layer that evaluates the estimated contact forces decoupled from predicted joint torques. Thus, the model not only allows training on readily available motion data, but also constrains the predicted variables using both dynamic directions for optimal satisfaction of the EOM. The neural network maintains stable performance even with small labeled datasets consisting of dynamics data of only two or three subjects by learning generalization capability on larger unlabeled motion sets. Furthermore, the method enables self-supervised transfer learning to different motion types, movement speeds and skeleton characteristics.

The presented learning-based inverse dynamics approaches are evaluated using a self-recorded dataset of walking and running sequences performed by 22 subjects as well as a public dynamics dataset [39] and gait sequences from the well-known CMU database [18]. The self-recorded dataset is available to the research community.

KURZFASSUNG

Diese Dissertation beschäftigt sich mit maschinellen Lernverfahren für die inverse Dynamik der menschlichen Bewegung. Unter inverser Dynamik versteht man die Ableitung von wirkenden Kräften und Momenten aus der Bewegung eines kinematischen Modells. Genauer gesagt geht es um die Abschätzung von Gelenkmomenten, Bodenreaktionskräften und Bodenreaktionsmomenten an beiden Füßen basierend auf der dreidimensionalen Eingangsbewegung eines Skelettmodells. Von besonderem Interesse sind die Gelenkmomente, die auch als Netto-Gelenkmomente bezeichnet werden, da sie der Gesamtwirkung aller Kräfte an den Gelenken entsprechen. Im Rahmen biomechanischer Untersuchungen stellen sie ein gängiges Maß für die Bean-spruchung von Gelenken dar.

Traditionelle Ansätze formulieren das Problem als eine Optimierung, die die Bewegungsgleichung (Equation of Motion, EOM) eines physikalischen Modells des menschlichen Körpers einbezieht. Die EOM wird entweder in einem vorwärts gerichteten oder einem inversen Sinn verwendet, was entweder eine Integration oder eine Differenzierung der Kinematik impliziert. Beide Verfahren sind anfällig für Fehlerfortpflanzung und erschweren die Konvergenz des Optimierungsalgorithmus. Darüber hinaus führt die zu einem Mehrkörpersystem, wie dem modellierten menschlichen Körper, gehörende EOM, zu einer hochgradig nichtlinearen und nichtkonvexen Zielfunktion, die schwer zu optimieren ist. Zudem stützen sich konventionelle Methoden in der Regel auf gemessene externe Reaktionskräfte und -momente, was die zu analysierenden Bewegungen aufgrund der erforderlichen Laborumgebung stark einschränkt.

Angesichts dieser Einschränkungen eröffnen datengesteuerte maschinelle Lernverfahren enorme Möglichkeiten, da sie generell eine schnelle und rauschresistente Datenanalyse erlauben. Diese Arbeit untersucht die Anwendbarkeit solcher Methoden auf die inverse Dynamik der menschlichen Bewegung und beschäftigt sich mit dem Entwurf geeigneter Regressionsmodelle. Die vorgeschlagenen Methoden sind in der Lage, die zugrundeliegenden Gelenkmomente und äußeren Kräfte mit hoher Genauigkeit (bei Gangsequenzen: relative mittlere quadratische Fehler von 7,0 %, 16,1 % und 11,9 % für Reaktionskräfte, Reaktionsmomente und Gelenkmomente, was Pearson's Korrelationskoeffizienten von 0,91, 0,83 und 0,82 entspricht) vorherzusagen und gleichzeitig die Berechnungszeiten um zwei Größenordnungen im Vergleich zur traditionellen Optimierung zu reduzieren.

Ein allgemeines Merkmal menschlicher Bewegungsdaten ist die Diskontinuität an Kontaktphasenübergängen, z.B. im Moment der Bodenberührung des Fußes. Durch Veränderung der Anzahl der Kontaktpunkte des menschlichen Modells zu seiner Umgebung wird der

Satz der dynamischen Gleichungen grundlegend verändert, und zwar in dem Sinn, dass äußere Einflüsse an den entsprechenden Punkten erlaubt oder verboten werden. Motiviert durch diese Eigenschaft, wird ein mehrstufiger Regressionsansatz vorgestellt. Das Verfahren identifiziert zunächst die aktuelle Gangphase und beschränkt die Inferenz von Gelenkmomenten und Kontaktkräften auf den resultierenden Unterraum. Auf diese Weise wird die Regression unrealistischer endlicher Kräfte während der Schwungphasen im Vergleich zu einem Modell, das die Kräfte ohne Kenntnis des Kontaktzustandes schätzt, deutlich reduziert.

Aktuelle Probleme von maschinellen Lernmethoden zur Lösung der inversen Dynamik sind ein Mangel an geeigneten Datensätzen und dass die Einhaltung der EOM durch die vorhergesagten Größen nicht garantiert ist. Beide Probleme werden durch ein in dieser Arbeit vorgestelltes selbst-überwachtes Lernverfahren adressiert. Der Ansatz erlaubt ein zykluskonsistentes Training eines künstlichen neuronalen Netzes mit reinen Bewegungsdaten, d.h. ohne jegliche Ground-Truth-Kräfte und -Momente. Anstatt einen direkten Verlust auf die Zielkräfte zu minimieren, löst das Modell ein Anfangswertproblem basierend auf den vorhergesagten Kräften und minimiert den Abstand zwischen der resultierenden Simulation und der Eingangsbewegung. Dies wird durch die Implementierung einer differenzierbaren vorwärtsdynamischen Verlustschicht realisiert, die einen Rückwärtsfluss von Gradienten erlaubt und in das Training des neuronalen Netzes integriert werden kann. Zusätzlich enthält das Modell eine entsprechende Schicht für die inverse Dynamik, die die geschätzten Kontaktkräfte entkoppelt von den vorhergesagten Gelenkmomenten auswertet. Somit ermöglicht das Modell nicht nur das Training auf leicht verfügbaren Bewegungsdaten, sondern beschränkt auch die vorhergesagten Variablen unter Verwendung beider dynamischer Richtungen zur optimalen Erfüllung der EOM. Das neuronale Netzwerk behält seine stabile Leistung auch bei kleinen gelabelten Datensätzen, die aus Dynamikdaten von nur 2 bis 3 Probanden bestehen, indem es die Fähigkeit zu generalisieren auf größeren nicht gelabelten Bewegungsdatensätzen lernt. Darüber hinaus ermöglicht die Methode selbstüberwachtes Transferlernen unbekannter Bewegungstypen, Bewegungsgeschwindigkeiten und abweichender Skelettmerkmale.

Die vorgestellten lernbasierten inversen Dynamikansätze werden anhand eines selbst aufgezeichneten Datensatzes von Geh- und Laufsequenzen, die von 22 Probanden ausgeführt wurden, sowie eines öffentlichen Dynamikdatensatzes [39] und Gangsequenzen aus der bekannten CMU-Datenbank [18] evaluiert. Der selbst aufgezeichnete Datensatz steht der Forschungsgemeinschaft zur Verfügung.

1 INTRODUCTION

The human locomotor system is a complex construction consisting of the skeleton, the nervous system, muscles, tendons and ligaments. Its proper functioning enables us to move in and interact with our environment. It thus represents a basic human need. The study of the locomotor system is subject of biomechanics. It is the foundation of numerous research fields, significant to the quality of human life, like diagnostics of diseases and locomotor disorders, development of rehabilitation techniques and prosthetics, optimization and observation of work spaces and movement patterns. In the course of these studies researchers require measures to quantify healthiness and effectiveness of movement. One common choice are the net joint moments which unite the effect of all forces acting on the linkage of body segments, e. g. muscle activation forces, tension of ligaments and bone-on-bone forces. These net moments form a first approximation to the stress at skeletal joints and can be used to calculate the expended metabolic energy. They yield a foundation for assessing human motion. Joint moments cannot be measured in a non-invasive way, but can be estimated through **inverse dynamics** analysis.

1.1 APPLICATIONS AND CHALLENGES OF INVERSE DYNAMICS

In the research of neurological disorders inverse dynamics plays an important role. The analysis of joint moments facilitates early diagnosis in diabetes-induced peripheral neuropathy [123], it helps to asses the risk of falling for patients suffering from Parkinson's disease [109] and to understand the pathological mechanics in cerebral palsy to offer optimal treatment [41]. These are just a few examples of the broad applicability of inverse dynamics analysis in the medical sector. Many of the related studies focus on the gait pattern of patients, since walking is the most natural form of movement. The human gait is a periodic movement consisting of multiple gait phases as depicted in Figure 1. Regarding one foot, the gait period can be divided into *load response*, *single support*, *pre-swing* and *swing phase*. Load response and pre-swing are also called *double support* phases. The associated ground reaction forces and joint moments exhibit a periodic pattern with progressions that are typical of the current phase. Example curve progressions can be seen in Figure 2. The rehabilitation of a normal gait pattern that has been affected by disease or injury is at the forefront of physical therapy. This is also the case in prosthesis design and alignment. Here, inverse dynamics can be used to estimate the load at the prosthetic device and the expended energy during its movement [159].

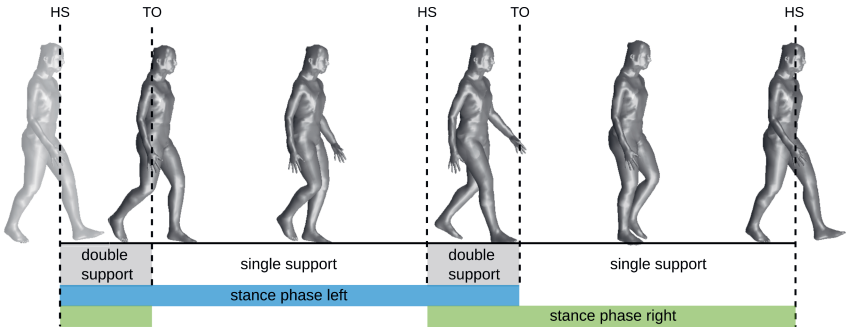


Figure 1: Regular gait period. Heel-strike (HS) and toe-off (TO) are marked by dashed vertical lines. The representation of the person was generated using SMPL, a learned model of human body shape and posture [84].

Inverse dynamics analysis is also of great use in the field of sports science where joint moments are considered in order to evaluate relevant movement sequences: In weight lifting the correlation between lower limb moments and resulting bar velocities and accelerations is investigated to find the most important factors for improving performance [63]. Knee moments during bodyweight squats are analyzed to determine the optimal stance and lower the risk of harmful overloading of the joints. The effect of trunk lean on ball velocities and upper extremity joint moments during pitching is studied in order to investigate the cause of frequently occurring injury [1].

Just as performance in sports can be assessed by joint moments, so can posture and movement in the workplace, which is a relevant topic for the general population. Inverse dynamics enables researchers to evaluate the effect of sitting postures [71] or workspace restrictions [42] on the load at spine joints. In summary, the application possibility of inverse dynamics is broad and not restricted to the mentioned examples. The method is relevant for numerous further research and industrial fields, such as computer graphics (synthesis of movement and games) and robotics (optimal trajectory planning), which will not be discussed in detail.

Now we will touch on the method itself: The desired joint moments are calculated inversely from the motion of a physical model and the exterior forces using a system of equations that describe the dynamics of a physical model, the *equations of motion* (EOM). When the human body is in contact with the ground, its weight transmits a force to the ground. Conversely, because of Newton's third law, a force is transmitted from the ground to the body called the *ground reaction force* (GRF). In many relevant scenarios, like gait analysis, this force represents the only external force, apart from gravity. In fact, the GRF alone already yields significant information for the analysis of movement [156, 158].

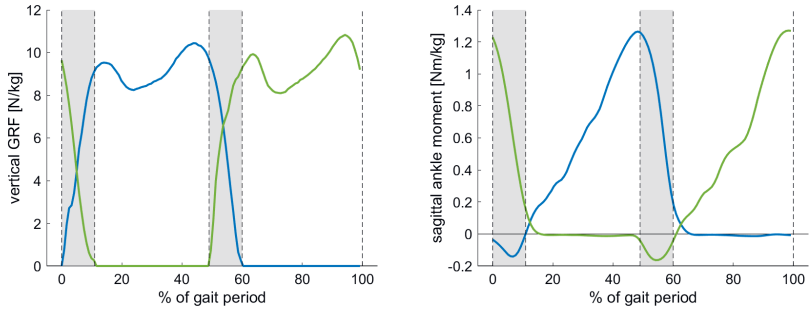


Figure 2: Typical force and moment progressions during one period of human gait. The left hand side shows vertical ground reaction forces and the right hand side shows net ankle moments in the sagittal plane[1]. The grey areas mark double support.

The clinical standard procedure of inverse dynamics is to inversely calculate the joint moments from the GRF vector, its center of pressure (COP), the geometry of the skeleton and the recorded motion of the body. For this purpose a person's movement has to be recorded by a motion capture system and the GRF and COP have to be measured with force plates, which restricts the method to a laboratory setup and necessitates expensive measurement systems. The straight-forward inverse dynamics approach is to consider all body parts in a free body diagram. The sum of all acting forces accounts for the observed linear and angular acceleration of the body's center of mass. This is expressed in Newton-Euler equations for each body. If the exterior influence is known, these equations yield the joint forces and moments passed to the adjacent body segment. This way, forces and moments are propagated along the kinematic chain representing the human body. The process can be done in a bottom-up or top-down procedure starting at any end-point of the kinematic chain where the exterior influence is known. If there exists only one contact point to the environment, e. g. the ground, the corresponding contact force and moment as well as the joint moments are fully defined by the kinematics and inertial properties of the model. In the case of multiple contact points, the system of dynamical equations is overdetermined and an infinite number of solutions exists. With regards to locomotion this is always the case during double support, when both feet are in contact with the ground. A problem of this propagation approach is that measurement errors accumulate along the kinematic chain, so that the accuracy of the resulting joint moments is depending on the placement of the joint in the chain. Furthermore, motion capture uncertainties are propagated with the squared capture rate to the accelerations, which makes them quite error-prone.

1 The moment vector is orthogonal to the plane spanned by the direction of movement and the vertical.

The inverse dynamics problem can also be solved by optimization. In this case, the EOM are rearranged to output the generalized forces which are composed of the joint moments and forces acting on the global coordinates of the model. The latter are zero in reality. This way, the uncertainty of the joint moments is equally distributed. The according optimization problem is formulated as follows: The kinematics are optimized and kept close to the observation. At each step, the EOM are solved for the generalized forces and the non-physical global actuation of the model is forbidden by constraints or minimized by regularization. The remaining components are the sought-after joint moments [160]. The approach is referred to as *inverse dynamics optimization*. Alternative optimization approaches used to determine net joint moments are *forward dynamics optimization* and *predictive dynamics optimization* [161]. The latter will be described in more detail in Chapter 4.

1.2 LEARNING INVERSE DYNAMICS

Although the described methods for dynamics optimization are well established, they also entail some considerable problems: First of all, the computational cost is high including calculation of complex equations at every single iteration. Furthermore, the methods depend on measured contact forces due to the ambiguity of exterior and interior forces during the double support of the model. This fact limits the application to laboratory use and severely restricts the movements studied, since test subjects must hit stationary force plates that record the ground interaction. Alternative measurement approaches, like pressure insoles [37, 38], often only provide a rough approximation of the three-dimensional force vector and include new challenges, e. g. impairment of normal movement due to interfering bulky inlays. Apart from necessary expensive laboratory equipment, the optimization routine itself includes several challenges as well: The convergence behavior and the final result are very sensitive to the initialization of the optimization and to the weighting of multiple included objectives. This is owed to the non-convexity of the problem, but also to model inaccuracies and measurement noise which lead to a necessary violation of the modeled EOM.

Machine learning techniques offer a promising alternative to the optimization of human dynamics and are not or less affected by the addressed challenges above. In such a framework the connection between a motion pattern and the underlying forces is learned on the basis of a training set of dynamical data [59, 64, 171]. This way, a new movement pattern can be assessed using the knowledge gained in the training phase; i. e. unobservable joint moments (and contact forces) are directly inferred from the motion data. Compared to traditional inverse dynamics techniques, machine learning is characterized by some advantageous features, like fast prediction times, robustness against measurement inaccuracies and

independence from complex dynamical models and expensive sensors. In contrast to a dynamical calculation or optimization, a data-driven model does not require the full kinematics as input. Therefore, the method can also be applied to reconstructed motions from sensor or image data that typically do not include the position and orientation of the person in a global frame. Once a regression model has been trained, it can be directly applied without recording of contact forces or tedious optimization.

A computationally efficient and robust analysis of the forces acting on the human musculoskeletal system holds many opportunities, especially with regards to nowadays ever growing vision-based applications. For example, widely used fitness apps could be equipped with video-based software that allows to check the correctness of performed exercises. This would prevent injuries and undesirable stresses caused by incorrect movement patterns. The same principle could be applied in workplace monitoring to detect and alert to unhealthy postures. Likewise, rehabilitation programs could benefit from autonomous monitoring of progress by patients at home. These are just a few examples demonstrating the diverse prospects of learning-based motion analysis. To implement this vision algorithms must solve several subtasks. Humans have to be recognized in videos and joint trajectories have to be estimated. The use of deep neural networks has led to tremendous progress in recent years in solving these problems [22, 68, 82, 90, 93, 100, 149]. The remaining step to allow a fully learning-based analysis of the dynamics is the prediction of contact forces and joint moments from motion, which is subject of this thesis. To this end, several well-known regression methods are applied to inverse dynamics learning and contrasted to each other. The focus, however, is on artificial neural networks, because of their success on related problems of human motion analysis, as mentioned above, and their versatility and good scaling behavior. Neural networks are capable to automatically extract useful features from raw input data and their training and inference can be implemented efficiently using multicore processing [126].

Learning of human dynamics especially with multilayer neural networks that include many weights to be tuned requires large datasets. The necessary data consists of kinematics (3D motion of a kinematic model), exterior GRF, the corresponding COP or ground reaction moment (GRM) and the driving joint torques. Unfortunately, corresponding datasets are few and often very restricted in terms of size and included motion types. When training a model on datasets with too little variability, there is a high risk of overfitting to dataset specific features. A common approach to address this issue is the technique of *data augmentation*. Existing data points are artificially changed and multiplied to generate new points. In the case of image data, this is typically achieved using transformations like color modifications, rotations and mirroring. Considering human dynamics data, augmentation is bound to be more complex since it has to comply with the physical context between motion, forces and moments. One possible method is the use of a physics engine to

simulate dynamics (here the motion of a human model) based on randomly sampled input parameters [122]. A newer approach to deal with lacking training data is the so-called *self-supervision*. Rather than artificially increasing the amount of data, self-supervision enables training on unlabeled data, e. g. by defining auxiliary tasks to learn a meaningful embedding of the input data in the feature space or by defining auxiliary loss functions that are independent of labels or unique correspondences between data points. In the considered problem this corresponds to training networks purely using motion data without the need for GRF/M or joint torque information by use of multiple physics-based loss layers. The realization of self-supervision for learning of inverse dynamics is a major contribution of the present thesis.

Another approach that can be helpful when dealing with missing training data is *transfer learning*. Here, a distinction is made between a source and a target domain (consisting of data or features and their marginal probability distribution). Typically, a model exists that performs a prediction task in the source domain, and the goal is to transfer it to the target domain of interest in such a way that the model's predictive ability in the target domain benefits from the knowledge gained in the source domain. Thus, transfer learning can be used to generate powerful models despite limited data availability. This objective is also pursued in the present work. The realized self-supervised learning allows the extension and transfer of models to different locomotion types and dataset characteristics without the need for fully labeled data of the target domain.

1.3 CONTRIBUTIONS

The goal of this work is the realization of learning-based inverse dynamics, i. e. the combined regression of GRF/M and net joint moments from human motion data. The focus is on analyzing locomotion, such as walking and running, and to investigate the applicability of machine learning models to these fundamental forms of movement. Influenced by the scarcity of suitable dynamics datasets, the contributions of this thesis include

a) Generation of a dynamics dataset, encompassing human gait and running at various speeds, and which is made available to the community[2].

b) Supervised learning of inverse dynamics using different machine learning methods such as artificial neural networks, random forests, ridge regression and support vector machines.

c) Self-supervised learning of inverse dynamics using artificial neural networks

2 http://www.tnt.uni-hannover.de/datasets/HumanLocomot.zip

a) Dataset Generation

The necessary dynamics dataset to realize machine learning of inverse dynamics was recorded in the laboratory at the Institute for Information Processing (TNT). This included 3D motion capture and force plate measurements yielding the GRF/M. A predictive dynamics optimization is customized to the characteristics of the recorded data and used to estimate the net joint moments. For this purpose, a physical model of the human body is designed and equations of motion are formulated. In contrast to inverse and forward dynamics optimization, the predictive dynamics approach optimizes all relevant quantities and is able to correct for model and measurement inaccuracies to some extent.

b) Supervised Inverse Dynamics Learning

Several learning-based methods are proposed and applied to the considered problem. In particular, end-to-end trainable models are compared to a multi-stage approach. The periodicity of gait, shown in Figure 1, brings a clear structure to the data, which has been exploited in the design of this method. The contact state of the kinematic chain, representing the human skeleton, already contains important information for the estimation of the underlying forces. Therefore, the multi-stage method includes a classification of gait phases and regresses GRF/M and joint torques using the resulting data subset. To facilitate this classification, suitable features are extracted from the raw motion input, namely, the absolute velocities of feet points.

In order to include the inverse dynamics regression into vision or sensor-based motion analysis, the proposed methods have to be applicable to reconstructed motions. This data generally lacks global information about the position and orientation of the human model. Accordingly, the proposed methods are designed to operate only on local coordinates, i. e. joint angles and angular velocities: The multi-stage method is extended by an initial regression of the global coordinates to be able to calculate foot velocities. The end-to-end regressions can be easily trained using the corresponding subset of the input data.

In summary, the contributions of this part of the thesis are:

- Supervised learning of inverse dynamics is realized.
- A multi-stage approach is proposed and compared to end-to-end learnable methods.

c) Self-Supervised Inverse Dynamics Learning

In order to address the lack of public datasets that include human dynamics, this thesis proposes inverse dynamics learning with self-supervision. To this end, a cycle-consistent

training scheme operating on pure motion data is introduced using two novel differentiable neural network layers that calculate a loss without requiring force or moment training data. One loss layer, termed *forward layer*, integrates the EOM of a physical model to generate a simulated motion based on the neural network output (predicted forces and moments). The neural network and subsequent forward layer realize a cycle from motions to forces and back allowing the use of a cycle-consistent loss. In addition, the model includes an *inverse layer* that penalizes GRF/M which do not match the observed accelerations. In contrast to the forward dynamics layer, it considers the ground reaction independently from joint torques, which supports decoupled control of both variables during training. The introduced model is applied in a semi-supervised setting that alternates between training with and without force and moment ground truth. This way, the effect of extending a labeled training set by means of self-supervision and the accompanying increase in data variability is investigated.

The contributions of this part can be summarized as follows:

- Self-supervised cycle-consistent learning of inverse dynamics is introduced using a differentiable forward dynamics layer.
- An additional differentiable inverse dynamics layer is included to decouple the training of GRF/M and joint torques.

1.4 STRUCTURE OF THE THESIS

The thesis is structured in the following way (visualized in Figure 3):

Chapter 2: Overview of related work. The problem is placed in its scientific context by description of relevant state-of-the-art methods. The review is structured into traditional inverse dynamics, learning-based inverse dynamics and related work addressing motion analysis with self-supervision.

Chapter 3: A presentation of the theoretical background. First, the modeling of human motion divided into kinematics and dynamics is described including the mathematical foundation to parameterize rigid motion and the acting forces and torques. Subsequently, used machine learning algorithms are presented.

Chapter 4: Data recording and necessary pre-processing steps. The recorded dataset including 3D motion capture and force plate measurements is presented as well as general parameters that predefine the skeletal human model. The estimation of net joint moments by predictive dynamics optimization is described in detail.

Chapter 5: Supervised learning of inverse dynamics for human motion. Two approaches are compared: end-to-end trainable models and multi-stage regression consisting of root regression, contact feature extraction, gait phase classification and control regression.

Chapter 6: Dynamics Net, a self-supervised artificial neural network for inverse dynamics. Two differentiable neural network layers to calculate loss functions independent from force and moment data enable cycle-consistent training. One of them implements a forward dynamics step to generate a simulated motion, while the other executes a bottom-up inverse dynamics step to control the matching between motion and ground reaction.

Chapter 7: The results are summarized and discussed. Furthermore, the advantages but also the limitations of the methods are emphasized indicating perspectives for future research.

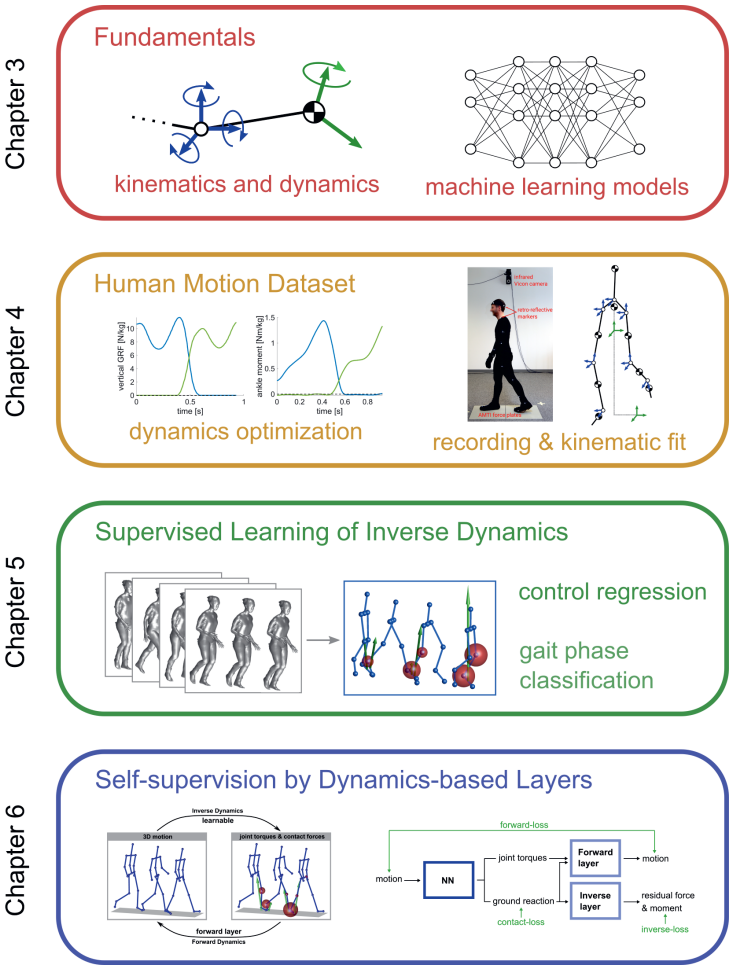


Figure 3: Structure of the dissertation.

1.5 PUBLICATIONS

The following list includes all publications released during the time at the *Institute for Information Processing* in the field of human motion analysis.

[169] Petrissa Zell und Bodo Rosenhahn. "A physics-based statistical model for human gait analysis." In: *German Conference on Pattern Recognition (GCPR).* Oct. 2015.

Physics-based modeling is a powerful tool for human gait analysis and synthesis. Unfortunately, its application suffers from high computational cost regarding the solution of optimization problems and uncertainty in the choice of a suitable objective energy function and model parametrization. Our approach circumvents these problems by learning model parameters based on a training set of walking sequences. We propose a combined representation of motion parameters and physical parameters to infer missing data without the need for tedious optimization. Both a k-nearest-neighbour approach and asymmetrical principal component analysis are used to deduce ground reaction forces and joint torques directly from an input motion. We evaluate our methods by comparing with an iterative optimization-based method and demonstrate the robustness of our algorithm by reducing the input joint information. With decreasing input information the combined statistical model regression increasingly outperforms the iterative optimization-based method.

[173] Petrissa Zell, Bastian Wandt and Bodo Rosenhahn. "Joint 3D Human Motion Capture and Physical Analysis from Monocular Videos." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.* July 2017.

Motion analysis is often restricted to a laboratory setup with multiple cameras and force sensors which requires expensive equipment and knowledgeable operators. Therefore it lacks in simplicity and flexibility. We propose an algorithm combining monocular 3D pose estimation with physics-based modeling to introduce a statistical framework for fast and robust 3D motion analysis from 2D video data. We use a factorization approach to learn 3D motion coefficients and join them with physical parameters, that describe the dynamic of a mass-spring-model. Our approach does neither require additional force measurement nor torque optimization and only uses a single camera while allowing to estimate unobservable torques in the human body. We show that our algorithm improves the monocular 3D reconstruction by enforcing plausible human motion and resolving the ambiguity of camera and object motion. The performance is evaluated on different motions and multiple test data sets as well as on challenging outdoor sequences.

[170] Petrissa Zell and Bodo Rosenhahn. "Learning-Based Inverse Dynamics of Human Motion." In: The IEEE International Conference on Computer Vision (ICCV) Workshops. Oct. 2017, pp. 842-850.

In this work we propose a learning-based algorithm for the inverse dynamics problem of human motion. Our method uses Random Forest regression to predict joint torques and ground reaction forces from motion patterns. For this purpose we extend temporally incomplete force plate data via a direct Random Forest regression from motion parameters to force vectors. Based on the resulting completed data we estimate underlying joint torques using a modified physics-based predictive dynamics approach. The optimization results for model states and controls act as predictors and responses for the final Random Forest regression from motion to joint torques and ground reaction forces. The evaluation of our method includes a comparison to state-of-the-art results and to measured force plate data and a demonstration of the robust performance under influence of noisy and occluded input.

[174] Petrissa Zell, Bastian Wandt and Bodo Rosenhahn. "Physics-based Models for Human Gait Analysis." In: *Handbook of Human Motion.* Cham: Springer International Publishing, 2018, pp. 267-292.

This chapter deals with fundamental methods as well as current research on physics-based human gait analysis. We present valuable concepts that allow efficient modeling of the kinematics and the dynamics of the human body. The resulting physical model can be included in an optimization-based framework. In this context, we show how forward dynamics optimization can be used to determine the producing forces of gait patterns. To present a current subject of research, we provide a description of a 2D physics-based statistical model for human gait analysis that exploits parameter learning to estimate unobservable joint torques and external forces directly from motion input. The robustness of this algorithm with respect to occluded joint trajectories is shown in a short experiment. Furthermore, we present a method that uses the former techniques for video-based gait analysis by combining them with a nonrigid structure from motion approach. To examine the applicability of this method, a brief evaluation of the performance regarding joint torque and ground reaction force estimation is provided.

[171] Petrissa Zell and Bodo Rosenhahn. "Learning inverse dynamics for human locomotion analysis." In: *Neural Computing and Applications* 32.15 (2020), pp. 11729-11743.

In this work, learning-based inverse dynamics algorithms are proposed for the analysis of human motion. Immeasurable joint torques and exterior contact forces are directly estimated from motions by machine learning techniques including deep neural

networks, random forests and Ridge regression. A multistage subclass approach is introduced. The method recovers occluded motion data and generates meaningful features, as well as gait phase labels to restrict and facilitate the regression of forces and moments. In contrast to the state-of-the-art inverse dynamics optimization, the learning-based methods are independent of ground reaction force measurements and the global position and orientation of the human body. These properties make the application to reconstructed poses from videos or inertial measurements possible, creating fast and simple access to the underlying dynamics of recorded human motions. The performance of the proposed methods is evaluated on a self-recorded data set including walking and running motions and on a publicly available gait data set by Fukuchi et al. (PeerJ 6:e4640, 2018). Furthermore, the applicability to reconstructed gait sequences taken from the well-known CMU database (Human motion capture database, 2014. http://mocap.cs.cmu.edu/) is investigated. Finally, the method is tested as a tool to detect abnormal torque distributions in gait, based on a reconstructed 3D motion of a limping subject.

[172] Petrissa Zell, Bodo Rosenhahn and Bastian Wandt. "Weakly-supervised Learning of Human Dynamics." In: *European Conference on Computer Vision (ECCV).* Aug. 2020.

This paper proposes a weakly-supervised learning framework for dynamics estimation from human motion. Although there are many solutions to capture pure human motion readily available, their data is not sufficient to analyze quality and efficiency of movements. Instead, the forces and moments driving human motion (the dynamics) need to be considered. Since recording dynamics is a laborious task that requires expensive sensors and complex, time-consuming optimization, dynamics data sets are small compared to human motion data sets and are rarely made public. The proposed approach takes advantage of easily obtainable motion data which enables weakly-supervised learning on small dynamics sets and weakly-supervised domain transfer. Our method includes novel neural network (NN) layers for forward and inverse dynamics during end-to-end training. On this basis, a cyclic loss between pure motion data can be minimized, i. e. no ground truth forces and moments are required during training. The proposed method achieves state-of-the-art results in terms of ground reaction force, ground reaction moment and joint torque regression and is able to maintain good performance on substantially reduced sets.

[150] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin and Bodo Rosenhahn. "CanonPose: Self-Supervised Monocular 3D Human Pose Estimation in the Wild." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* Jun. 2021.

Human pose estimation from single images is a challenging problem in computer vision that requires large amounts of labeled training data to be solved accurately. Unfortunately, for many human activities (e. g.outdoor sports) such training data does not exist and is hard or even impossible to acquire with traditional motion capture systems. We propose a self-supervised approach that learns a single image 3D pose estimator from unlabeled multi-view data. To this end, we exploit multi-view consistency constraints to disentangle the observed 2D pose into the underlying 3D pose and camera rotation. In contrast to most existing methods, we do not require calibrated cameras and can therefore learn from moving cameras. Nevertheless, in the case of a static camera setup, we present an optional extension to include constant relative camera rotations over multiple views into our framework. Key to the success are new, unbiased reconstruction objectives that mix information across views and training samples. The proposed approach is evaluated on two benchmark datasets (Human3.6M and MPII-INF-3DHP) and on the in-the-wild SkiPose dataset.

2

RELATED WORK

This chapter addresses the current state of the art regarding inverse dynamics of human motion. The description will be split into traditional physics-based inverse dynamics analysis (2.1), learning-based methods for inverse dynamics (2.2) and related problems and machine learning with reduced supervision in the broader field of human motion analysis (2.3).

2.1 INVERSE DYNAMICS BY PHYSICAL SIMULATION

The study of human movement has a very long tradition in human history. As a form of art it already appears in the classical antiquity and resurfaces throughout the epochs reflecting the ever-present interest in understanding the way animals and in particular humans move. The invention of chronophotography in the late 19th century allowed to visualize even fast motion sequences and thus paved the way for the evolution of human motion analysis. The technique was used in a famous experiment by Eadweard Muybridge in 1878. He proofed the existence of a flight phase in a horse's gallop by inspecting an image sequence taken by 12 cameras which recorded in rapid succession. This kind of serial recording is considered a predecessor of *moving pictures*. In the same manner Muybridge studied human movement like walking downstairs, boxing and the gait of children. His work is considered influential on the emergence of biomechanics as an independent research field [65].

In many of the following biomechanical studies physical models were needed to describe the human body and perform inverse dynamics analysis. Physical human models can be classified into three categories: skeletal, musculoskeletal and neuromusculoskeletal models. In skeletal models, which are used in this thesis, no muscle activation and muscle contraction dynamics are considered. They consist of multiple rigid bodies linked by joints. The effect of the muscle and tendon forces at the linkage is approximated by net joint moments that produce the motion of the rigid body system [31]. Each body segment is associated with inertial parameters, like mass, location of center of gravity and moment of inertia. Average values of these parameters in the population and their relation to body dimensions were determined by extensive studies using different techniques from cadaver studies to immersion methods and measurement of reaction force displacements [29]. As a founding father of modern biomechanics, David A. Winter provides a fundamental overview in *Biomechanics and Motor Control of Human Movement* [157].

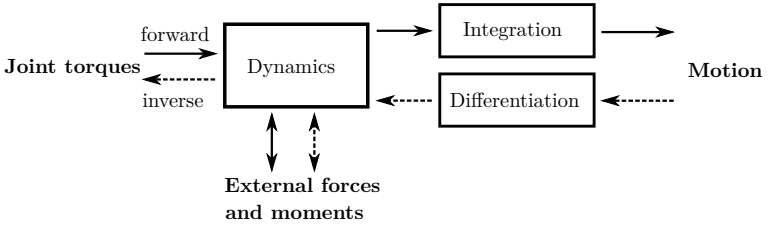


Figure 4: Schematic visualization of explicit dynamics formulations.

Based on the dynamics of a skeletal model, the inverse dynamics problem can be solved using different formulations. In general, a distinction is made between explicit and implicit formulations. In explicit dynamics, the dynamical equations are either integrated or differentiated referred to as forward and inverse approaches, respectively[1]. Figure 4 illustrates the corresponding procedures. Implicit formulations, on the other hand, use the dynamical equations indirectly as constraints of an optimization problem. The predictive dynamics approach used in Chapter 4 falls into this category. Based on a specific problem formulation, different solution methods are possible. Here one distinguishes between non-optimization and optimization approaches. In the following the different problem formulations and solving methods will be presented in more detail.

2.1.1 *Inverse Approaches*

The traditional inverse approach is a non-optimization approach that directly solves the equations of motion for the joint torques [38, 92]. It is referred to as *Newton-Euler method*, since the dynamics of each body segment are expressed by Newton-Euler equations. In the case of single support, the number of unknown variables is identical to the number of equations and a unique solution exists. For double support, the partitioning of the external forces and moments between the contact points is unknown, so that additional measurement of contact forces is necessary to directly solve the system of equations. Conventionally, the solution of the problem is done in a sequential manner propagating joint forces and moments along the kinematic chain. The direct calculation of joint torques is quick and straightforward, however, the uncertainties introduced by multiple error sources are directly propagated to the joint torques as well. Common sources of error are capturing uncertainties and differentiation of motion states, measurement inaccuracies of contact forces and model approximation. The propagated errors often lead to unrealistic torque profiles [121].

1 There also exist mixed approaches that aim to benefit from the advantages of both, forward and inverse dynamics.

If the ground reaction is measured over the whole gait period separately for each foot, e. g. with multiple force plates or using pressure insoles, and used as additional input to the captured kinematics, the system of dynamical equations becomes overdetermined even for double support. In other words, a bottom-up inverse dynamics algorithm, starting at both feet with the corresponding external forces and moments, would yield a residual force and moment at the last segment (e. g. the head). This residual is a measure for the discrepancy between kinematics and contact dynamics and for the accuracy of the resulting joint torques [32]. It is equivalent to non-existent external actuation of the end segment. The goal of inverse optimization approaches, in general, is to reduce residual forces and moments. This is achieved by adapting the kinematics, the exterior forces, the anthropometric model parameters or a combination of the above [16, 25, 66, 70, 119].

2.1.2 *Forward Approaches*

Forward approaches use the joint torques as input and integrate the dynamical equations to generate a simulated motion. In order to apply a forward approach to the inverse dynamics problem, the forward step is included in an optimization routine. The joint torques are the optimization variables and the deviation of the simulated motion from the motion capture data yields the objective. A general challenge of forward dynamics optimization is the stabilization of the simulation. On top of the method dependent truncation error of numerical integration, joint torque uncertainties are passed on to the poses multiplied by the squared integration time. This is due to the twofold integration of the equations of motion (2nd order differential equations). Researchers apply different schemes to control and stabilize the simulation, namely full actuation, under-actuation and kinematic constraints [32].

Full actuation controls all degrees of freedom (DOF) of the model including non-physical actuation of the global coordinates [97]. These global forces and moments are equivalent to the residuals of the inverse approach described earlier. Fully-actuated control is often combined with a minimization of the non-physical actuation as in the *residual reduction algorithm* of OpenSim [25]. Here, residual reduction is used prior to the forward dynamics step and only adjusts the kinematics of the HAT (head, arms and torso) to reduce the discrepancy between ground reaction and motion.

In under-actuated control, the number of actuators is less than the DOF of the system. Joint torques only exist for actual joint coordinates, i. e. global root coordinates are not actuated explicitly. Instead, the missing control is achieved implicitly by modelling the ground contact [79, 97, 140]. A corresponding foot-contact model uses damped spring forces and Coulomb friction to determine the vertical and horizontal forces as well as a geometric model to distribute the pressure and determine the COP [127].

Instead of modeling the contact forces, they can be included implicitly in the form of algebraic constraints on the equations of motion [15, 34]. Then the system is fully-actuated, but additional constraint equations must be applied alternatingly switching between stance and swing phases. This requires a detection of impact and handling of the associated discontinuity to avoid numerical instabilities due to exploding accelerations.

To improve the stability of forward dynamics, feedback-control approaches have been developed predominantly in robotics research. In these methods, the joint torques are adjusted according to the current deviation of the simulated motion from the target motion. An example of such a feedback-controller is *computed torque control* [94]. This method uses the inverse dynamics torques as input for the forward step and adjusts them according to the feedback-error. A feedback-controller realized by means of optimization is *model predictive control* [67]. It considers a temporal prediction horizon starting at the current time and uses an internal dynamical model to predict the output, in our case the motion. In order to track trajectories, an objective function is defined on the prediction horizon and the next control values are determined by optimization. The procedure allows for online execution. These feedback-control approaches can be classified as mixed models that adopt both forward and inverse dynamics.

2.1.3 *Implicit Approaches*

Optimal control methods are among the implicit approaches [30, 101, 107]. The goal of these methods is to find optimal movement trajectories together with the acting controls that minimize some form of performance measure, like the expended energy/effort. Instead of explicitly solving or integrating the EOM, they are used implicitly in terms of constraints of the optimization routine [144]. Predictive dynamics by Xiang et al. [161] is an example of such a method. The approach optimizes the kinematics as well as the joint torques while EOM are treated as equality constraints. The optimization procedure used in this thesis to generate joint torque training data (cf. Chapter 4) is motivated by this method. In the implementation by Xiang et al., a performance measure, such as the dynamic effort, is minimized while subjected to a number of constraints. In addition to the EOM, the constraints include joint limits, torque limits, ground penetration, dynamic balance, etc..

2.2 LEARNING-BASED INVERSE DYNAMICS

In the previous section, traditional physics-based methods for the inverse dynamics problem have been described. In general, these approaches are characterized by complex modeling and relatively long computation times. As discussed above, further challenges are measurement and model discrepancies and stabilization of the dynamical system during

forward integration. Alternatively, researchers propose learning-based methods to achieve a higher level of robustness.

First the works that are most closely related to the present thesis are considered. The corresponding methods predict joint torques and/or ground reactions from skeletal kinematics during human locomotion [59, 103, 118]. Johnson et al. [59] use sparse coding to encode joint angle and corresponding joint torque data and map between the sets using ridge regression and neural networks. However, the achieved torque errors in the order of 60 Nm are considered too large by the authors to compete with a traditional inverse dynamics analysis. In addition, the dataset including gait data of a single subject does not allow for evaluation of inter-subject generalization. Oh et al. [103] propose a hybrid-approach that calculates GRF/M by inverse dynamics during the single support and predicts the distribution to both feet during double support using an artificial neural network. Based on the estimated GRF/M, inverse dynamics yields the corresponding joint moments.

A data-driven optimization method by Lv et al. [85] pursues the same objective, i. e. the estimation of joint torques without measurement of GRF/M. The optimization includes a model of the prior probability of contact states to establish appropriate exterior forces and moments. The prior is estimated by a local Gaussian mixture model of the principal component scores of neighbouring data samples. The neighbourhood is defined by a k-nearest neighbour algorithm based on the kinematics of the current frame. In Chapters 5 and 6 the results of this data-driven optimization are used for comparison with the learning-based methods proposed in this work.

Further related works can be found in the field of robotics, e. g. learning optimal control of humanoid motion, exoskeletons and industrial robots. These works either consider a different model (a non humanoid robot [81, 130]), a different motion type (e. g. elbow movements [76]) or focus on synthesis of motion without explicit evaluation of joint torques [118]. For example, a work by Liu et al. [81] addresses learning-based inverse dynamics for industrial robots. The proposed deep neural network model includes an LSTM-layer (long short-term memory) to model temporal connectivity of the data. The method is evaluated using a robot arm with six DOF. This number is very small in comparison to skeletal human models. Other related works aim to achieve stable gait of robots [56, 153] or animated characters [23, 110, 111, 166] using genetic algorithms and (deep) reinforcement learning. Here, the policy learned by the reinforcement agent represents the motor control of the system, e. g. actuating torques or the activation of musculotendo units [112]. The main objective is to generate stable movement through balance conditions and motion imitation by keyframes rather than subtle stylistic movement traits of the individual characters. Deep reinforcement learning (based on deep neural networks) is also gaining a

foothold in the control of complex musculoskeletal models [73] since it allows learning of policies in high dimensional control and action spaces [133].

There exist further related learning methods that use different input and/or investigate different motion types. Joint torques are estimated based on signals from sensors like inertial measurement units (IMU), force plates and electromyography [3, 78, 136, 164]. For example, Yang et al. [164] developed a smart shoe that is equipped with gyroscope, accelerometer and magnetometer to estimate 3D motion and allows for mobile learning-based inverse dynamics analysis. The authors propose a dependent Gaussian process algorithm that utilizes correlations between kinematics and dynamics and predicts the joint torques based on the sensor signals. The model is trained using data from motion capture and force plates in addition to the sensor output. In contrast to the present thesis, training and evaluation is performed using trials of the same subject. A work by Lim et al. [78] investigates learning of inverse dynamics based on a single IMU located at the lower back of a subject. The method exploits the dynamical relationship between the center of mass of the whole body and the lower extremities. Several segment angles, joint torques and the GRF are predicted from the IMU data by an artificial neural network. The method achieves fair results given the low dimensional input. Further works consider inverse dynamics learning for specific motion types other than gait, e. g. arm and hand movements [35, 132, 154], vertical jumps [83] and rapid side-stepping [60].

Related problems from gait analysis that have been addressed using machine learning include estimation of muscle activation [114, 115, 165], classification of gait changes due to age [4, 5, 40], fall detection [102], person identification by gait pattern [99] and many more that will not be discussed in detail.

2.3 DECREASING SUPERVISION

Many of the aforementioned approaches to learning-based inverse dynamics are trained and evaluated on comparatively small datasets, since the data acquisition and the necessary pre-processing of the dynamics data is very complex and time-consuming [59, 78, 164]. Such a lack of training data can be addressed by reducing the supervision of the learning algorithm, which is subject of many recent publications in machine learning research. Several new techniques and terms have emerged in this context: weak supervision, self-supervision, semi-supervision, distant supervision, few-shot, one-shot and zero-shot learning. It should be noted that these terms are sometimes used ambiguously and are difficult to divide clearly in the literature. Here, only the approaches, closely related to this work will be discussed.

Weak supervision means that an existing model is used to generate *weak* labels for otherwise unlabeled data points. A model trained on a large but weakly labeled set can

show significantly improved performance compared to a model trained on a small dataset with ground truth labels [167]. The model that yields weak supervision is either trained on a small dataset of the same type or solves a different task but can be exploited to create the required labels. In the field of human motion analysis, weak supervision is applied to facilitate problems like monocular motion capture [21, 52, 53, 149, 177], action recognition [108, 152, 176] and motion prediction from single images [61]. The last example is somewhat closer to the problem at hand, since a temporal connection is learned on top of single poses and shapes.

In *semi-supervised* learning, the dataset is split into a usually small portion with labels and a larger portion without labels. The objective is to leverage the unlabeled data to produce a model that performs better than a fully supervised baseline trained using only the small labeled set [104]. This problem presents a realistic scenario where only a small dataset is available and additional data points can be obtained in a simple manner, but are too costly to label. In the present thesis, a semi-supervised learning task is realized in Chapter 6 by artificially reducing the number of sequences that include target GRF/M and joint torques and using the physics-based layers to enable training on the unlabeled part of the data. In human motion analysis, semi-supervised learning is applied e. g. for 3D human pose estimation [96] person identification based on gait [75, 77] and action recognition [117, 131]. For example, it is possible to personalize an existing action classifier based on the signal recorded during use of a wearable sensor without explicit labeling effort. For this purpose, a proxy label approach is applied where the original model generates labels for data points selected according to an information-theoretic criterion. The new data samples are then included in a subject-specific set and used to create a personalized classifier [131]. This proxy labeling is closely related to the weak supervised learning methods described above, but here the weak labels are used to extend an existing set of strong labels.

Another way to circumvent the need for large labeled datasets is *self-supervised* learning. This term generally refers to the generation of a supervisory signal in the absence of labeled training data. Corresponding methods can be roughly divided into those that use an auxiliary task (for which labels can be easily generated) to learn data representations that facilitate the original task [10, 44, 113] and those that use an auxiliary loss function which does not depend on labeled data at all [21, 149, 150, 168]. The self-supervised approach of the present thesis belongs to the latter category. In the following, selected works of this type will be presented.

Zanfir et al. [168] propose a semantic body part alignment loss for self-supervised 3D pose and shape reconstruction. Together with normalizing flow based kinematic priors the loss can be used either in deep learning or direct optimization. Self-supervised training of 3D pose lifting networks has been realized by inclusion of a reprojection loss that enforces self-consistency in the 2D regime [21, 149]. In combination with a discriminator feedback

this allows for training without 2D-3D correspondences. The approach uses a form of cycle-consistency which is an increasingly popular concept in deep learning and has been largely promoted by cycle-consistent adversarial networks [178]. A related self-supervised approach is proposed by Bhatnagar et al. [7] to learn 3D point correspondences for fitting a 3D human model to 3D surface scans. A cycle is implemented by a network that, given an input scan, predicts the point correspondences on the human model (in canonical shape and pose) and a forward map parameterized by the human model.

With respect to the original problem of learning-based inverse dynamics, cycle-consistent loss functions can be designed by implementing subsequent inverse and forward dynamics. The idea, which is also followed in the present work, is to learn the inverse step and utilize a differentiable physics-engine for forward simulation. Differentiable physics-engines are predominantly used in robotics [24, 143, 151], fluid dynamics [124] and reinforcement learning [6]. For a more general application in deep learning, Chen et al. propose *Neural ordinary differential equations* [20] as a new type of neural network whose output is computed by a black box differential equation solver. The backpropagation through arbitrary ordinary differential equations solvers is described allowing for integration in larger end-to-end trainable models.

In the present thesis the self-supervised training procedure is not only applied to improve the fitting of models despite small training sets, but also to realize transfer learning (more precisely domain adaption or expansion) to a target domain lacking recorded forces. The transfer is realized by means of weight sharing and fine-tuning the model to the unlabeled target domain. Since the exact problem of transferring the prediction of forces and moments across different human motion data has not been investigated by other works, listed here are some publications that address related tasks in the broader field of human motion analysis. A popular use case is to leverage simulated labeled data, which can be easily generated in large quantities, as the source domain and transfer the knowledge to real-world data representing the target domain. This approach is used in 3D pose estimation [28, 89], e. g. using adversarial training with a gradient reversal layer [43] to achieve domain invariant feature representation. In the context of human gait analysis, transfer learning based on pre-trained models is used, e. g. to support biometric person identification directly from videos without needing additional feature extraction [54] and across motion types transferring from gait to squats [146]. For the purpose of pathological gait classification across multiple pathologies, a transfer learning approach [145] uses a pre-trained convolutional neural network and fine-tunes the model to extract optimal features for the classification task. This way, extensive unlabeled image data is leveraged to achieve better generalizability in different domains of pathology.

3

FUNDAMENTALS

3.1 RIGID BODY MOTION

Due to the stiffness of skeletal bones, the human body can be well modelled as a system of rigid bodies representing body parts like head, torso and limbs. In order to describe the motion of the entire system, the motions of the individual body parts need to be considered. To this end, a parameterization of their position and orientation in space is required. This section deals with an according representation for translations and rotations and introduces the notion of homogeneous transformations.

3.1.1 *Representation of Position*

The **position** of a point p in Cartesian space is always defined with respect to a reference coordinate frame i using the point coordinates $\boldsymbol{p}^i \in \mathbb{R}^3$. The frame is indicated with a superscript. In addition to points, that represent locations in space, we operate with *vectors* - sometimes called *free vectors* - that define a direction and magnitude and are not constrained to originate at a specific point [134]. In the context of this thesis, vectors are used, for instance, to specify displacements, velocities and forces. While a vector does not change under coordinate transformation, meaning that its direction and magnitude stay the same, its coordinate representation is depending on the reference frame. Therefore, $\boldsymbol{v}^i$ denotes a vector coordinate representation with respect to frame i (cf. Figure 5).

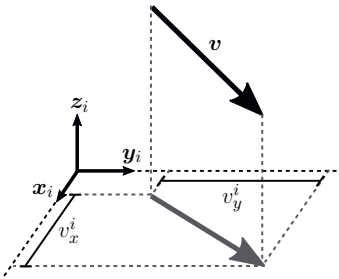


Figure 5: A vector $\boldsymbol{v}$ and its x and y coordinates v_x^i and v_y^i in frame i. To keep the drawing clear, the z coordinate v_z^i is not shown.

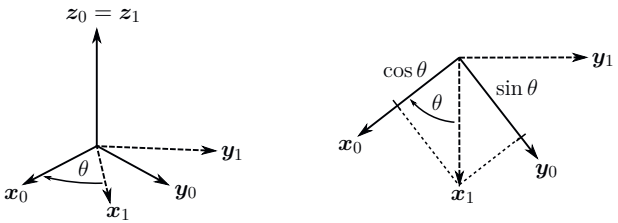


Figure 6: Basic rotation around the z-axis. The left side illustrates the 3D view and the right side shows the axis projection by trigonometric functions.

Algebraic equations, like the EOM of a rigid body or a system of rigid bodies, are only valid if their components, i. e. points and vectors, are represented in the same reference frame. For vectors, the reference frames only need to have parallel axes, since their location in space is not defined. For a rigid body system, properties like center of mass and moment of inertia of individual bodies are defined in a local coordinate frame connected to the respective body. In consequence, one requires coordinate transformations between different reference frames to formulate dynamical equations for rigid body systems.

3.1.2 *Representation of Orientation*

In order to represent **orientation**, we attach coordinate frames to rigid bodies and consider the rotation between them. For a rotation between two coordinate frames denoted by 0 and 1 with orthonormal basis $(\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{z}_0)$ and $(\boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{z}_1)$, respectively, the rotation matrix from frame 1 to frame 0 can be defined by the representation of $(\boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{z}_1)$ with respect to $(\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{z}_0)$:

$$\boldsymbol{R}_1^0 = \left[\boldsymbol{x}_1^0, \boldsymbol{y}_1^0, \boldsymbol{z}_1^0\right] . \tag{1}$$

Since the coordinate axes are unit vectors, this representation can be formulated by projecting $(\boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{z}_1)$ onto $(\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{z}_0)$ using dot products:

$$\boldsymbol{R}_1^0 = \begin{bmatrix} \boldsymbol{x}_1 \cdot \boldsymbol{x}_0 & \boldsymbol{y}_1 \cdot \boldsymbol{x}_0 & \boldsymbol{z}_1 \cdot \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \cdot \boldsymbol{y}_0 & \boldsymbol{y}_1 \cdot \boldsymbol{y}_0 & \boldsymbol{z}_1 \cdot \boldsymbol{y}_0 \\ \boldsymbol{x}_1 \cdot \boldsymbol{z}_0 & \boldsymbol{y}_1 \cdot \boldsymbol{z}_0 & \boldsymbol{z}_1 \cdot \boldsymbol{z}_0 \end{bmatrix} . \tag{2}$$

As an example basic rotations around the coordinate axes (of the reference frame) by an angle θ are presented here. The projected axes are given by trigonometric functions as shown in Figure 6. The resulting rotation matrices are

$$\boldsymbol{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \tag{3}$$

$$\boldsymbol{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{4}$$

$$\boldsymbol{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

Formally, rotations constitute the *special orthogonal group* of order n defined by

$$SO(n) = \left\{ \boldsymbol{R} \in \mathbb{R}^{n\times n} \,:\, \boldsymbol{R}\boldsymbol{R}^T = \mathbb{1},\ \det \boldsymbol{R} = 1 \right\}. \tag{6}$$

The dynamics considered in this work take place in three dimensional space, so that $\boldsymbol{R} \in SO(3)$. In order to mathematically describe the human body as a system of rigid bodies, the following concepts and properties of rotations are useful.

The transformation of point coordinates $\boldsymbol{p}^0$ from frame 0 to a rotated frame 1 is performed by

$$\boldsymbol{p}^1 = \boldsymbol{R}_0^1 \boldsymbol{p}^0. \tag{7}$$

The same concept applies for the consideration of rigid bodies. If a rigid body B is transformed from coordinate frame 0 to coordinate frame 1 by the rotation $\boldsymbol{R}_0^1$, then the set of N points $\{\boldsymbol{p}_i^0\}_B$ with $i = 1, \ldots, N$ on B is transformed to $\{\boldsymbol{p}_i^1 = \boldsymbol{R}_0^1 \boldsymbol{p}_i^0\}_B$ in the same manner.

In a general sense, rotation matrices represent basis transformations. Therefore, they can also be applied to convert linear transformation matrices from one reference frame to another. Let $\boldsymbol{A}$ be a general linear transformation defined in reference frame 0, then the according transformation $\boldsymbol{B}$ with respect to the rotated frame 1 is calculated by the *similarity transformation*

$$\boldsymbol{B} = (\boldsymbol{R}_1^0)^{-1} \boldsymbol{A} \boldsymbol{R}_1^0. \tag{8}$$

This concept will be needed for the transformation of the inertia matrix from a moving coordinate frame, attached to the body, to the fixed, global frame in Section 3.3.

Another fundamental and advantageous property of rotation matrices is the simple way in which a series of rotations can be expressed. Two rotations $\boldsymbol{R}_1^0$ and $\boldsymbol{R}_2^1$ that are performed consecutively result in the overall rotation $\boldsymbol{R}_2^0 = \boldsymbol{R}_1^0 \boldsymbol{R}_2^1$. In general, this composition of rotation matrices can be written as the product

$$\boldsymbol{R}_n^0 = \prod_{i=1}^{n} \boldsymbol{R}_i^{i-1} \,. \tag{9}$$

Note that the rotations are always defined with respect to the current frame, which is a necessary condition for this equation to hold. Every subsequent rotation is post-multiplied, whereas a following rotation about fixed axes needs to be pre-multiplied. The order cannot be changed, since the matrix multiplication is not commutative.

3.1.3 *Homogeneous Transformations*

The described representations for position and orientation build the basis to formally introduce the term *rigid motion* which is a motion that preserves **relative** distances and orientations. A rigid motion in 3D is an ordered pair $(\boldsymbol{t}, \boldsymbol{R})$ consisting of a translation $\boldsymbol{t} \in \mathbb{R}^3$ and a rotation $\boldsymbol{R} \in SO(3)$. It is an element of the *special Euclidean group* of order 3: $SE(3) = \mathbb{R}^3 \times SO(3)$. A rigid motion $(\boldsymbol{t}^0, \boldsymbol{R}_0^1)$, specified with respect to coordinate frame 0 and including a rotation to coordinate frame 1, is applied to the coordinate vector $\boldsymbol{p}^0$ by

$$\boldsymbol{p}^1 = \boldsymbol{R}_0^1 \boldsymbol{p}^0 + \boldsymbol{t}^0 \,. \tag{10}$$

This operation is referred to as a *rigid transformation*.

A composition of multiple rigid transformations of this form would lead to long equations. This can be avoided using a matrix notation which is provided by the concept of *homogeneous transformations* that operate on *homogeneous coordinates*. Homogeneous coordinates allow representing of affine transformations and, more generally, perspective transformations by matrices [95]. A rigid motion is an affine transformation consisting of rotation and translation and can be expressed as homogeneous transformation matrix

$$\boldsymbol{H} = \left[\begin{array}{ccc|c} & \boldsymbol{R} & & \boldsymbol{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \,, \; \boldsymbol{R} \in SO(3) \,, \; \boldsymbol{t} \in \mathbb{R}^3 \,. \tag{11}$$

Since $\boldsymbol{H}$ is simply a different representation of a rigid motion, it is valid to write $\boldsymbol{H} \in SE(3)$. In the context of this thesis, the term homogeneous transformation always refers to a matrix as defined in Eq. (11).

A homogeneous transformation $\boldsymbol{H}_0^1$ transforming from reference frame 0 to reference frame 1 is applied to the homogeneous coordinates $\boldsymbol{P}^0 = [\boldsymbol{p}^0, 1]^T$ by

$$\boldsymbol{P}^1 = \boldsymbol{H}_0^1 \boldsymbol{P}^0 \,. \tag{12}$$

Multiple homogeneous transformations can be composed in the same manner as rotations. Every following rigid motion, referring to the current coordinate frame, is post-multiplied. This leads to the composition rule

$$\boldsymbol{H}_n^0 = \prod_{i=1}^{n} \boldsymbol{H}_i^{i-1} \,. \tag{13}$$

A successive transformation that is executed with respect to fixed axes needs to be pre-multiplied.

3.2 KINEMATICS OF A RIGID BODY SYSTEM

The previous section dealt with the representation of rigid motion for individual bodies. In order to model the motion of an entire human skeleton, we need to consider a multibody system consisting of rigid parts that are interconnected by joints with varying degrees of freedom. To describe the kinematics and the dynamics of such a system, we require

- a geometric model that specifies body dimensions, locations of joints on the bodies and the linkage (topology) between system parts.
- a formalism to describe the mutual influence of connected bodies in consideration of the constraints introduced by the joint linkage.

In other words, a parameterization is needed to derive the position and orientation of each body given the position and orientation of the remaining bodies and the configuration of all joints. A corresponding parameterization yields the Denavit-Hartenberg convention [26]. For a clear description of this convention, the representation of the model topology as a kinematic tree and accompanying notation details are the subject of the next section.

3.2.1 *Kinematic Trees*

The geometry and topology of a skeletal model can be described in terms of a *kinematic tree* in accordance with graph theory. The term *kinematic tree* indicates that the connectivity

graph of the multibody system is a topological tree - free of loops - which is a legitimate approximation for the human skeleton.

To allow an unambiguous description, a distinction between *joints* and *links* is made. The term *joint* denotes a skeletal joint that may have more than one *degree of freedom* (DOF), whereas a *link* refers to a building block of the kinematic tree generated by a set of Denavit-Hartenberg parameters as described below. Each link accounts for one DOF.

The linkage of the considered model can be fully described using *prismatic* and *revolute* links. Both types have only one DOF. While prismatic links allow a translation along the link axis, revolute links allow a rotation about it. These link types are sufficient to create the whole kinematic tree, since all joints with higher DOF, such as free joints (6 DOF) and spherical joints (3 DOF), can be modelled as a sequence of revolute and prismatic links with connections of zero length in between.

The base frame of the kinematic tree is called *root frame* and indexed with 0. The associated link DOF is indexed with 1. It is the first of 3 translational DOF modelled as prismatic links to specify the global position of the model. Subsequently, three rotational DOF, modelled as revolute links, define the global orientation. The coordinate frame attached to a link is always labeled with a decremented index: frame $i-1$ specifies the rigid motion of link i.

The whole tree can be subdivided into kinematic chains without further split-up, e. g. representing the human extremities. The set of all link indices of the kinematic tree is denoted by T and the set of link indices belonging to a kinematic chain is specified by $C \subseteq T$. In order to describe the topology of the kinematic tree, several further quantities have to be defined: For each link j (except for the root), the *predecessor* is the link that is positioned immediately prior to link j in the associated kinematic chain C with $j \in C$. The link directly following link j in C is termed *child* of j. The subset of links on the path from the root to link j, excluding j, is called *support set* $\kappa(j)$ of j. This is also expressed by the phrase: Link k *supports* link j if $k \in \kappa(j)$. The set of links belonging to the subtree starting at link j (with inclusion of j) is called *subtree set* $\mu(j)$. In order to describe the part of a kinematic chain between two links i and j, the term *subchain set* of (i, j) is used and denoted by $\nu(i, j)$. Here, the first bounding link is included in the set, whereas the second bounding link is excluded: $i \in \nu(i, j)$, $j \notin \nu(i, j)$.

3.2.2 *The Denavit-Hartenberg Convention*

In 1955 Jacques Denavit and Richard Hartenberg introduced a formalism to parameterize a kinematic chain using only four parameters for each link. The formalism attaches coordinate frames to each link of the chain and the Denavit-Hartenberg parameters specify the transformation between successive link frames [2]. In general, it requires six degrees

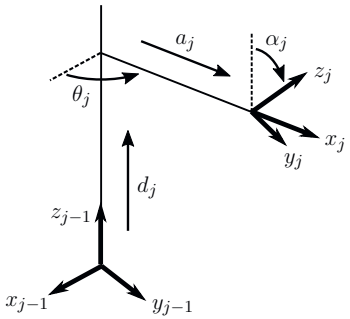


Figure 7: Denavit-Hartenberg transformation from frame $j-1$ to frame j.

of freedom to define the transformation between two coordinate frames. Under certain restrictions, however, it is possible to use fewer parameters. Denavit and Hartenberg use only four parameters to specify the transformation that is invoked by common joint types like revolute and prismatic joints. The necessary conditions for the existence and uniqueness of the resulting transformation are listed at the end of this section.

The Denavit-Hartenberg parameters for the transformation $\boldsymbol{H}_j^{j-1}$ between link frame o_{j-1} and link frame o_j are denoted by $[\theta_j, d_j, \alpha_j, a_j]$ and comprise two rotational and two translational DOF. The coordinate frame $j-1$ can be mapped to frame j by the following consecutive operations: a rotation around the z_{j-1}-axis by the angle θ_j, a translation along the z_{j-1}-axis by the distance d_j, a rotation around the new x_j-axis by the angle α_j and a translation along the x_j-axis by the distance a_j. An example of these operations is illustrated in Figure 7.

Representing the listed operations as homogeneous transformations and using the composition rule of Eq. (12) results in the total transformation matrix for link j:

$$\boldsymbol{H}_j^{j-1} = \boldsymbol{H}_z(\theta_j, d_j)\boldsymbol{H}_x(\alpha_j, a_j) \tag{14}$$

$$= \begin{bmatrix} \boldsymbol{R}_z(\theta_j) & d_j\boldsymbol{e}_z \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_x(\alpha_j) & a_j\boldsymbol{e}_x \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \tag{15}$$

$$= \begin{bmatrix} \boldsymbol{R}_j^{j-1} & \boldsymbol{t}_j^{j-1} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix}. \tag{16}$$

Here, $\boldsymbol{R}_x$ and $\boldsymbol{R}_z$ are the basic rotations around the $\boldsymbol{x}$- and $\boldsymbol{z}$-axis presented in Eq. (3) and (5). The translations are performed along unit vectors $\boldsymbol{e}_x = [1,0,0]^T$ and $\boldsymbol{e}_z = [0,0,1]^T$. As a rigid motion, $\boldsymbol{H}_j^{j-1}$ can be separated into a pure rotation $\boldsymbol{R}_j^{j-1}$ and a pure translation $\boldsymbol{t}_j^{j-1}$ equivalent to Eq. (11).

To obtain the position and orientation of each link j with respect to the root frame, the transformations of all links belonging to the support set $\kappa(j)$ have to be composed:

$$\boldsymbol{H}_j^0 = \prod_{i \in \kappa(j)} \boldsymbol{H}_i^{i-1} \,. \tag{17}$$

Here, a simplified notation is used assuming that all link indices belonging to the considered kinematic chain increase incrementally. This notation will be used in the following sections as well. In the same manner, the rigid motion of a body of the multibody system can be derived given the transformations of its supporting links. In this case, j denotes the link frame attached to the center of mass of the body.

For a general understanding, it is useful to draw the connection to Euler angles [33]. The described parameterization of three consecutive revolute links is equivalent to Euler angles that specify rotations about the current rotating coordinate frame axes. Correspondingly, every spherical joint of the skeletal model is exposed to singularities similar to Euler angles. The implications of singularities for the formulation of dynamical equations and a way of analysing them will be discussed in Section 3.2.3.

Finally, it is worth addressing the topic of existence and uniqueness of Denavit-Hartenberg transformations. In contrast to general rigid motions, only four parameters instead of six are used to define the transformation. This fact naturally restricts the set of representable rigid motions. However, it is possible to formulate assumptions concerning the four Denavit-Hartenberg parameters to guarantee the existence of a unique solution. For a transformation from frame j to frame $j-1$ these assumptions are:

1. The axes z_{j-1} and x_j are perpendicular.
2. The axes z_{j-1} and x_j intersect each other.

A corresponding proof of the existence can be found in [134].

3.2.3 *Velocity and Acceleration Kinematics*

This section addresses the kinematics of the multiybody system in terms of velocity and acceleration. In the previous section, a representation of a kinematic tree by a set of link parameters (Denavit-Hartenberg parameters) was introduced. In general, this set can be divided into fixed parameters that specify constant model dimensions and a set of independent coordinates that describe the motion of the model. The latter are referred to as *generalized coordinates* $\boldsymbol{q}$ in accordance with Lagrangian mechanics. Each generalized coordinate is related to the z-axis of a link in the kinematic tree defining either the translation direction or the axis of rotation. The number of generalized coordinates is equal to the DOF of the rigid body system: $\boldsymbol{q} \in \mathbb{R}^d$.

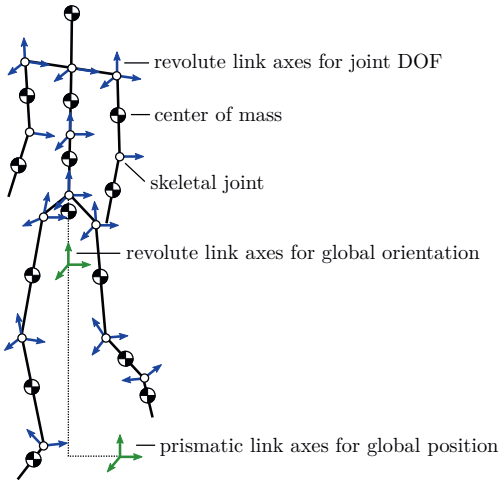


Figure 8: Kinematic tree of the human model with link axes represented by arrows.

Figure 8 shows the skeletal model used in this work to approximate the human locomotor system. The skeletal joints of the model are purely rotational. Therefore, the generalized coordinates are composed of six global coordinates describing the position and the orientation of the pelvis (the first body of the kinematic tree) and a number of rotational DOF modelled as revolute links. The corresponding link axes are depicted as arrows in the figure. The Denavit-Hartenberg convention yields transformations between $\boldsymbol{q}$ and the position and orientation of any link in world coordinates. For the formulation of EOM, the transformation from $\boldsymbol{q}$ to those frames attached to the center of mass of body parts is of particular interest. Consistent with robotics terminology, these frames are often referred to as *end-effector* frames indicating that their rigid motion is sought-for. The partial derivatives of the transformation rule (from $\boldsymbol{q}$ to the rigid motion of end-effectors) with respect to the individual components of $\boldsymbol{q}$ build the Jacobian $\boldsymbol{T}$. This matrix transforms from $\dot{\boldsymbol{q}}$ to the so called body velocity vector $\boldsymbol{\xi} \in \mathbb{R}^{6N}$ composed of the linear and angular velocities of each body in the kinematic tree:

$$\boldsymbol{\xi} := [\boldsymbol{v}_1, \dots, \boldsymbol{v}_N, \boldsymbol{\omega}_1, \dots \boldsymbol{\omega}_N]^T = \boldsymbol{T}\dot{\boldsymbol{q}}\,. \tag{18}$$

Here, N is the number of bodies in the kinematic tree, $\boldsymbol{v}_i$ is the linear velocity of body i and $\boldsymbol{\omega}_i$ the related angular velocity. The connection between generalized and body velocity is

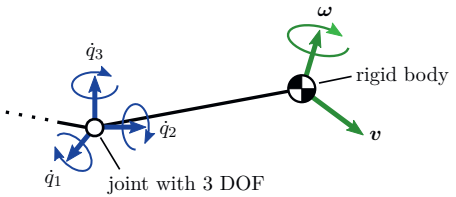


Figure 9: Example of the body velocity $\boldsymbol{\xi} = [\boldsymbol{v}^T, \boldsymbol{\omega}^T]^T$ induced by the generalized velocity $\dot{\boldsymbol{q}} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$. The depicted joint has 3 rotational DOF.

exemplified in Figure 9. Temporal differentiation of Eq. (18) leads to the body acceleration vector

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \frac{d}{dt}\left[\boldsymbol{T}(\boldsymbol{q}(t))\dot{\boldsymbol{q}}(t)\right] \\ &= \boldsymbol{T}\ddot{\boldsymbol{q}} + \frac{\partial}{\partial \boldsymbol{q}}\left(\boldsymbol{T}\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}}\,. \end{aligned} \tag{19}$$

The second term of Eq. (19) is denoted by $\boldsymbol{\zeta}$ and is sometimes referred to as *convective acceleration* in the literature. To abbreviate following derivations, it is written in matrix form as

$$\begin{aligned} \boldsymbol{\zeta}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \frac{\partial}{\partial \boldsymbol{q}}\left(\boldsymbol{T}\dot{\boldsymbol{q}}\right)\dot{\boldsymbol{q}} \\ &= \boldsymbol{G}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\,. \end{aligned} \tag{20}$$

In the further course of this section, the Jacobian $\boldsymbol{T}$ will be presented based on the introduced representation by Denavit-Hartenberg parameters. Based on this, the convective acceleration matrix $\boldsymbol{G}$ will be derived. For this purpose, the special case of a kinematic chain consisting purely of prismatic and revolute links is considered. To lay the foundation for the derivation of $\boldsymbol{G}$, skew symmetric matrices are introduced and the temporal differentiation of rigid motion is described in the following subsections.

Skew Symmetric Matrices

A square matrix $\boldsymbol{S}$ is called *skew symmetric*, if

$$\boldsymbol{S} + \boldsymbol{S}^T = 0\,. \tag{21}$$

In terms of the matrix entries, this is equivalent to

$$s_{ij} = -s_{ji}\,. \tag{22}$$

The set of 3×3 skew symmetric matrices is denoted by $so(3)$. Based on Eq. (22), the matrix $\boldsymbol{S}(\boldsymbol{\omega}) \in so(3)$ can be defined as

$$\boldsymbol{S}(\boldsymbol{a}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{23}$$

with $\boldsymbol{\omega} \in \mathbb{R}^3$.

There are several properties of a skew-symmetric matrix $\boldsymbol{S} \in so(3)$ that are of importance for the derivation of the Jacobian and the convective acceleration matrix:

1. Relation to cross-product: For $\boldsymbol{\omega}, \boldsymbol{r} \in \mathbb{R}^3$

$$\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{r} = \boldsymbol{\omega} \times \boldsymbol{r} \,. \tag{24}$$

2. Similarity transformation: For $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{\omega} \in \mathbb{R}^3$

$$\boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R}^T = \boldsymbol{S}(\boldsymbol{R}\boldsymbol{\omega}) \,. \tag{25}$$

3. Derivative of a rotation matrix: For a rotation $\boldsymbol{R}(\theta) \in SO(3)$, depending on a single variable θ, the derivative is calculated by

$$\frac{d\boldsymbol{R}(\theta)}{d\theta} = \boldsymbol{S}\boldsymbol{R}(\theta) \,. \tag{26}$$

This relation can be specified for a rotation $\boldsymbol{R}_{\boldsymbol{z},\theta} \in SO(3)$ around an arbitrary axis $\boldsymbol{z}$ by angle θ:

$$\frac{d\boldsymbol{R}_{\boldsymbol{z},\theta}}{d\theta} = \boldsymbol{S}(\boldsymbol{z})\boldsymbol{R}_{\boldsymbol{z},\theta} \,. \tag{27}$$

Temporal Differentiation of Rigid Motion

Let $\boldsymbol{\omega}(t)$ be the time dependent angular velocity of a rotating frame. The corresponding rotation matrix $\boldsymbol{R}(t)$ can be differentiated with respect to time using the concept of skew symmetric matrices:

$$\dot{\boldsymbol{R}}(t) = \boldsymbol{S}(\boldsymbol{\omega}(t))\boldsymbol{R}(t) \,. \tag{28}$$

Since $\boldsymbol{\omega}$ is a free vector, it may be expressed in arbitrary coordinates by multiplication with a corresponding rotation matrix. Furthermore, it may be a composition of multiple

angular velocities that can be summed if they are represented in the same coordinates. Therefore, the temporal differentiation of the total rotation of an end-effector frame n can be written as

$$\dot{\boldsymbol{R}}_n^0 = \boldsymbol{S}(\boldsymbol{\omega}_n^0)\boldsymbol{R}_n^0 \tag{29}$$

with

$$\boldsymbol{\omega}_n^0 = \boldsymbol{\omega}_1^0 + \sum_{\substack{i>1 \\ i\in\kappa(n)}} \boldsymbol{R}_{i-1}^0 \boldsymbol{\omega}_i^{i-1} . \tag{30}$$

Here, $\boldsymbol{\omega}_i^{i-1}$ denotes the angular velocity caused by the rotation $\boldsymbol{R}_i^{i-1}$ around axis z_i and expressed in frame $i-1$. The time dependencies are left out for better readability.

To find an expression for linear velocities represented in the moving frame n, the time-dependent homogeneous transformation needs to be considered. It consists of rotation $\boldsymbol{R}_n^0(t)$ and translation $\boldsymbol{t}_n^0(t)$. Point coordinates $\boldsymbol{p}^n$ in the moving frame are transformed as

$$\boldsymbol{p}^0(t) = \boldsymbol{R}_n^0(t)\boldsymbol{p}^n + \boldsymbol{t}_n^0(t) . \tag{31}$$

Temporal differentiation of this equation leads to the linear velocity

$$\begin{aligned} \boldsymbol{v}_n^0 &= \boldsymbol{S}(\boldsymbol{\omega}_n^0)\boldsymbol{R}_n^0\boldsymbol{p}^n + \dot{\boldsymbol{t}}_n^0 && (32)\\ &= \boldsymbol{\omega}_n^0 \times \boldsymbol{r}^0 + \dot{\boldsymbol{t}}_n^0 && (33) \end{aligned}$$

with $\boldsymbol{r}^0 = \boldsymbol{R}_n^0\boldsymbol{p}^n$. The second line is derived using Eq. (24).

The Jacobian and the Convective Acceleration Matrix

The presented relations build the basis for a geometric derivation of the Jacobian $\boldsymbol{T}$ that fulfills Eq. (18). Such a derivation can be found in [134]. Here, merely the result will be presented to focus on the derivation of the convective acceleration matrix $\boldsymbol{G}$. This derivation is hardly ever found in the literature and is conducted using assumptions specific to the present model.

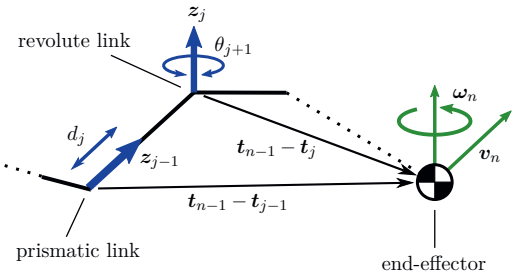


Figure 10: Kinematic subchain with a prismatic and a revolute link that influence the end-effector's rotational velocity $\boldsymbol{\omega}_n$ and linear velocity $\boldsymbol{v}_n$. The positions of the link frames are denoted by $\boldsymbol{t}_{j-1}$ and $\boldsymbol{t}_j$ and the end-effector frame is located at $\boldsymbol{t}_{n-1}$. The prismatic link causes a translation d_j along axis $\boldsymbol{z}_{j-1}$ and the revolute link can be rotated by the angle θ_{j+1} around $\boldsymbol{z}_j$. The dotted lines indicate the continuation of the kinematic chain.

The Jacobian of the considered kinematic chain is composed of a linear and an angular part with

$$\boldsymbol{v}_n = \sum_j \boldsymbol{T}_{v_{nj}} \dot{q}_j \,, \tag{34}$$

$$\boldsymbol{\omega}_n = \sum_j \boldsymbol{T}_{\omega_{nj}} \dot{q}_j \,. \tag{35}$$

$\boldsymbol{T}_{v_{nj}}$ and $\boldsymbol{T}_{\omega_{nj}}$ are 3×1-dimensional column vectors that describe the influence of link coordinate q_j on the end-effector link n with associated frame $n-1$. Every considered end-effector frame is attached to the center of mass of a rigid body of the multibody system. During dynamics simulation these frames will be effected by modelled forces and moments. The Jacobian components are calculated from the Denavit-Hartenberg transformation matrices (cf. Section 3.2.2) using

$$\boldsymbol{T}_{v_{nj}} = \begin{cases} \boldsymbol{z}_{j-1} & j \text{ pri.} \\ \boldsymbol{z}_{j-1} \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{j-1}) & j \text{ rev.} \end{cases}, \tag{36}$$

$$\boldsymbol{T}_{\omega_{nj}} = \begin{cases} \boldsymbol{0} & j \text{ pri.} \\ \boldsymbol{z}_{j-1} & j \text{ rev.} \end{cases} \tag{37}$$

with link $j \in \kappa(n)$ and $\boldsymbol{z}_{j-1}$ denoting the z-axis of the respective link frame and $\boldsymbol{t}_{j-1}$ its origin [134]. The axes and translation vectors are represented in world coordinates, but for better readability the corresponding superscript is omitted. The abbreviations *pri.* and

rev. stand for *prismatic* and *revolute* links. An example of a kinematic chain is depicted in Figure 10 to visualize the used notation.

Similar to the Jacobian, the matrix $\boldsymbol{G}$ is composed of a linear and an angular part: $\boldsymbol{G} = [\boldsymbol{G}_v, \boldsymbol{G}_\omega]^T$. In the following, the acceleration components

$$\boldsymbol{G}_{v_{nj}} = \frac{\partial}{\partial q_j} \left(\boldsymbol{T}_{v_n} \dot{\boldsymbol{q}} \right) , \tag{38}$$

$$\boldsymbol{G}_{\omega_{nj}} = \frac{\partial}{\partial q_j} \left(\boldsymbol{T}_{\omega_n} \dot{\boldsymbol{q}} \right) \tag{39}$$

of end-effector n, caused by a change of link j, will be derived. For clarity, the cases of prismatic and revolute links are treated separately. The results are summarized at the end of this section in Eq. (62).

Starting with the **linear** case the corresponding part of $\boldsymbol{G}$ is

$$\boldsymbol{G}_{v_{nj}} = \frac{\partial}{\partial q_j} \left(\sum_{\substack{i \in \kappa(n) \\ i \text{ pri.}}} \boldsymbol{z}_{i-1} \dot{d}_i + \sum_{\substack{i \in \kappa(n) \\ i \text{ rev.}}} \boldsymbol{z}_{i-1} \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1}) \dot{\theta}_i \right) . \tag{40}$$

Note, that the supporting links i can either be prismatic or revolute regardless of the nature of link j. For all following considerations link j has to support link n in order for the derivative to be unequal zero: $j \in \kappa(n)$. First, let j be **prismatic**; i. e. the generalized coordinate q_j is equal to the Denavit-Hartenberg parameter d_j (cf. Figure 7). Due to the fact that the coordinates of free vectors are translationally invariant, $\frac{\partial}{\partial d_j} \boldsymbol{z}_{i-1} = 0$. Therefore, Eq. (40) simplifies to

$$\boldsymbol{G}_{v_{nj},\text{pri.}} = \sum_{\substack{i \in \kappa(n) \\ i \text{ rev.}}} \boldsymbol{z}_{i-1} \times \frac{\partial}{\partial d_j} (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1}) \dot{\theta}_i \, . \tag{41}$$

The remaining term is only unequal zero, if link j lies on the path between link $i-1$ and end-effector n, i. e. $j \in \nu(i, n)$. Otherwise $(\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})$ is constant. This case does not occur for the considered human model, since all skeletal joints are modelled as a combination of revolute links. The only prismatic links are placed at the beginning of the kinematic chain to describe the global position of the model. Thus, Eq. (41) can be further simplified to

$$\boldsymbol{G}_{v_{nj},\text{pri.}} = \boldsymbol{0} \, . \tag{42}$$

In the case that j is **revolute**, the first term of Eq. (40) can be dropped for the same reason described above: The topology of the model does not include revolute links in the support chain of prismatic links. The remaining component for a revolute link j is

$$\boldsymbol{G}_{v_{nj},\text{rev.}} = \frac{\partial}{\partial\theta_j} \sum_{\substack{i\in\kappa(n)\\ i \text{ rev.}}} \boldsymbol{z}_{i-1} \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})\dot{\theta}_i \tag{43}$$

$$= \frac{\partial}{\partial\theta_j} \sum_{\substack{i\in\kappa(n)\\ i \text{ rev.}}} \boldsymbol{R}^0_{i-1}\boldsymbol{e}_z \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})\dot{\theta}_i \tag{44}$$

with $\boldsymbol{e}_z = [0,0,1]^T$. Now the generalized coordinate q_j is equivalent to the Denavit-Hartenberg angle θ_j. Applying the chain rule for derivatives results in

$$\boldsymbol{G}_{v_{nj},\text{rev.}} = \sum_{\substack{i\in\kappa(n)\\ i \text{ rev.}}} \left[\frac{\partial \boldsymbol{R}^0_{i-1}}{\partial\theta_j}\boldsymbol{e}_z \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})\dot{\theta}_i \right.$$

$$\left. + \boldsymbol{z}_{i-1} \times \frac{\partial(\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})}{\partial\theta_j}\dot{\theta}_i \right] \tag{45}$$

$$=: \boldsymbol{\chi}_1 + \boldsymbol{\chi}_2\,. \tag{46}$$

For further simplification, the two terms of Eq. (45) are considered separately and denoted by $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$, respectively. The first term is only unequal zero, if link i is part of the subtree set of link j written as $i \in \mu(j)$. Only then the rotation $\boldsymbol{R}^0_{i-1}$ depends on θ_j. Together with the requirement that $i \in \kappa(n)$ this is synonymous with $i \in \nu(j,n)$. Simply put, if link i comes after link j and before link n, it lies on the subchain between j and n. Hence, the first term becomes

$$\boldsymbol{\chi}_1 = \sum_{\substack{i\in\nu(j,n)\\ i \text{ rev.}}} \frac{\partial \boldsymbol{R}^0_{i-1}}{\partial\theta_j}\boldsymbol{e}_z \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})\dot{\theta}_i\,. \tag{47}$$

In order to draw a connection between the angle θ_j and the orientation of frame $i-1$, the rotation matrix $\boldsymbol{R}^0_{i-1}$ can be decomposed into the individual rotations occurring along the subchain between j and i:

$$\boldsymbol{R}^0_{i-1} = \boldsymbol{R}^0_j \prod_{k\in\nu(j+1,i)} \boldsymbol{R}^{k-1}_k\,. \tag{48}$$

This is possible because link j supports link i. The derivative of this rotation with respect to θ_j is

$$\begin{aligned}\frac{\partial \boldsymbol{R}_{i-1}^0}{\partial \theta_j} &= \frac{\partial \boldsymbol{R}_j^0}{\partial \theta_j} \prod_{k \in \nu(j+1,i)} \boldsymbol{R}_k^{k-1} \\ &= \boldsymbol{S}(\boldsymbol{z}_{j-1}) \boldsymbol{R}_j^0 \prod_{k \in \nu(j+1,i)} \boldsymbol{R}_k^{k-1} \\ &= \boldsymbol{S}(\boldsymbol{z}_{j-1}) \boldsymbol{R}_{i-1}^0 \,. \end{aligned} \tag{49}$$

Only the leading rotation has to be differentiated, since the rotations starting at link $j+1$ do not change with varying θ_j being represented with respect to the current link coordinate frame. The second line is calculated using Eq. (27) and in the last line, the decomposition of Eq. (48) is reversed. Inserting the result into Eq. (47) and applying Eq. (24) yields

$$\begin{aligned}\boldsymbol{\chi}_1 &= \sum_{\substack{i \in \nu(j,n) \\ i \text{ rev.}}} \boldsymbol{S}(\boldsymbol{z}_{j-1}) \boldsymbol{R}_{i-1}^0 \boldsymbol{e}_z \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1}) \dot{\theta}_i \\ &= \sum_{\substack{i \in \nu(j,n) \\ i \text{ rev.}}} \boldsymbol{z}_{j-1} \times \left[\boldsymbol{z}_{i-1} \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1}) \right] \dot{\theta}_i \,. \end{aligned} \tag{50}$$

The second term $\boldsymbol{\chi}_2$ of Eq. (45) is only different from zero if the translation $(\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})$ depends on θ_j. This is the case, if link j lies between the links i and n, so that i also supports j: $i \in \kappa(j)$. Hence, the term can be rewritten as

$$\boldsymbol{\chi}_2 = \sum_{\substack{i \in \kappa(n) \cap \kappa(j) \\ i \text{ rev.}}} \boldsymbol{z}_{i-1} \times \frac{\partial (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})}{\partial \theta_j} \dot{\theta}_i \,. \tag{51}$$

Given the specified order of links, it is sufficient to calculate $\frac{\partial \boldsymbol{t}_{n-1}}{\partial \theta_j}$. Using Eq. (10), the translation can be expressed in terms of coordinate frames $j-1$ and j as

$$\boldsymbol{t}_{n-1} = \boldsymbol{R}_j^0 \boldsymbol{t}_{n-1}^j + \boldsymbol{R}_{j-1}^0 \boldsymbol{t}_j^{j-1} + \boldsymbol{t}_{j-1} \,. \tag{52}$$

The derivative of this equation with respect to θ_j is

$$\begin{aligned}\frac{\partial \boldsymbol{t}_{n-1}}{\partial \theta_j} &= \frac{\partial \boldsymbol{R}_j^0}{\partial \theta_j} \boldsymbol{t}_{n-1}^j + \boldsymbol{R}_{j-1}^0 \frac{\partial \boldsymbol{t}_j^{j-1}}{\partial \theta_j} \\ &= \boldsymbol{S}(\boldsymbol{z}_{j-1}) \boldsymbol{R}_j^0 \boldsymbol{t}_{n-1}^j + \boldsymbol{S}(\boldsymbol{z}_{j-1}) \boldsymbol{R}_{j-1}^0 \boldsymbol{t}_j^{j-1} \qquad (53) \\ &= \boldsymbol{S}(\boldsymbol{z}_{j-1}) \left(\boldsymbol{R}_j^0 \boldsymbol{t}_{n-1}^j + \boldsymbol{R}_{j-1}^0 \boldsymbol{t}_j^{j-1} \right) \\ &= \boldsymbol{S}(\boldsymbol{z}_{j-1}) \left(\boldsymbol{t}_{n-1} - \boldsymbol{t}_{j-1} \right) \,. \qquad (54) \end{aligned}$$

The second term of Eq. (53) is derived using the Denavit-Hartenberg representation $\boldsymbol{t}_j^{j-1} = [a_j \cos\theta_j, a_j \sin\theta_j, d_j]^T$:

$$\begin{aligned}
\boldsymbol{R}_{j-1}^0 \frac{\partial \boldsymbol{t}_j^{j-1}}{\partial \theta_j} &= \boldsymbol{R}_{j-1}^0 [-a_j \sin\theta_j, a_j \cos\theta_j, 0]^T \\
&= \boldsymbol{R}_{j-1}^0 \boldsymbol{S}(\boldsymbol{e}_z) \boldsymbol{t}_j^{j-1} \\
&= \boldsymbol{R}_{j-1}^0 \boldsymbol{S}(\boldsymbol{e}_z) (\boldsymbol{R}_{j-1}^0)^T \boldsymbol{R}_{j-1}^0 \boldsymbol{t}_j^{j-1} \\
&= \boldsymbol{S}(\boldsymbol{R}_{j-1}^0 \boldsymbol{e}_z) \boldsymbol{R}_{j-1}^0 \boldsymbol{t}_j^{j-1} \\
&= \boldsymbol{S}(z_{j-1}) \boldsymbol{R}_{j-1}^0 \boldsymbol{t}_j^{j-1}
\end{aligned} \tag{55}$$

Inserting Eq. (54) into Eq. (51) and applying Eq. (24) yields

$$\begin{aligned}
\boldsymbol{\chi}_2 &= \sum_{\substack{i \in \kappa(n) \cap \kappa(j) \\ i \text{ rev.}}} \boldsymbol{z}_{i-1} \times \boldsymbol{S}(\boldsymbol{z}_{j-1})(\boldsymbol{t}_{n-1} - \boldsymbol{t}_{j-1}) \dot{\theta}_i \\
&= \sum_{\substack{i \in \kappa(n) \cap \kappa(j) \\ i \text{ rev.}}} \boldsymbol{z}_{i-1} \times \left[\boldsymbol{z}_{j-1} \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{j-1})\right] \dot{\theta}_i \,.
\end{aligned} \tag{56}$$

Finally, the sum of $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$, noted in Eq. (50) and Eq. (56), results in the sought-after acceleration component

$$\begin{aligned}
\boldsymbol{G}_{v_{nj},\text{rev.}} = &\sum_{\substack{i \in \nu(j,n) \\ i \text{ rev.}}} \boldsymbol{z}_{j-1} \times \left[\boldsymbol{z}_{i-1} \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{i-1})\right] \dot{\theta}_i \\
&+ \sum_{\substack{i \in \kappa(n) \cap \kappa(j) \\ i \text{ rev.}}} \boldsymbol{z}_{i-1} \times \left[\boldsymbol{z}_{j-1} \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_{j-1})\right] \dot{\theta}_i \,.
\end{aligned} \tag{57}$$

Following a similar approach, the **angular** part of $\boldsymbol{G}$ can be derived. Based on the Jacobian components of Eq. (37), the acceleration component is

$$\boldsymbol{G}_{\omega_{nj}} = \frac{\partial}{\partial q_j} \sum_{\substack{i \in S_n \\ i \text{ rev.}}} \boldsymbol{z}_{i-1} \dot{\theta}_i \,. \tag{58}$$

As before, we distinguish between prismatic and revolute links. If j is **prismatic**, the term vanishes due to $\frac{\partial}{\partial d_j} \boldsymbol{z}_{i-1} = 0$. Thus, we get

$$\boldsymbol{G}_{\omega_{nj},pri.} = \boldsymbol{0} \,. \tag{59}$$

If j is **revolute**, Eq. (58) can be written as

$$\begin{aligned}\boldsymbol{G}_{\omega_{nj},rev.} &= \frac{\partial}{\partial\theta_j}\sum_{\substack{i\in S_n\\ i\text{ rev.}}}\boldsymbol{z}_{i-1}\dot{\theta}_i \\ &= \sum_{\substack{i\in S_n\\ i\text{ rev.}}}\frac{\partial\boldsymbol{R}_{i-1}^0}{\partial\theta_j}\boldsymbol{e}_z\dot{\theta}_i \\ &= \sum_{\substack{i\in\nu(j,n)\\ i\text{ rev.}}}\boldsymbol{S}(\boldsymbol{z}_{j-1})\boldsymbol{R}_{i-1}^0\boldsymbol{e}_z\dot{\theta}_i\,. \end{aligned} \tag{60}$$

The rotation $\boldsymbol{R}_{i-1}^0$ is partially differentiated according to Eq. (49). Using Eq. (24) we get the final result

$$\boldsymbol{G}_{\omega_{nj},\text{rev.}} = \boldsymbol{z}_{j-1}\times\sum_{\substack{i\in\nu(j,n)\\ i\text{ rev.}}}\boldsymbol{z}_{i-1}\dot{\theta}_i\,. \tag{61}$$

To summarize, the derived formulas for the convective acceleration matrix are

$$\begin{aligned}\boldsymbol{G}_{v_{nj},\text{pri.}} &= \boldsymbol{0}\,,\\ \boldsymbol{G}_{\omega_{nj},\text{pri.}} &= \boldsymbol{0}\,,\\ \boldsymbol{G}_{v_{nj},\text{rev.}} &= \sum_{\substack{i\in\nu(j,n)\\ i\text{ rev.}}}\boldsymbol{z}_{j-1}\times\left[\boldsymbol{z}_{i-1}\times(\boldsymbol{t}_{n-1}-\boldsymbol{t}_{i-1})\right]\dot{\theta}_i\\ &\quad+\sum_{\substack{i\in\kappa(n)\cap\kappa(j)\\ i\text{ rev.}}}\boldsymbol{z}_{i-1}\times\left[\boldsymbol{z}_{j-1}\times(\boldsymbol{t}_{n-1}-\boldsymbol{t}_{j-1})\right]\dot{\theta}_i\,,\\ \boldsymbol{G}_{\omega_{nj},\text{rev.}} &= \boldsymbol{z}_{j-1}\times\sum_{\substack{i\in\nu(j,n)\\ i\text{ rev.}}}\boldsymbol{z}_{i-1}\dot{\theta}_i\,. \end{aligned} \tag{62}$$

It can be seen that all prismatic components vanish which is consistent with the constraints of the considered multibody system. The convective acceleration represents the part of the acceleration caused by skeletal constraints. Since the used model is only constrained by revolute links, only the corresponding components of $\boldsymbol{G}$ are unequal zero.

Singularities of the Jacobian

A singularity is a joint configuration $\boldsymbol{q}$ at which the Jacobian $\boldsymbol{T}(\boldsymbol{q})$ is rank deficient, i. e. the number of independent rows or columns is smaller than the maximum number reached at different configurations. At a singularity one or several of the following circumstances applies:

1. Specific directions of motion are not possible starting from a singularity.

2. A finite change in end-effector velocity is associated with an infinite change in joint velocity $\dot{\boldsymbol{q}}$.

3. A finite change of end-effector forces and moments can only be realized by infinite change of joint moments.

4. Near singularities there may not exist a unique solution to the inverse kinematics problem.

The described effects of singularities emphasize the importance of their investigation. In order to analyze the occurrence of singularities of $\boldsymbol{T}$, the matrix is considered in parts $\boldsymbol{T}_c$, representing subchains of the kinematic tree. If $\boldsymbol{T}_c$ is a square matrix, a singularity can be identified using

$$\det(\boldsymbol{T}_c) = 0\,. \tag{63}$$

In the present model, the number of rotational links per skeletal joint is $k <= 3$. Let $k = 3$ and the link indices $i = 1, 2, 3$, w.l.o.g.. According to Eq. (36) and Eq. (37), the linear and angular parts of the Jacobian are

$$\begin{aligned} \boldsymbol{T}_{c_v} &= \begin{bmatrix} \boldsymbol{z}_0 \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_0) & \boldsymbol{z}_1 \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_1) & \boldsymbol{z}_2 \times (\boldsymbol{t}_{n-1} - \boldsymbol{t}_2) \end{bmatrix}, \\ \boldsymbol{T}_{c_\omega} &= \begin{bmatrix} \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{z}_2 \end{bmatrix}. \end{aligned} \tag{64}$$

Since the link origins coincide, the translations are set to $\boldsymbol{t}_{n-1} - \boldsymbol{t}_0 = \boldsymbol{t}_{n-1} - \boldsymbol{t}_1 = \boldsymbol{t}_{n-1} - \boldsymbol{t}_2 = \boldsymbol{t}'$:

$$\begin{aligned} \boldsymbol{T}_{c_v} &= \begin{bmatrix} \boldsymbol{z}_0 \times \boldsymbol{t}' & \boldsymbol{z}_1 \times \boldsymbol{t}' & \boldsymbol{z}_2 \times \boldsymbol{t}' \end{bmatrix}, \\ \boldsymbol{T}_{c_\omega} &= \begin{bmatrix} \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{z}_2 \end{bmatrix}. \end{aligned} \tag{65}$$

Both matrices are rank deficient, if two link axes are collinear. This result is in correspondence with the singularities of Euler angles [33]. The de facto loss of one DOF at the singularity is also called *gimbal lock*. Due to the nature of the considered locomotion movements including only joint angles smaller than $\pi/2$ the gimbal lock does not affect the calculation and optimization of kinematics and dynamics in this work.

3.3 DYNAMICS OF A RIGID BODY SYSTEM

A dynamical consideration of a system of rigid bodies deals with the evaluation of an EOM for this system. The equation yields the interrelationship between the motion of

the system and the active forces. It can either be used to predict motion trajectories on the basis of applied forces or to find underlying forces based on the observed motion. These two procedures are termed *forward* and *inverse dynamics*, respectively. This section addresses the formulation of EOM for constrained rigid body systems using the so-called *TMT-method* [125].

3.3.1 *TMT-Method*

The TMT-method offers a practical approach to formulating EOM for multibody systems combining concepts of Newtonian and Lagrangian mechanics. The name of the method is inspired by matrix multiplications in the resulting EOM, as will become clear below. In order to describe a multibody system using the Newton-Euler equations, they have to be extended to differential algebraic equations to incorporate kinematic constraints. The solution of these equations using numerical integration, however, is a complex process that requires the initial states to match the constraints and involves either a transformation to ordinary differential equations or large-scale numerical solvers [120]. In contrast to that, the Euler-Lagrange equation considers kinetic and potential energy of the system and the EOM is derived for a minimum set of independent generalized coordinates. For this purpose, however, all arising energies have to be identified and their derivatives have to be calculated which can cause tremendous computational effort when dealing with complex models. To avoid the disadvantages and exploit the benefits of both concepts, the TMT-method uses a *force approach* similar to Newton-Euler, but incorporates the kinematic constraints in a transformation $\boldsymbol{T} \in \mathbb{R}^{6N \times d}$ from independent generalized velocities $\dot{\boldsymbol{q}} \in \mathbb{R}^{d}$ to mutually dependent body velocities $\boldsymbol{\xi} \in \mathbb{R}^{6N}$. This transformation is the Jacobian introduced in Section 3.2.3.

The representation of body velocities $\boldsymbol{\xi}$ and accelerations $\dot{\boldsymbol{\xi}}$ in terms of generalized coordinates $\boldsymbol{q}$ has been introduced in Eq. (18) and Eq. (19), respectively. To allow for a better understanding of the following derivation the definition is repeated here:

$$\boldsymbol{\xi} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_N, \boldsymbol{\omega}_1, \dots \boldsymbol{\omega}_N]^T = \boldsymbol{T}\dot{\boldsymbol{q}}\,, \tag{66}$$

$$\dot{\boldsymbol{\xi}} = \boldsymbol{T}\ddot{\boldsymbol{q}} + \frac{\partial}{\partial \boldsymbol{q}} \left(\boldsymbol{T}\dot{\boldsymbol{q}}\right) \dot{\boldsymbol{q}}\,. \tag{67}$$

The starting point of the TMT-method is Newton's law

$$\boldsymbol{M}\dot{\boldsymbol{\xi}} - \boldsymbol{f} = \boldsymbol{0}\,. \tag{68}$$

The force $\boldsymbol{f}$ is the sum of all conservative and non-conservative forces acting on the system. The inertia matrix $\boldsymbol{M} \in \mathbb{R}^{6N \times 6N}$ is of diagonal shape. Every linear component of $\dot{\boldsymbol{\xi}}$ is

multiplied with the mass value $m_i \in \mathbb{R}$ and every rotational component with the tensor of inertia $\boldsymbol{I}_i \in \mathbb{R}^{3\times3}$ of the corresponding rigid body i in the kinematic tree:

$$\boldsymbol{M} = \mathrm{diag}(m_1, m_1, m_1, \ldots, m_N, m_N, m_N, \boldsymbol{I}_1, \ldots, \boldsymbol{I}_N)\,. \tag{69}$$

Every tensor of inertia needs to be transformed from the moving end-effector frame n, attached to body i, to the world frame by means of similarity transformation according to Eq. (8):

$$\boldsymbol{I}_i = \boldsymbol{R}_n^0 \boldsymbol{I}_i^n (\boldsymbol{R}_n^0)^T\,. \tag{70}$$

The further derivation of the method is based on the *principle of virtual work* which allows for the formulation of EOM for a multibody system under constraints. The principle states that the virtual work performed by the forces effecting a mechanical system is zero for all virtual displacements from static equilibrium [147]. The TMT-method uses an alternative approach by applying virtual velocities that satisfy the system constraints and considering the virtual work rate which is required to be zero as well [116]. The dot product of Eq. (68) with the virtual body velocitiy $\delta\boldsymbol{\xi}$ yields the virtual work rate of the system:

$$\delta\boldsymbol{\xi}^T \left(\boldsymbol{M}\dot{\boldsymbol{\xi}} - \boldsymbol{f}\right) = 0 \tag{71}$$

$$\Leftrightarrow (\boldsymbol{T}\delta\dot{\boldsymbol{q}})^T \left(\boldsymbol{M}\dot{\boldsymbol{\xi}} - \boldsymbol{f}\right) = 0\,. \tag{72}$$

In Eq. (72), the virtual body velocities are expressed as a function of virtual generalized velocities $\delta\dot{\boldsymbol{q}}$ using the transformation of Eq. (66). Since $\delta\dot{\boldsymbol{q}}$ are independent (representing the DOF of the system), the vanishing dot product of Eq. (72) is equivalent to the vector equation

$$\boldsymbol{T}^T \left(\boldsymbol{M}\dot{\boldsymbol{\xi}} - \boldsymbol{f}\right) = \boldsymbol{0}\,. \tag{73}$$

Substituting the body acceleration of Eq. (67) in Eq. (73) and rearranging yields the EOM

$$\boldsymbol{T}^T\boldsymbol{M}\boldsymbol{T}\ddot{\boldsymbol{q}} = \boldsymbol{T}^T\left(\boldsymbol{f} - \boldsymbol{M}\boldsymbol{\zeta}\right) \tag{74}$$

where $\boldsymbol{\zeta}$ is the convective acceleration introduced in Eq. (20).

Defining the *reduced*[1] *inertia matrix* $\boldsymbol{\mathcal{M}} = \boldsymbol{T}^T\boldsymbol{M}\boldsymbol{T}$ and the *reduced force* $\boldsymbol{\mathcal{F}} = \boldsymbol{T}^T(\boldsymbol{f} - \boldsymbol{M}\boldsymbol{\zeta})$, the EOM can be written in brief terms as

$$\boldsymbol{\mathcal{M}}\ddot{\boldsymbol{q}} = \boldsymbol{\mathcal{F}}\,. \tag{75}$$

The composition of the reduced inertia matrix is responsible for the naming of the method.

In the considered case of a skeletal model, the total force $\boldsymbol{f}$ consists of joint torques $\boldsymbol{\tau}$, which represent the generalized forces, a combined contact force and moment vector $\boldsymbol{f}_c$ and the gravitational force $\boldsymbol{M}\boldsymbol{g}$. While the joint torques directly effect the generalized coordinates, the remaining forces are applied to the centers of mass of the individual bodies and therefore need to be represented in terms of body coordinates. The resulting EOM is

$$\boldsymbol{\mathcal{M}}\ddot{\boldsymbol{q}} = \boldsymbol{\tau} + \boldsymbol{T}^T(\boldsymbol{f}_c + \boldsymbol{M}(\boldsymbol{g} - \boldsymbol{\zeta}))\,. \tag{76}$$

The vector $\boldsymbol{f}_c \in \mathbb{R}^{6N}$ is composed of three dimensional contact forces $\boldsymbol{f}_{c_i}$ and three dimensional contact moments $\boldsymbol{m}_{c_i}$ for all bodies $i = 1, \ldots, N$ of the model:

$$\boldsymbol{f}_c = \begin{bmatrix} \boldsymbol{f}_{c_1} \\ \vdots \\ \boldsymbol{f}_{c_N} \\ \boldsymbol{m}_{c_1} \\ \vdots \\ \boldsymbol{m}_{c_N} \end{bmatrix}\,. \tag{77}$$

The application of the contact forces and joint moments is exemplified in Figure 11. The gravitational acceleration vector $\boldsymbol{g}$ contains $g = -9.81\,m/s^2$ at all components corresponding to vertical linear accelerations. All other components are equal to zero.

3.4 MACHINE LEARNING

This chapter deals with fundamental concepts of machine learning and introduces algorithms used to realize learning-based inverse dynamics estimation. Following the introduction of the necessary terminology, the chapter includes a presentation of *support vector machines*, *ridge regression*, *random forests* and *neural networks*.

1 The term *reduced* refers to the reduction to a minimal set of independent generalized coordinates.

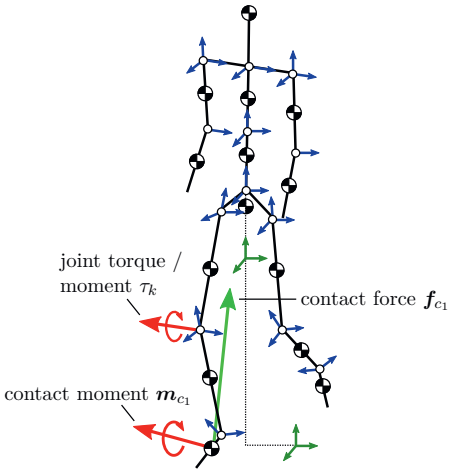


Figure 11: Skeletal model with contact force $\boldsymbol{f}_{c_1}$, contact moment $\boldsymbol{m}_{c_1}$ and the sagittal knee moment τ_k as an example for a joint torque component.

3.4.1 *Terminology and General Concepts*

In machine learning we distinguish between *supervised* and *unsupervised learning*. In supervised learning the model function is fitted using a training set of annotated data. In contrast, unsupervised learning is applied to data without annotations. In this thesis two basic approaches of machine learning are used, namely *classification* and *regression*. Both tasks are part of supervised learning.

A classifier is defined as a function $f_c : \mathbb{R}^m \to \mathbb{N}, \boldsymbol{x} \mapsto y$. The vector $\boldsymbol{x}$ is either a raw datum or a feature vector that has been extracted from the datum prior to the classification. The number y denotes the *class*, the datum is assigned to. While classification assigns a class label (an integer) to the feature vector, a regression function f_r maps to a vector of real numbers $\boldsymbol{y}$ of the target data type, such that $f_r : \mathbb{R}^m \to \mathbb{R}^n, \boldsymbol{x} \mapsto \boldsymbol{y}$. In this work, for instance, the designed regression functions map from motion features to force and moment features.

3.4.2 *Support Vector Machines*

A *support vector machine* (SVM) is a classification method that finds a hyperplane in the feature space, which separates the classes from each other. A hyperplane is given by

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = 0 \tag{78}$$

with the normal vector $\boldsymbol{w}$ and the position w_0 [8]. If $\boldsymbol{w}$ is of unit length, then w_0 is the distance between the plane and the coordinate origin. Let $\{\boldsymbol{x}_i\}$ with $i = 1, \ldots, N$ be the training feature vectors belonging to two classes S_1 and S_2. The classes are linearly separable if a hyperplane can be found that fully separates the sets. Such a hyperplane, however, is not unique. The algorithm chooses a hyperplane by maximizing the margins to both classes, so that the risk of misclassifying unseen data is minimized. The distance of a point $\boldsymbol{x}$ to the hyperplane is

$$z = \frac{|g(\boldsymbol{x})|}{\|\boldsymbol{w}\|}\,. \tag{79}$$

By scaling the parameters $\boldsymbol{w}$ and w_0, the value of $g(\boldsymbol{x})$ can be set to 1 and -1 at the nearest points belonging to S_1 and S_2, respectively. This way, the total margin is $\frac{2}{\|w\|}$ and the following conditions hold:

$$\begin{aligned} &\boldsymbol{w}^T\boldsymbol{x} + w_0 \geq 1\,,\ \forall \boldsymbol{x} \in S_1\,,\\ &\boldsymbol{w}^T\boldsymbol{x} + w_0 \leq -1\,,\ \forall \boldsymbol{x} \in S_2\,. \end{aligned} \tag{80}$$

To determine the hyperplane parameters, the maximization of the margin $\frac{2}{\|w\|}$ is implemented as the equivalent quadratic optimization task

$$\begin{aligned} &\min_{\boldsymbol{w}, w_0} \left\{\frac{1}{2}\|\boldsymbol{w}\|^2\right\}\\ &\text{s.t. } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1\,,\ i = 1, \ldots, N\\ &\qquad y_i = \operatorname{sgn}(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \end{aligned} \tag{81}$$

with $y_i \in \{-1, 1\}$ indicating the class of the training feature vectors. The resulting classifier has the same form

$$f_c(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^T\boldsymbol{x} + w_0)\,. \tag{82}$$

An example of two-dimensional linearly separable data points classified using an SVM is illustrated in Figure 12.

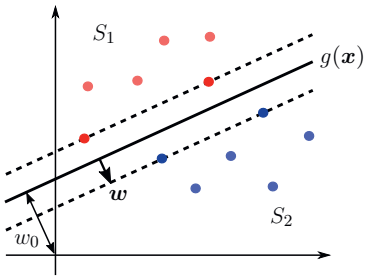


Figure 12: Two sets S_1 and S_2 separated by the hyperplane $g(\boldsymbol{x})$. The data points lying on the margin (dashed lines) represent the support vectors used to determine the hyperplane parameters $\boldsymbol{w}$ and w_0.

The case of linearly separable classes is highly improbable in real-world scenarios, due to outliers or naturally overlapping classes. Therefore, an extension of the algorithm allows for a violation of the conditions in Eq. (80). This is realized by introducing *slack variables* $\xi_i \geq 0$ that are equal to the respective constraint violation. The new optimization task is

$$\begin{aligned} &\min_{\boldsymbol{w},w_0,\boldsymbol{\xi}} \left\{ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \xi_i \right\} \\ &\text{s.t. } y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 - \xi_i\,,\ i = 1,\ldots,N\,. \end{aligned} \tag{83}$$

The positive constant C is used to balance both terms.

In practise, an SVM solves the dual-problem of the presented optimization problem. The dual form can be derived using Lagrange multipliers $\boldsymbol{\lambda}$ and Karush-Kuhn-Tucker conditions [141]. The resulting problem is

$$\begin{aligned} &\max_{\boldsymbol{\lambda}} \left\{ \sum_{i=1}^{N} \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i\lambda_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j \right\} \\ &\text{s.t. } 0 \leq \lambda_i \leq C\,,\ i = 1,\ldots,N \text{ and } \sum_{i=1}^{N} \lambda_i y_i = 0\,. \end{aligned} \tag{84}$$

The parameter $\boldsymbol{w}$ of the optimal hyperplane is a linear combination of all $N_s \leq N$ feature vectors with Lagrange multipliers $\lambda_i \neq 0$:

$$\boldsymbol{w} = \sum_{i=1}^{N_s} \lambda_i y_i \boldsymbol{x}_i\,. \tag{85}$$

These feature vectors are called *support vectors* and reside either on the parallel hyperplanes $\boldsymbol{w}^T\boldsymbol{x} + w_0 = \pm 1$ or within the margin (if $\xi_i > 0$). The Lagrange multipliers of all points

lying within the margin take the maximum value $\lambda_i = C$ and thus maximally influence the solution.

A further extension to the SVM is based on the idea to map the data to a higher dimensional space where a linear separation is more likely. Let $\phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}, \boldsymbol{x} \mapsto \phi(\boldsymbol{x})$ be a mapping from dimension d_1 to dimension d_2. Since the objective of Eq. (84) contains feature vectors only in terms of dot products, a *kernel trick* can be applied: The mapping is efficiently included into the optimization problem using a *kernel function*

$$k(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^T \phi(\boldsymbol{y}) \tag{86}$$

that calculates the dot product in the higher dimensional space without explicit transformation to d_2. The resulting hyperplane normal $\boldsymbol{w} = \sum_{i=1}^{N_s} \lambda_i y_i \phi(\boldsymbol{x}_i)$ yields a classifier of the form

$$f_c(x) = \operatorname{sgn}(\boldsymbol{w}^T \phi(\boldsymbol{x}) + w_0) = \operatorname{sgn}\left(\sum_{i=1}^{N_s} \lambda_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) + w_0 \right) . \tag{87}$$

The objective function of an SVM's optimization problem is convex in both, $\boldsymbol{w}$ and w_0, which enables an efficient solution, e. g. by sub-gradient decent [141]. Furthermore, the linear separation of classes is less prone to overfitting, than more complex methods. Due to these properties, the SVM represents a frequently used classifier. In this work, it is one of the applied methods to classify gait phases as part of the multi-stage regression approach that is subject of Chapter 5.

3.4.3 *Ridge Regression*

A basic regression method applied in this work is linear regression with Tikhonov regularization also called *ridge regression* [57]. It produces a linear function

$$y = \boldsymbol{x}^T \boldsymbol{w} \tag{88}$$

that maps a feature vector $\boldsymbol{x}$ to the regression value y by solving a linear least squares problem of the form

$$\min_{\boldsymbol{w}} \left\{ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \alpha \|\boldsymbol{w}\|_2^2 \right\} . \tag{89}$$

Here, $\boldsymbol{y}$ are the targets, $\boldsymbol{X}$ are the features arranged in a matrix and $\boldsymbol{w}$ are the estimation coefficients that define the regression function. The L_2-regularization of the coefficients penalizes large values of $\boldsymbol{w}$ and allows an analytical solution of the linear least squares problem despite multicollinear variables in $\boldsymbol{X}$. The factor $\alpha \geq 0$ adjusts its strength. Unlike

the ordinary least squares estimator, the ridge estimator eliminates multicollinearity by adding a scaled unit matrix $\alpha \boldsymbol{I}$ to guarantee the existence of the inverse in

$$\begin{aligned} &(\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{I})\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y} \\ &\Rightarrow \boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}\,. \end{aligned} \tag{90}$$

This formulation is called *normal equation* and yields a one-step analytical solution of the problem with minimal squared error. A further approach is to solve the optimization problem in Eq. (89) using gradient descent which is computationally advantageous if $\boldsymbol{X}$ is very large.

3.4.4 *Random Forests*

A *random forest* (RF) [12] is an ensemble method for classification and regression that combines the predictions of multiple base estimators, in this case, *decision trees*. Prior to the description of the RF as an ensemble, the principle and construction of the individual decision trees is presented.

Decision Trees

A decision tree is a nonlinear classifier that reaches a decision by successively excluding possible classes. Most commonly, decision trees are implemented as binary trees, whose nodes split the feature space into hyperrectangles using threshold functions as illustrated in Figure 13. For example, the range of values of feature x_i is split by $x_i \leq \alpha \in \mathbb{R}$. The application of a decision tree to a feature vector can be illustrated as the propagation of the vector through the tree. At each node the corresponding decision function determines the direction of further propagation. The terminal node, also called *leave*, assigns a class or a real number (in the case of regression) to the feature vector.

For the construction of a decision tree, various methods have been developed. One popular algorithm is CART (Classification and Regression Trees) by Breiman et al. [13]. The algorithm generates binary trees by choosing the feature and threshold at each node to achieve maximal information gain. The target variables can be numerical allowing for the construction of regression trees. Mathematically expressed, CART recursively partitions a set $Q = \{(\boldsymbol{x}^{(i)}, y_i)\}_{i=1}^m$ consisting of training vectors $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$ and target variables y_i into disjoint subsets Q_{left} and Q_{right}. The partition is done using a split tuple $\theta = (j, t)$ consisting of feature index j and threshold t. The resulting subsets are

$$Q_{\text{left}}(\theta) = \{(\boldsymbol{x}^{(i)}, y_i) | x_j^{(i)} \leq t\}_{i=1}^m\,, \tag{91}$$

$$Q_{\text{right}}(\theta) = Q \backslash Q_{\text{left}}(\theta)\,. \tag{92}$$

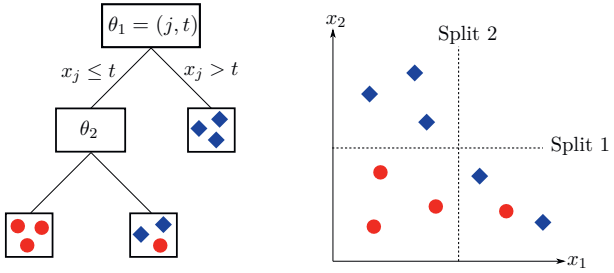


Figure 13: Example of a decision tree that classifies two-dimensional data into two classes. The tree structure is shown in the left and the feature space with resulting decision boundaries on the right.

For each partition, the goal is to reduce the impurity in the subsets. Depending on the task, i. e. classification or regression, different impurity functions $H()$ are deployed to calculate the weighted impurity of the split

$$G(Q, \theta) = \frac{m_{\text{left}}}{m} H(Q_{\text{left}}(\theta)) + \frac{m_{\text{right}}}{m} H(Q_{\text{right}}(\theta)) \,. \tag{93}$$

Here, m_{left} and m_{right} are the respective sample numbers in the subsets. The split θ is determined by

$$\min_{\theta} G(Q, \theta) \tag{94}$$

and the partitioning is recursively continued until a stopping criterion is reached. Common stopping criteria are the maximum tree depth and the minimum number of samples in the leave nodes. Possible impurity functions for the construction of a classification tree are *Entropy* and *Gini-Index*:

$$H_{\text{Entropy}}(Q) = -\sum_k p_k \log(p_k) \,, \tag{95}$$

$$H_{\text{Gini}}(Q) = \sum_k p_k (1 - p_k) \,. \tag{96}$$

The probability p_k that a sample of Q belongs to class k is approximated by the proportion of samples with the respective label:

$$p_k = \frac{1}{m} \sum_{i=1}^{m} I_k(y_i) \,, \;\; I_k(y) = \begin{cases} 1 & y = k \\ 0 & y \neq k \end{cases} \,. \tag{97}$$

For a regression tree, typical impurity functions are the *mean squared error* (MSE) and the *mean absolute error* (MAE) of the target values y_i with respect to the mean and the median of all target values in Q:

$$H_{\text{MSE}}(Q) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y})^2\,, \quad \bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i\,, \tag{98}$$

$$H_{\text{MAE}}(Q) = \frac{1}{m}\sum_{i=1}^{m}|y_i - \underset{i=1,\ldots,m}{\text{median}}(y_i)|\,. \tag{99}$$

A further element of CART is *minimal cost-complexity pruning*. Post-pruning of trees is an important step to avoid overfitting. Unnecessary subtrees that result from noise in the dataset are retrospectively cut. In minimal cost-complexity pruning, the cost-complexity measure of a tree T is given by

$$R_\alpha(T) = R(T) + \alpha|T|\,. \tag{100}$$

The measure contains the misclassification rate $R(T)$ of the tree and a parameter $\alpha \geq 0$ that includes the complexity in terms of the number of leaves $|T|$. Minimization of $R_\alpha(T)$ yields the subtree with minimal cost-complexity.

Combining Base Estimators

Ensemble methods can be roughly separated into averaging methods (e. g. *bagging*) that average over the predictions of independently constructed base estimators and *boosting* methods that built every additional estimator with regard to the bias of the entire ensemble.

In this work, *bagging* [11] is used to build an RF consisting of diverse decision trees. The RF prediction is the average of the individual tree predictions. Bagging incorporates two sources of randomness: random subsets of data and random subsets of features. For the construction of each tree, a random subset of the training set is chosen with replacement. This way, the effect of outliers and inaccurate training samples is reduced. The second source of randomness concerns the splitting of nodes. During the construction of trees, a random subset of features is used to find the best split at each node.

Both sources of randomness support the diversity of trees and lead to an ensemble estimator with reduced variance, i. e. reduced sensitivity to small changes in the training set. The decrease in variance is often combined with a slight increase of the bias which is the difference between the expectation value of the model and the true value. In general, this increase is negligible compared to the reduced variance resulting in a superior estimator.

An RF estimator provides a number of advantages compared to other estimators:

- Features of variable nature can be processed.

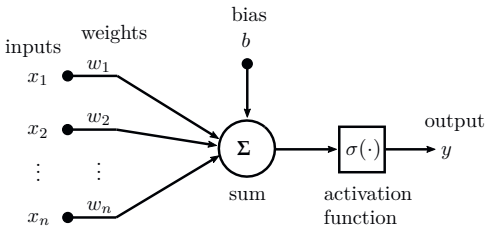


Figure 14: Schematic of an artificial neuron receiving n input signals and generating an output signal y according to Eq. (101).

- No data normalization is necessary to regulate the feature space.
- The processing of data is transparent.
- Parallel processing of decision trees allows for short training and application times.

In contrast to neural networks, however, an RF is less suited to extract features from data automatically (especially for large data types like images) and is unable to extrapolate data that falls outside of the space spanned by the training set.

3.4.5 *Neural Networks*

An artificial neural network (NN) imitates the information processing in the nervous system of animals. In consistence with a biological neural network, the model consists of interconnected neurons that receive, process and relay signals in order to derive some meaningful output. The idea of NNs was first introduced in 1943 by Warren McCulloch and Walter Pitts [91] but did not become part of practical research until the 1980 s due to the simple lack of computational power. Since then, NNs have substantially gained importance in machine learning problems. Nowadays, NNs dominate the research field of machine learning, both in terms of performance and frequency of application. In particular, the success of convolutional NNs in image processing has accelerated the research field tremendously [55, 58, 69, 129, 138, 139]. The pool of different NN types, designed for different purposes, is constantly growing. Here, only the type relevant to this work, a fully connected feed-forward NN, will be presented. In the following, a mathematical description of such a model will be given starting with the working principle of a single neuron.

Figure 14 shows a neuron with n inputs and one output. Let $\boldsymbol{x} \in \mathbb{R}^n$ be the input signal. Then the output y is calculated by

$$y = \sigma\left(\sum_k w_k x_k + b\right) \tag{101}$$

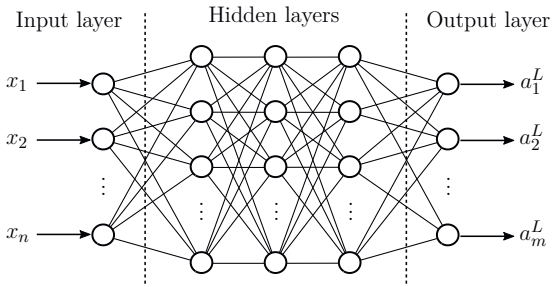


Figure 15: Schematic of a fully connected feed-forward NN with three hidden layers.

with the weight vector $\boldsymbol{w} \in \mathbb{R}^n$, the bias b and the activation function $\sigma()$. The output y is also referred to as *activation* of the neuron. Typical activation functions are described in Section 3.4.5.

When individual neurons are composed to an NN, they are arranged in layers. In traditional feed-forward NNs, connections only exist between neurons of two consecutive layers. Figure 15 illustrates a feed-forward NN with an input layer, a number of hidden layers and an output layer. To describe the activation a_j^l at neuron j of layer l, Eq. (101) is rewritten as

$$a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right) . \tag{102}$$

Now, the sum is executed over all neurons k in layer $l-1$. For the following derivations it is helpful to indicate the weighted sum (the input of σ) by its own variable

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l . \tag{103}$$

While the number of hidden layers and the number of neurons in each hidden layer are flexible design choices, the size of the outer layers is determined by the dimension of the input and output data. NNs can be used for classification as well as regression problems. In principle, only the output layer has to be adapted accordingly.

Error Backpropagation

Training an NN means optimizing the network parameters, e. g. weights and biases, with regard to a *loss function* that is supposed to be minimal. The optimization is performed using a gradient descent method, most commonly *stochastic gradient descent* (SGD) [9], Adam [62] or related methods. For this purpose, the gradients of the network parameters

with respect to the loss are required. The calculation of these gradients is subject of *error backpropagation* or simply *backpropagation*.

Let $C : \mathbb{R}^d \to \mathbb{R}$ be a loss function that rates the output $\boldsymbol{a}^L \in \mathbb{R}^d$ of the last layer L by assigning a real number $C(\boldsymbol{a}^L)$. The goal of backpropagation is the calculation of $\frac{\partial C}{\partial w_{jk}^l}$ and $\frac{\partial C}{\partial b_j^l}$ for all layers, neurons and connections. Application of the chain rule for derivatives yields

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{jk}^l}, \tag{104}$$

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial b_j^l}. \tag{105}$$

The first derivative in Eq. (104) and Eq. (105) is given its own variable

$$\delta_j^l = \frac{\partial C}{\partial z_j^l} \tag{106}$$

and is referred to as *activation error* of neuron j in layer l. If neuron j is not part of the output layer L, the activation error δ_j^l can only be computed indirectly by propagating the errors δ_j^L of the output layer backwards through the network. This is achieved using the following equations:

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L), \tag{107}$$

$$\delta_j^l = \sum_k w_{kj}^{l+1} \delta_k^{l+1} \sigma'(z_j^l). \tag{108}$$

In Eq. (108), the index k indicates neurons in the subsequent layer $l+1$. A complete derivation of these inference rules is given in [48]. Insertion of δ_j^l into Eq. (104) and Eq. (105) and partial differentiation of the weighted sum z_j^l (defined in Eq. (103)) yields the sought-after gradients:

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l, \tag{109}$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l. \tag{110}$$

In order to realize a gradient descent algorithm, an update rule for the network parameters is required. Based on the calculated gradients, the weights and biases are changed by

$$\begin{aligned} \Delta w_{jk}^l &= -\eta a_k^{l-1} \delta_j^l \,, \\ \Delta b_j^l &= -\eta \delta_j^l \,. \end{aligned} \tag{111}$$

The *learning rate* η additionally scales the parameter change. A slightly more sophisticated update rule involves the *momentum* of earlier updates to avoid local minima. For instance, the update at step $t+1$ can be calculated using

$$\begin{aligned} \Delta w_{jk}^l(t+1) &= -(1-\alpha)\eta a_k^{l-1} \delta_j^l + \alpha \Delta w_{jk}^l(t) \,, \\ \Delta b_j^l(t+1) &= -(1-\alpha)\eta \delta_j^l + \alpha \Delta b_j^l(t) \end{aligned} \tag{112}$$

with $0 < \alpha < 1$. Based on the described concepts, a typical learning algorithm for neural networks can be summarized by the following steps:

1. Input of a training sample $\boldsymbol{x}$: The related input layer activations are set to $\boldsymbol{a}^1 = \boldsymbol{x}$.
2. Feedforward: The activations of each layer $l = 2, \ldots, L$ are computed according to Eq. (102).
3. Output error: The activation error of the last layer is calculated using Eq. (107).
4. Backpropagation: The activation errors of each neuron in the layers $l = L-1, \ldots, 2$ are inferred using Eq. (108).
5. Gradient descent: For all layers $l = 2, \ldots, L$ the weights and biases are changed according to a given update rule similar to Eq. (111).

Activation Functions

The output z_j^l of Eq. (103) is a linear combination of the neuron activations of the previous layer. Without an additional non-linearity, introduced by the activation function σ, the neural network would only learn linear mappings. There exist various activation functions that are commonly used in deep learning. Depending on the target task, the network topology and the position in the network, different characteristics are required. In general, however, these functions are monotonously increasing [48]. In the following some common activation functions will be briefly described.

The **logistic function** is a continuously differentiable sigmoidal function with values in the range between zero and one:

$$\sigma_{\log}(z) = \frac{1}{1+e^{-z}} \,. \tag{113}$$

Sigmoidal functions were the most common choice of activation up until 2011. Thereafter, they were mainly replaced by *ReLu* activations that proofed beneficial for the training of increasingly deep networks. The sigmoidal shape can lead to saturation of activations and vanishing gradients. To avoid this behaviour, networks with sigmoidal activation were often pre-trained without supervision in order to find a suitable initialization for supervised training. In the case of the logistic function, the non-zero mean is a further disadvantage, since the lower saturation regime hinders the flow of gradients around zero. This becomes problematic in the early stages of training that predominantly rely on biases and push activations towards zero [45].

In recent research, the logistic function is mainly used in the output layer of binary classifiers, since it produces activations between zero and one that are used to represent probabilities of class affiliation. Furthermore, the sigmoidal shape of the curve pushes activations towards the saturation areas near zero and one and thus supports a clear classification.

The **hyperbolic tangent** $\tanh(z)$ is similar to the logistic function in terms of shape, but is zero-centered. This poses an advantage from an optimization point of view, since it allows modelling of negative inputs and activations can easily change back from zero.

The **ReLU** (rectified linear unit) is a piecewise linear function defined as

$$\sigma_{\text{rec}}(z) = \max(0, z)\,. \tag{114}$$

It is motivated by biological measurements that indicate sparse brain activity and one-sided activation of neurons. Deep NNs with ReLU activations have been shown to converge to an equally good minimum with and without unsupervised pre-training [46]. The resulting model can be understood as a composition of linear models represented by the active subset of neurons. These linear functions allow for an easy calculation of gradients. Furthermore the one-sided activation naturally results in sparse models which are more robust towards small input changes than dense models, because less neurons are effected by them. The hard deactivation of neurons can also be considered a disadvantage. Due to the zero gradient in the negative regime, it is likely that once neurons are deactivated, they remain in this state throughout the rest of the training process. This phenomenon is referred to as the *dying ReLU problem* and makes whole network branches redundant. Therefore, the use of ReLU activations requires larger networks in general. A further problem is that the unconstrained positive values of the linear function can lead to large gradients and ultimately cause numerical problems. This is called *exploding gradients*.

The **leaky ReLU** is a variant of the ReLU designed to avoid the dying ReLU problem. It has a small positive slope in the negative region and thus enables a gradual return of the neuron to an active state. The leaky ReLU is defined by

$$\sigma_{\text{rec}}(z) = \begin{cases} 0.01\,z & z < 0 \\ z & z \geq 0 \end{cases} . \tag{115}$$

The **softmax** is a continuously differentiable activation function that maps outputs z_k to the interval $[0, 1]$ with all activations adding to one:

$$\sigma_{\text{softm}}(\boldsymbol{z})_k = \frac{\exp z_k}{\sum_{i=1}^{n} \exp z_i} , \; k = 1, \ldots, n\,. \tag{116}$$

Consequently, softmax can be used to model probability distributions and is usually applied in the last layer of a multinomial classification[2] network.

Loss Functions

The loss function of an NN has a similar purpose as the objective function in optimization. It measures the error of the network output and is minimized during the training process. Depending on the category of the problem, i. e. regression or classification, different functions are used. The default choice for a regression network arises from the objective of maximum likelihood estimation and is the *mean squared error* (MSE). Assuming Gaussian target distribution, minimizing the MSE is equivalent to maximizing the likelihood, i. e. both approaches yield the same model parameters [48]. The MSE is defined as

$$MSE(\boldsymbol{a}^L, \boldsymbol{t}) = \|\boldsymbol{a}^L - \boldsymbol{t}\|_2^2 \tag{117}$$

with the last layer output $\boldsymbol{a}^L$ and the targets $\boldsymbol{t}$. Note that in the case of a regression network, the activation function of the last layer should be linear in order to receive unrestricted output values. Due to squaring, the error of a variable that is naturally more distributed is amplified in relation to its absolute value. Since it is common to normalize the input and output data, this does not have an irregular effect in general. However, if the model has to predict unscaled values the *mean squared logarithmic error* is an appropriate alternative. Another possible loss function for regression networks is the *mean absolute error* (MAE) which is convenient for data with frequent outliers.

For classification networks, the default loss function is the *cross entropy* (CE) loss [98]. Similar to the MSE, the CE is motivated by a maximum likelihood perspective. In a classification problem with multiple classes but only one label (assigned simultaneously),

2 A multinomial classification includes three or more classes.

the target is a one-hot vector $\boldsymbol{t}$ that has only one component equal to one while the rest are zero. To approximate this behavior, the activations of the last layer are processed with a softmax function (cf. Eq. (116)) resulting in probability values p_i for each class i. Maximizing the likelihood of the observations is equivalent to minimizing the negative log-likelihood

$$L = -\sum_i t_i \log(p_i) = -\log(p_j)\,,\ t_j = 1\,. \tag{118}$$

Interpreting the output vector $\boldsymbol{p}$ and the target vector $\boldsymbol{t}$ as discrete probability distributions over the random variable $\boldsymbol{x}$ leads to the equivalence with the CE

$$H(\boldsymbol{x}) = -\sum_i t_i(\boldsymbol{x}) \log p_i(\boldsymbol{x})\,. \tag{119}$$

In information theory, the CE represents the expected number of bits required to encode a message $\boldsymbol{x}$ with the suboptimal encoding scheme based on the probability distribution $\boldsymbol{p}(x)$ when the actual distribution is $\boldsymbol{t}(x)$.

With respect to the integration of loss functions, it should generally be noted that the implementation of backpropagation requires at least an approximate differentiation of all included functions, such as losses and activations (as well as other common features like pooling and normalization layers). To avoid manual implementation of the derivative of each contained function, current deep learning frameworks include automatic differentiation procedures. These methods build a computational graph that decomposes functions into basic operations and stores all required quantities, such as intermediate results, derivatives and (depending on the mode) operations, to enable the application of the chain rule for differentiation [49].

3.4.6 *Generalization*

A key issue in training machine learning models is their ability to generalize to unknown data, i. e. we are interested in the prediction error of the model on previously unseen data rather than the training data. This is expressed by the *generalization error* [98]

$$e(\boldsymbol{f}) = \int_X \int_Y L(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) p(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \tag{120}$$

that calculates the expected loss L (or another performance measure) of the model function $\boldsymbol{f}$ by integration over all possible inputs $\boldsymbol{x} \in X$ and outputs $\boldsymbol{y} \in Y$ weighted by their joint

probability distribution $p(\boldsymbol{x}, \boldsymbol{y})$. Since the latter is generally unknown, the expected loss is usually estimated by the *empirical error*

$$e_n(\boldsymbol{f}) = \frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{y}_i) \tag{121}$$

on a test set of sample size n. One speaks of a generalizing function if $\lim_{n\to\infty} e_n(\boldsymbol{f}) - e(\boldsymbol{f}) = 0$.

A closely related issue is the *overfitting* of models to noise that is specific to the training set. This occurs if the model complexity is too large compared to the sample size. An arbitrarily flexible model can theoretically fit the training data perfectly, but will perform poorly on unseen test data. The generalization error will be high. Thus, the generalization error can be used to detect overfitting. The established procedure is to split a dataset into training, validation and test set. The training set is used to fit the model, the test set allows empirical estimation of the generalization error and the validation set is necessary to estimate the generalization error on unseen data during the training process in order to optimize hyper parameters of the learning algorithm. The validation set must be different from the test set, since tuning the hyper parameters is also an adjustment of the model. If several of these dataset splits are used, the process is called cross-validation.

For random forests, important hyper parameters are, for example, the number of trees and the maximum depth of trees. With regards to neural networks, it is crucial to stop the optimization, before the model starts to overfit, i. e. before the loss on the validation set increases. Further fundamental hyper parameters are the optimization algorithm, the learning rate and the network architecture. For example, an architecture with a smaller number of neurons in hidden layers than in the input and output layer is beneficial to avoid overfitting. Such an architecture is often referred to as a *bottle-neck*. It enforces the encoding of information into a relatively small feature vector, so that an informative representation is learned. Further successful methods that support generalization are random training data augmentation to increase the sample number, regularization of model complexity (e. g. weight regularization using an L_2-norm) and specific to neural networks, the use of *drop-out*, where a number of neuron activations is randomly ignored in each calculation step [135]. In the context of hyper parameter search, it is worth mentioning that in recent years the research field of *automated machine learning* has been established with the goal to automate hyper parameter tuning [80, 142]. Corresponding techniques, however, are not used in this work.

3.4.7 *Transfer Learning*

In the case where the available dataset is too small (or even unlabeled) and a different but related dataset (usually larger and including labels) exists, transfer learning provides

a way to exploit the knowledge gained from the related set to facilitate the application to the data of interest. Transfer learning, in general terms, means adapting a model to different data *domains* or *tasks*. A domain $\mathcal{D}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$ with $X = (x_1, \ldots, x_n),\ x_i \in \mathcal{X}$:

$$\mathcal{D} = \{\mathcal{X}, P(X)\}\,. \tag{122}$$

Associated with a domain, a *task* is defined by

$$\mathcal{T} = \{\mathcal{Y}, f(x)\} = \{\mathcal{Y}, P(Y|X)\}\,. \tag{123}$$

It is composed of label space $\mathcal{Y}$ and predictive function $f : \mathcal{X} \to \mathcal{Y}$ which can also be represented by a conditional probability distribution $P(Y|X)$ (taking the statistical viewpoint). Formally, the objective of transfer learning is to learn the conditional probability distribution $P(Y_t|X_t)$ (or the predictive function $f_t(X_t)$) belonging to a target task $\mathcal{T}_t$ in the target domain $\mathcal{D}_t$ based on a given source task $\mathcal{T}_s$ in the source domain $\mathcal{D}_s$ [106]. Here, both domains and associated tasks differ in at least one of the following aspects: the feature spaces, the label spaces, the marginal probability distributions or the conditional probability distributions. These possibilities give rise to different scenarios that ask for different transfer learning approaches. In the literature, a common high-level categorisation uses the terms homogeneous transfer (same feature spaces $\mathcal{X}_y = \mathcal{X}_t$) and heterogeneous transfer (different feature spaces $\mathcal{X}_y \neq \mathcal{X}_t$) [155].

The transfer learning problems addressed in this work can be categorized as homogeneous transfer learning. For example, a transfer from walking motions to running motions does not change the feature or label space: Motions and forces are represented by vectors in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, and the dimensions n and m do not change when transferring between motion types. The marginal probability distribution of the features (motion parameters), however, changes significantly, e. g. including higher velocities for running than for walking. The conditional probability distributions, representing the predictive functions that map from motion to forces, are determined by the underlying physics and thus are expected to stay the same or to be closely related in the least.

Basic approaches of homogeneous transfer learning are instance-based, feature-based and parameter-based. In instance-based transfer, the marginal probability distribution of the source domain is adjusted to the target domain by reweighting or resampling of source data points which are close to the target domain. Feature-based transfer uses a transformation that aligns the domains, reducing the gap between the feature spaces[3] or the marginal probability distributions. Finally, in parameter-based transfer, weights are shared between models of the source and target domain. This approach is mainly

3 Feature-based transfer learning techniques can also facilitate heterogeneous transfer.

applied to deep neural networks by pre-training on source domain data and fine-tuning to the target domain [155]. The transfer learning performed in this thesis follows the same approach with the fine-tuning realized by means of self-supervised learning.

4

HUMAN MOTION DATASET

A learning-based approach to inverse dynamics of human motion requires suitable *dynamics datasets*. These datasets include the 3D motion of a kinematic model, parameters to describe the geometry and inertia of the model, the contact forces, their point of application and the net joint moments. The following sections describe the recording of kinematics and contact forces using marker-based motion capture and force plates. In Section 4.3 the approximation of inertial model parameters is presented. The estimation of interior joint torques by means of optimization techniques is subject of Section 4.4. Details of the recorded dataset are summarized in Section 4.5 and the generation of the final data points used for the training of machine learning models is presented in Section 4.6.

4.1 MOTION CAPTURE AND KINEMATIC OPTIMIZATION

In marker-based motion capture, retro-reflective markers are attached to the body of the subject and their motion is recorded and reconstructed in 3D using multiple calibrated infrared cameras. The used system is a *Vicon T-series* motion capture system consisting of 8 infrared cameras. The left of Figure 16 shows a subject whose gait pattern is recorded in the lab. The capture rate was set to 100 Hz. The data recording is performed with the *Vicon Nexus 2* software. To facilitate the post-processing in *Vicon Blade 3*, the standard marker set configuration of this software was used [47]. Post-processing in *Vicon Blade 3* includes manual label correction of wrongly assigned markers, skeleton kinematic fitting and filtering. The output is a skeletal model with 66 DOF.

In this work, a skeletal model with lower DOF is chosen in order to facilitate the use of machine learning algorithms by effectively limiting the input as well as the output parameter space. In general, a smaller parameter space enables training with fewer examples and accelerates the process. A full body model is used for an initial kinematic fit to the Vicon model. It has 38 DOF and 14 rigid bodies. The full body model builds a framework for the definition of a more simple model consisting only of legs and one torso segment which represents the center of mass of the upper body. The simplified model has 24 DOF and 8 rigid bodies. Both models are depicted in Figure 16 on the right

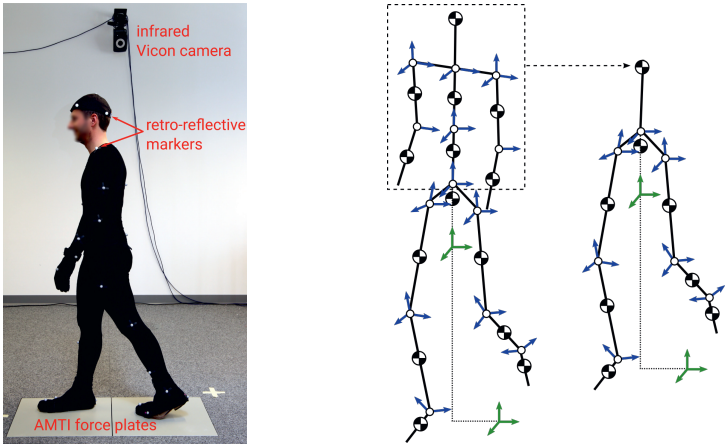


Figure 16: **Left**: Capture setup - the subject wears a suit with attached retro-reflective markers and walks over the force plates. The motion of the markers is recorded by 8 infrared cameras. **Right**: Skeletal models - the blue arrows represent joint DOF and the green arrows global translation and rotation. The simplified model on the right has one torso segment that approximates the kinematics and inertia of the entire upper body on the left.

The kinematics $\boldsymbol{q}(t)$ of the full body model are fitted to the joint trajectories of the Vicon skeleton using constrained optimization. At each frame, the joint positions

$$\boldsymbol{p}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{t}_{j_1} \\ \vdots \\ \boldsymbol{t}_{j_M} \end{bmatrix} \in \mathbb{R}^{3M} \tag{124}$$

are constrained to be equal to the target positions $\boldsymbol{p}_{\text{vicon}}$. Here, $(j_1, \ldots, j_M)$ are the indices of the kinematic link frames that coincide with the positions of the skeletal joints, $\boldsymbol{t}_{j_i}$ are the translations of the corresponding Denavit-Hartenberg transformations (cf. Eq. (17)) and M is the number of joints.

In general, the joint positions are insufficient for a unique identification of all joint angles. If an end-effector position can be produced by the summed rotation around multiple joint axes, an equal distribution of rotation around the contributing axes is assumed. This

is achieved by minimizing squared joint angles. Therefore, for each frame, the following optimization problem is solved:

$$\begin{aligned} &\min \left\{ \sum_{i=7}^{d} q_i^2 \right\} \\ &\text{s.t. } \boldsymbol{p}(\boldsymbol{q}) = \boldsymbol{p}_{\text{vicon}} \,. \end{aligned} \tag{125}$$

Here, d denotes the total number of generalized coordinates specifying the model configuration. The sum starts with index 7 excluding the 6 global DOF. The indication of the frame t is omitted for readability. The optimization is performed using *Sequential Quadratic Programming* [36]. To derive the simplified model from the full body kinematics, the center of mass $\boldsymbol{p}_{ub}$ of the upper body (ub) is calculated by

$$\boldsymbol{p}_{ub} = \frac{\sum_{i \in \text{ub}} m_i \boldsymbol{p}_i}{\sum_{i \in \text{ub}} m_i} \tag{126}$$

and the two remaining torso DOF are optimized in the same manner.

4.2 FORCE PLATE MEASUREMENTS

For measurement of contact forces, two *AMTI force plates* are included in the motion capture system and synchronized to the cameras using an MX Giganet output synchronization signal [47]. The plates are embedded in the ground and record the ground reaction force, the plate moment and the center of pressure on the plate. The capture rate is 10 times higher than the frame rate of the visual system, i. e. 1000 Hz.

The force plates measure three force components (f_x, f_y, f_z) and three moment components (m_x, m_y, m_z) with respect to the plate coordinate system. The measurement is realized with strain gauges that are attached to load cells at the four corners of each plate. The origin of the plate coordinate system is positioned at the center of the plate at a distance z_0 below the surface. The parameter z_0 is supplied in a calibration file. The plate moments can be calculated from the applied force components and the point of application (x, y) by

$$\begin{aligned} m_x &= -f_y z_0 + f_z y + t_x \\ m_y &= f_x z_0 - f_z x + t_y \\ m_z &= -f_x y + f_y x + t_z \end{aligned} \tag{127}$$

with the torsional torque components (t_x, t_y, t_z). Generally only t_z is unequal zero, so that

$$\begin{aligned} m_x &= -f_y z_0 + f_z y \\ m_y &= f_x z_0 - f_z x \\ m_z &= -f_x y + f_y x + t_z \,. \end{aligned} \tag{128}$$

Rearranging of these equations yields the central location of force application on the plate

$$\begin{aligned} y &= (m_x + f_y z_0)/f_z \\ x &= (-m_y + f_x z_0)/f_z \,. \end{aligned} \tag{129}$$

The point $\boldsymbol{r}_{COP} = (x, y, 0)^T$ is also referred to as *center of pressure* (COP) on the force plate. The moment effecting the center of mass of a model's foot is denoted by $\boldsymbol{m}_r$ and is referred to as *ground reaction moment* (GRM). It can be calculated using the cross product of the vector pointing from the foot center of mass $\boldsymbol{r}_{foot}$ to $\boldsymbol{r}_{COP}$ and the *ground reaction force* (GRF) $\boldsymbol{f}_r = [f_x, f_y, f_z]^T$:

$$\boldsymbol{m}_r = (\boldsymbol{r}_{COP} - \boldsymbol{r}_{foot}) \times \boldsymbol{f}_r + [0, 0, t_z]^T \,. \tag{130}$$

The described forces and moments present two equivalent ways of modelling ground interaction: Either we apply the GRF $\boldsymbol{f}_r$ at the center of pressure $\boldsymbol{r}_{COP}$ and additionally apply the torsional torque t_z to the foot segment, or we apply $\boldsymbol{f}_r$ at the center of mass $\boldsymbol{r}_{foot}$ and the total GRM $\boldsymbol{m}_r$ to the foot segment. The latter approach is chosen in this work, because it does not require modelling of additional contact points at the foot segments.

4.3 ESTIMATION OF INERTIAL PROPERTIES

To describe the dynamics of the rigid body system, inertial properties of the individual bodies are needed. Since this work focuses on the learning of the overall connection between motion and forces, rather than the precise description of a body's shape, a simple and computationally efficient approach is chosen. Each segment is modelled as a simple geometrical body with constant density, such as ellipsoids and cylinders. The dimensions of these geometrical bodies are determined using the subject specific segment lengths $\boldsymbol{l}$ and a set of relative scale factors $\boldsymbol{s}$ to approximate the remaining dimensions of the shapes. While $\boldsymbol{l}$ is determined from the joint positions $\boldsymbol{p}$ of the skeletal kinematics, the scale factors are average values taken from surface body scans of the participating subjects. Therefore, $\boldsymbol{s}$ are dataset specific parameters. Further necessary parameters are the relative distances $\boldsymbol{c}$ of the centers of mass from the root of the segment with respect to the total

segment length and relative segment masses $\boldsymbol{m}$ normalized to the total body mass. Both, $\boldsymbol{c}$ and $\boldsymbol{m}$, are set to literature values [157]. The used geometrical shapes are illustrated in Figure 17. Based on the shape and the specific segment parameters a tensor of inertia can be calculated for each rigid body. The components I_x, I_y and I_z correspond to rotations around the x, y and z axis through the centroid of the body.

(a) Triaxial ellipsoid:

$$\begin{aligned} I_x &= \frac{m}{5}\left(r_y^2 + r_z^2\right) \\ I_y &= \frac{m}{5}\left(r_x^2 + r_z^2\right) \\ I_z &= \frac{m}{5}\left(r_x^2 + r_y^2\right) \end{aligned} \tag{131}$$

(b) Semi ellipsoid:

$$\begin{aligned} I_x &= m\left(\frac{r_y^2 + r_z^2}{5} - \left(\frac{3r_z}{8}\right)^2\right) \\ I_y &= m\left(\frac{r_x^2 + r_z^2}{5} - \left(\frac{3r_z}{8}\right)^2\right) \\ I_z &= \frac{m}{5}\left(r_x^2 + r_y^2\right) \end{aligned} \tag{132}$$

(c) Elliptical cylinder:

$$\begin{aligned} I_x &= m\left(\frac{r_y^2}{4} + \frac{r_z^2}{12}\right) \\ I_y &= m\left(\frac{r_x^2}{4} + \frac{r_z^2}{12}\right) \\ I_z &= \frac{m}{4}\left(r_x^2 + r_y^2\right) \end{aligned} \tag{133}$$

The used geometrical models and parameters for every rigid body are listed in Table 1. The table also includes calculation rules for r_x, r_y and r_z, since these depend on the considered segment. The relative center of mass distance c is used to determine the position $\boldsymbol{t}_{n-1}$ of the end-effector frame n located at the center of mass. This position is required to calculate the Jacobian $\boldsymbol{T}$ (cf. Eq. (36)). The multiplication $\boldsymbol{T}^T\boldsymbol{M}$ of the inertia matrix by the Jacobian leads to a shift of the moments of inertia consistent with the *parallel axis theorem* [51]. Therefore, in the EOM specified in Eq. (74), the inertia is related to a rotation around the actual pivot.

To verify this statement a simple example can be considered: A body with mass m and tensor of inertia $\boldsymbol{I}_c$ (related to a rotation around its center of mass) is rotating around

axis $\boldsymbol{z}$ with rotational velocity $\dot{q}$. There is no additional translation, so that the distance r between rotation axis and the body's center of mass is constant. Furthermore, the rotation axis remains fixed at $\boldsymbol{z} = [0, 0, 1]^T$. The position of the center of mass can be parameterized using polar coordinates:

$$\boldsymbol{r} = \begin{bmatrix} r\cos q \\ r\sin q \\ 0 \end{bmatrix} . \tag{134}$$

According to Eq. (36) and (37), the Jacobian transformation between generalized velocity $\dot{q}$ (of the rotation around $\boldsymbol{z}$) and body velocity $\boldsymbol{\xi}$ is

$$\boldsymbol{\xi} = \begin{bmatrix} \dot{\boldsymbol{r}} \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{T}\dot{q} = \begin{bmatrix} \boldsymbol{z} \times \boldsymbol{r} \\ \boldsymbol{z} \end{bmatrix} \dot{q} \tag{135}$$

with the rotational velocity $\boldsymbol{\omega} = \dot{q}\boldsymbol{z}$. Based on this tranformation, a torque effecting the generalized coordinate q can be expressed as

$$\begin{aligned} \tau &= \boldsymbol{T}^T \boldsymbol{M} \dot{\boldsymbol{\xi}} \\ &= \begin{bmatrix} \boldsymbol{z} \times \boldsymbol{r} \\ \boldsymbol{z} \end{bmatrix}^T \begin{bmatrix} m\boldsymbol{I}_{3\times 3} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_c \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{r}} \\ \ddot{q}\boldsymbol{z} \end{bmatrix} \\ &= (\boldsymbol{z} \times \boldsymbol{r})^T m \ddot{\boldsymbol{r}} + \boldsymbol{z}^T \boldsymbol{I}_c \ddot{q} \boldsymbol{z} . \end{aligned} \tag{136}$$

Using the polar coordinate representation of the position vector from Eq. (134), the first term can be transformed as follows:

$$\begin{aligned} (\boldsymbol{z} \times \boldsymbol{r})^T \ddot{\boldsymbol{r}} &= \begin{bmatrix} -r\sin q \\ r\cos q \\ 0 \end{bmatrix}^T \begin{bmatrix} -r\dot{q}^2 \cos q - r\ddot{q}\sin q \\ -r\dot{q}^2 \sin q + r\ddot{q}\cos q \end{bmatrix} \\ &= r^2(\dot{q}^2 \sin q \cos q + \ddot{q}\sin q^2 - \dot{q}^2 \cos q \sin q + \ddot{q}\cos q^2) \\ &= r^2 \ddot{q}(\sin q^2 + \cos q^2) \\ &= r^2 \ddot{q} . \end{aligned} \tag{137}$$

Inserting this result into Eq. (136) yields

$$\tau = (mr^2 + I_{c33})\ddot{q} . \tag{138}$$

Table 1: Geometric parameters and relative mass values specifying inertial properties of each body segment. The relative center of mass positions c and the relative masses m are literature values [157].

name (shape)	l	r_x	r_y	r_z	s_x	s_y	s_z	c	m
Head (a)	l_{he}	$s_x l_{\text{he}}/2$	$s_x l_{\text{he}}/2$	$s_z l_{\text{he}}/2$	0.735	0.875	1.25	1	0.081
Thorax (c)	l_{th} , l_{sh}	$l_{\text{sh}}/2$	$s_y(l_{\text{ab}}+l_{\text{th}})/2$	l_{th}	-	0.673	-	0.82	0.216
Abdomen (c)	l_{ab}	$l_{\text{pe}}/2$	$s_y(l_{\text{ab}}+l_{\text{th}})/2$	l_{ab}	-	0.673	-	0.44	0.139
Upper arm (c)	l_{ua}	$s_x l_{\text{ua}}/2$	$s_y l_{\text{ua}}/2$	l_{ua}	0.32	0.32	-	0.436	0.028
Forearm (c)	l_{fa}	$s_x l_{\text{fa}}/2$	$s_y l_{\text{fa}}/2$	l_{fa}	0.253	0.253	-	0.682	0.022
Upper body (c)	l_{ub}	$l_{\text{pe}}/2$	$s_y l_{\text{ub}}/2$	l_{ub}	-	0.673	-	0.626	0.536
Pelvis (c)	l_{pe}	$l_{\text{pe}}/2$	$s_y l_{\text{pe}}/2$	$s_z l_{\text{pe}}$	-	0.236	0.2	0	0.142
Thigh (c)	l_{t}	$s_x l_{\text{t}}/2$	$s_y l_{\text{t}}/2$	l_{t}	0.376	0.376	-	0.433	0.1
Shank (c)	l_{s}	$s_x l_{\text{s}}/2$	$s_y l_{\text{s}}/2$	l_{s}	0.296	0.296	-	0.433	0.047
Foot (b)	l_{f}	$s_x l_{\text{f}}/2$	$s_y l_{\text{f}}/2$	l_{f}	0.362	0.257	-	0.5	0.015

Thus, to obtain the moment of inertia around the axis $\boldsymbol{z}$, the moment of inertia I_{c33} around a parallel axis through the body's center of gravity is increased by mr^2 which is exactly the *Steiner shift* of the parallel axis theorem.

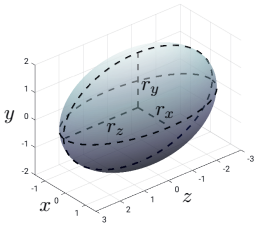
Based on the presented geometric models, the only remaining variables are the segment lengths $\boldsymbol{l}$. Their adjustment for every subject completely determines the dimensions of the kinematic tree and its inertial properties. The full body model is described by $\boldsymbol{l} = [l_{\text{he}}, l_{\text{th}}, l_{\text{sh}}, l_{\text{ab}}, l_{\text{ua}}, l_{\text{fa}}, l_{\text{pe}}, l_{\text{t}}, l_{\text{s}}, l_{\text{f}}]$ and the simplified leg model is specified by $\boldsymbol{l} = [l_{\text{ub}}, l_{\text{pe}}, l_{\text{t}}, l_{\text{s}}, l_{\text{f}}]$. The used indices are defined in Table 1.

4.4 OPTIMIZATION OF JOINT TORQUES

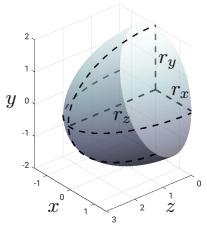
The estimation of joint torques is realized by means of optimization. The optimization procedure is performed using a sliding window across time frames. The sought-after quantities, e. g. joint torques, are parameterized using polynomial approximations with an order adapted to the chosen window length. Accordingly, the optimization variables are polynomial coefficients. After successful optimization of a sequence, i. e. if the optimization of all windows converged to an objective value smaller than a fixed threshold, smoothing filters are applied.

Mathematically, the joint moments can be deduced from the EOM. For better readability of the following description, the equation, that has already been introduced in Section 3.3.1, Eq. (76), is repeated with detailed dependencies:

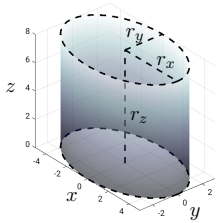
$$\begin{aligned}\boldsymbol{\mathcal{M}}(\boldsymbol{q}(t),\boldsymbol{l})\ddot{\boldsymbol{q}}(t) &= \boldsymbol{\mathcal{F}}(\boldsymbol{q}(t),\dot{\boldsymbol{q}}(t),\boldsymbol{\tau}(t),\boldsymbol{f}_c(t),\boldsymbol{l}) \\ &= \boldsymbol{\tau}(t) + \boldsymbol{T}(\boldsymbol{q}(t),\boldsymbol{l})^T\Big[\boldsymbol{f}_c(t) + \boldsymbol{M}(\boldsymbol{l})(\boldsymbol{g} - \boldsymbol{\zeta}(\boldsymbol{q}(t),\dot{\boldsymbol{q}}(t),\boldsymbol{l})\Big]\,.\end{aligned} \tag{139}$$



(a) Triaxial ellipsoid.



(b) Semi ellipsoid.



(c) Elliptic cylinder.

Figure 17: Geometric shapes used to approximate the inertia of the human body segments.

This equation now includes the segment lengths $\boldsymbol{l}$ introduced in the last section. The contact force vector $\boldsymbol{f}_c$ summarizes contact forces and moments of all rigid bodies. In the case of locomotion analysis, ground contact occurs only for foot segments, so that only those components are unequal zero. For each foot i the force plate measurement results $\boldsymbol{f}_r$ and $\boldsymbol{m}_r$ are inserted into the corresponding components of $\boldsymbol{f}_c$.

The capturing process of 3D motions and the combination with recorded exterior forces introduces multiple error sources. Especially the capturing of the feet motion is often affected by inaccurately placed and/or moving markers. The skeletal model itself, inevitably represents a strong approximation of the real human body. In contrast to that, the force plate measures the effect of the motion in a more direct way. Therefore, there is always a discrepancy between the fitted skeletal motion $\boldsymbol{q}(t)$ of Eq. (125) and its supposed effect, the contact force $\boldsymbol{f}_c$. The applied approach is closely related to *predictive dynamics optimization* [161] which addresses this issue by including the EOM as equality constraints and optimizing all relevant quantities which are the joint torques as well as the contact forces and the motion states. This way, it is possible to reduce model and measurement inaccuracies and simultaneously estimate the underlying joint torques. In practice, it is

beneficial for the convergence of the optimization algorithm to reformulate the equality constraint as a regularization term, so that a slight violation of the EOM is allowed.

In the following, all relevant quantities are represented as polynomials. The coordinates $\boldsymbol{q}$ and their derivatives are approximated by

$$\begin{aligned}
\boldsymbol{q}(t) &= \sum_{i=0}^{o_q} \boldsymbol{\alpha}_q^{(i)} t^i \\
\dot{\boldsymbol{q}}(t) &= \sum_{i=1}^{o_q} i\boldsymbol{\alpha}_q^{(i)} t^{i-1} \\
\ddot{\boldsymbol{q}}(t) &= \sum_{i=2}^{o_q} (i-1) i\boldsymbol{\alpha}_q^{(i)} t^{i-2} \,,
\end{aligned} \tag{140}$$

the contact forces are modelled using

$$\boldsymbol{f}_c(t) = \sum_{i=0}^{o_{f_c}} \boldsymbol{\alpha}_{f_c}^{(i)} t^i \,. \tag{141}$$

and the joint torques are parameterized by

$$\boldsymbol{\tau}(t) = \begin{bmatrix} \mathbf{0}_{6\times 1} \\ \sum_{i=0}^{o_\tau} \boldsymbol{\alpha}_\tau^{(i)} t^i \end{bmatrix} . \tag{142}$$

Here, the first six global components are set to zero. The generalized coordinates and their 1st order derivatives are summarized in the motion state vector $\boldsymbol{x} = \left[\boldsymbol{q}^T, \dot{\boldsymbol{q}}^T\right]^T$. With this notation, the optimization problem is formulated as

$$(\boldsymbol{\alpha}_\tau, \boldsymbol{\alpha}_{f_c}, \boldsymbol{\alpha}_q) = \arg\min \left\{ \boldsymbol{w}_{\mathrm{pd}} \left[E_{\mathrm{EOM}}, E_x, E_p, E_{f_c}, E_{d\tau} \right]^T \right\} \tag{143}$$

using a weighted sum over the individual terms

$$\begin{aligned}
E_{\mathrm{EOM}} &= \frac{1}{T} \sum_{t=1}^{T} \left\| \boldsymbol{\mathcal{M}}(\boldsymbol{q}(t)) \ddot{\boldsymbol{q}}(t) - \boldsymbol{\mathcal{F}}(\boldsymbol{x}(t), \boldsymbol{f}_c(t), \boldsymbol{\tau}(t)) \right\|_2^2 , \\
E_x &= \frac{1}{T} \sum_{t=1}^{T} \left\| \boldsymbol{x}(t) - \boldsymbol{x}^{(m)}(t) \right\|_2^2 , \\
E_p &= \frac{1}{T} \sum_{t=1}^{T} \left\| \begin{bmatrix} \boldsymbol{p}(\boldsymbol{q}(t)) \\ \dot{\boldsymbol{p}}(\boldsymbol{x}(t)) \end{bmatrix} - \begin{bmatrix} \boldsymbol{p}^{(m)}(t) \\ \dot{\boldsymbol{p}}^{(m)}(t) \end{bmatrix} \right\|_2^2 , \\
E_{f_c} &= \frac{1}{T} \sum_{t=1}^{T} \left\| \boldsymbol{f}_c(t) - \boldsymbol{f}_c^{(m)}(t) \right\|_2^2 , \\
E_{d\tau} &= \left\| \boldsymbol{\tau}(1)_w - \boldsymbol{\tau}(T - \delta t)_{w-1} \right\|_2^2 .
\end{aligned} \tag{144}$$

The weights $\boldsymbol{w}_{\text{pd}} = [1, 20, 10, 1, 1]$ are heuristically set. The approach is referred to as predictive dynamics optimization (PDO) in reference to the method by Xiang et al. [161]. Since it is an unconstrained optimization problem, a *BFGS Quasi-Newton method* [14] can be applied. The first term E_{eom} of Eq. (144) measures the deviation from the EOM. The following two terms, E_x and E_p, control the motion. Here, the target motion states $\boldsymbol{x}^{(m)}$ result from the kinematic fit of Eq. (125). In addition to this primarily angle-based view, the global joint positions $\boldsymbol{p}(\boldsymbol{q})$ and velocities $\dot{\boldsymbol{p}}(\boldsymbol{x}) = \boldsymbol{T}_j(\boldsymbol{q})\dot{\boldsymbol{q}}$ are compared to the target values. The velocity is calculated using the Jacobian $\boldsymbol{T}_j$ which is related to the kinematic links localized at the skeletal joints. An objective function that considers the global coordinates like E_p weights the positions of all joints equally, while an angle-related objective function like E_q causes a higher deviation of joint positions at the end points of kinematic chains. Therefore, the combination of both functions yields a more accurate measure for the proximity of two motions. Additionally, the contact forces are regularized by E_{f_c} and large changes of the joint torques from the previous window $w-1$ to the current window w are penalized by $E_{d\tau}$. Here, δt is the overlap between windows.

In order to avoid unrealistic interpolation of the abrupt contact dynamics, the window size is set to 3 frames which is the minimal value that still allows the calculation of acceleration. The overlap is set to 2 frames. With this choice, constant forces, torques and accelerations are sufficient, so that $o_{f_c} = o_\tau = 0$ and $o_q = 2$. After optimization, the results are concatenated and the values during overlapping time frames are averaged. This way, changes in value are possible at any frame despite constant forces during the individual windows. Some example results are shown in Figure 18, including sagittal knee angles, vertical GRF, medio-lateral GRM and sagittal ankle torques (from left to right). Each row corresponds to one sequence.

4.5 DATA SPECIFICATION

The recorded dataset encompasses 185 walking and 66 running sequences executed by 22 healthy subjects. The associated demographic information is provided in Table 2. All subjects volunteered to participate in the study and signed an informed consent form. The study is part of the “Individualized Implant Placement” project funded by the European Research Council (ERC-2013-PoC) and was approved by the ethics commission of the Hannover Medical School (MHH). Natural movements were achieved by instructing the subjects to walk and run at different speeds without paying attention to the force plates. Invalid trials with incorrect foot placement on the plates were sorted out afterwards. In order to increase and balance the dataset, augmentation is performed by mirroring the kinematics and dynamics at the sagittal plane doubling the number of sequences.

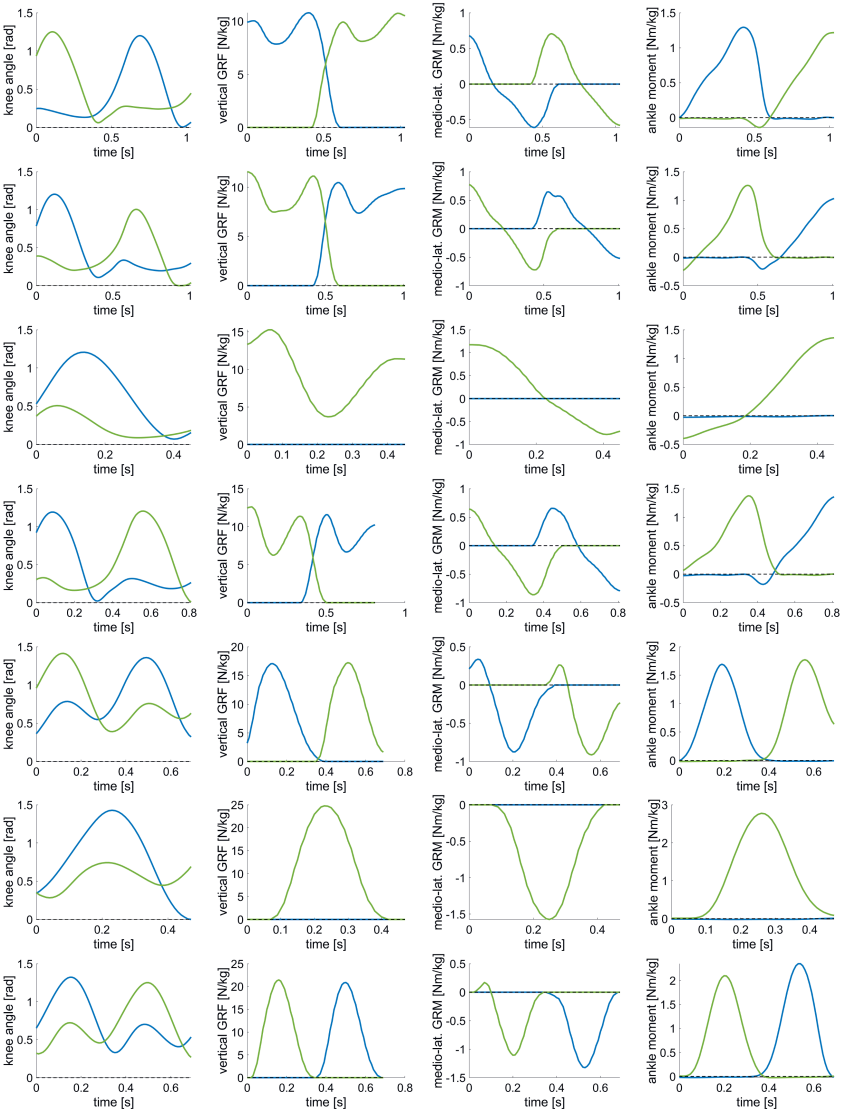


Figure 18: Predictive dynamics dataset examples. The figure includes sagittal knee angles, vertical GRF, medio-lateral GRM and sagittal ankle torques (from left to right). The plots of a row belong to one sequence of the set.

Table 2: Demographic table of participating subjects.

subject ID	gender	height	weight	BMI
1	f	1.69	65.5	23
2	m	1.88	74.4	21
3	m	1.71	61.8	21
4	m	1.81	79.3	24
5	f	1.71	66.6	23
6	m	1.85	74.5	22
7	m	1.86	96.6	28
8	m	1.67	83.8	30
9	m	1.85	95.8	28
10	m	1.84	68.8	20
11	m	1.81	67.9	21
12	m	1.75	81.4	27
13	m	1.72	79.4	27
14	f	1.70	68.0	24
15	m	1.94	88.8	24
16	m	1.80	72.4	22
17	m	1.78	93.5	30
18	m	1.86	68.3	20
19	m	1.80	83.5	26
20	m	1.80	68.3	21
21	m	1.79	69.9	22
22	f	1.73	55.7	19

4.6 GENERATION OF TRAINING DATA POINTS

The training of the learning-based algorithms requires datasets consisting of 3D motions, GRF/M and joint torques. For this purpose, the PDO results of the laboratory data (cf. Section 4.4) and a public dataset by Fukuchi et al. [39] are used. The latter consists of 308 pre-processed sequences of level ground and treadmill walking executed by 44 subjects. The data provided includes joint angles, joint moments and GRF. For the sake of readability, the predicitive dynamics set will be referred to as *PD-set* and the public dataset as *Fukuchi-set* in the following.

The PD-set yields sequential data of generalized coordinates $\boldsymbol{q}(t)$ and subject specific length parameters $\boldsymbol{l}$ which fully describe the kinematics. Furthermore, it includes the GRF/M $\boldsymbol{f}_c(t)$ and the joint torques $\boldsymbol{\tau}(t)$, whereas the Fukuchi-set only provides pre-processed joint angles, GRF and joint torques. In both cases, the goal is a regression

of the underlying forces and moments, also referred to as *control*, based on the motion parameters. The following description uses the notation for the PD-set. The adaptation for the application to the Fukuchi-set is straight forward by replacing the complete motion states $\boldsymbol{x}(t)$ with the provided joint angles and the GRF/M with the GRF.

Similar to the optimization method presented above, the learning-based approaches operate on sliding windows with polynomial approximations. Since the modelled dynamics are independent of the translation of the root joint and the global orientation around the vertical axis, all windows are aligned in terms of the these coordinates. The initial translation is shifted to zero by $\boldsymbol{q}_{1\ldots3}(t) = \boldsymbol{q}_{1\ldots3}(t) - \boldsymbol{q}_{1\ldots3}(0)$. Furthermore, the orientation around the vertical axis is roughly aligned by determining the average value of each motion sequence prior to the split into windows and by rotating all global quantities correspondingly, in particular, the contact forces and moments. After alignment, the input motion states $\boldsymbol{x}(t)$ are parameterized using the coefficients $\boldsymbol{\alpha}_x$ by

$$\boldsymbol{x}(t) = \sum_{i=0}^{o_x} \boldsymbol{\alpha}_x^{(i)} t^i \tag{145}$$

with polynomial order o_x and $t = t_0, \ldots t_{w-1}$ corresponding to w frames. In the case of 3D pose reconstruction results as input data, global translation and orientation information might not be available. This leaves the joint angles and angular velocities which are indicated by $\hat{\boldsymbol{x}} = [q_7, \ldots, q_d, \dot{q}_7, \ldots, \dot{q}_d]^T$. They are approximated by

$$\hat{\boldsymbol{x}}(t) = \sum_{i=0}^{o_x} \boldsymbol{\alpha}_{\hat{x}}^{(i)} t^i \,. \tag{146}$$

Note that the approximations of $\boldsymbol{x}(t)$ and $\hat{\boldsymbol{x}}(t)$ in Eq. (145) and Eq. (146), respectively, violate the differential relation between $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$. However, this is not important for the success of the regression. Similar to the motion states, the control vector

$$\boldsymbol{u}(t) = \begin{bmatrix} \boldsymbol{f}_c(t) \\ \boldsymbol{\tau}(t) \end{bmatrix} \tag{147}$$

is approximated by

$$\boldsymbol{u}(t) = \sum_{i=0}^{o_u} \boldsymbol{\alpha}_u^{(i)} t^i \tag{148}$$

using the coefficients $\boldsymbol{\alpha}_u$.

Based on these representations, the tasks examined in Chapter 5 and Chapter 6 are to predict $\boldsymbol{\alpha}_u$ from $\boldsymbol{\alpha}_x$ or from $\boldsymbol{\alpha}_{\hat{x}}$. All data points are shifted to the respective mean value of the training set and normalized to its standard deviation. The tuning of relevant hyper

data	split	training IDs	validation IDs	test IDs
walk	1	5, 8, 9, 10, 11, 14, 16, 17, 18, 19, 20, 21, 22	1, 2, 3, 13	4, 6, 12, 15
	2	1, 2, 4, 6, 8, 9, 12, 14, 16, 17, 18, 21, 22	3, 10, 11, 13	5, 15, 19, 20
	3	1, 2, 3, 4, 6, 9, 11, 12, 13, 14, 15, 16, 19	5, 18, 21, 22	8, 10, 17, 20
run	1	3, 4, 8, 9, 10, 11, 12, 13, 14	2, 6, 7	5, 15, 16
	2	2, 5, 6, 8, 9, 10, 13, 14, 15	3, 7, 16	4, 11, 12
	3	4, 7, 8, 9, 10, 11, 12, 13, 16	2, 3, 6	5, 14, 15

Table 3: Split of the PD-set into training, validation and test sets according to subject IDs. The validation sets are used to adjust hyper parameters and the test sets are used to evaluate the performance of the regression methods.

parameters and the quantitative evaluation of the proposed methods is performed using the PD-set. For this purpose three different splits into training, validation and test set are examined. The dataset is randomly split according to the subject IDs as listed in Table 3. Different splits are used for walking and running data, due to the smaller size of the *running set*.

5 SUPERVISED LEARNING OF INVERSE DYNAMICS

In this chapter, supervised learning of human dynamics is presented. The application of machine learning to inverse dynamics is motivated by the tremendous success of artificial neural networks in related problems such as 2D and 3D human pose estimation [19, 50, 90, 137, 162]. The regression of acting forces and moments from an observed motion represents the next natural step toward a complete learning-based analysis of human motion. As previously described, the goal is to estimate external forces and moments as well as joint torques from a kinematic representation of an input motion, as illustrated in Figure 19. In contrast to traditional model-based approaches, e. g. using inverse or forward dynamics formulations and solving by optimization, machine learning techniques include the following advantages[1]:

1. Robustness against noisy kinematic input, i. e. smoothing of outliers.
2. Applicability to incomplete or dimensionality reduced input representations.
3. Automatic extraction/selection of relevant features.
4. Low computation time of the application.

Motivated by these properties, several supervised regression methods for estimating dynamics in human locomotion are presented here. Parts of this chapter are based on previous publications [170, 171]. The proposed methods predict forces and moments during the full gait cycle using 3D motion data of multiple subjects. They can be divided into end-to-end regressions and multi-stage methods that are motivated by the limited amount of training data and the frequent criticism of neural networks for their lack of interpretability. The latter method performs a classification on a set of handcrafted features to identify the current motion phase and then infers the GRF/M and joint torques in the sub space corresponding to the class label. The gait phases are viewed as sub categories of the primary class, the motion type (e.g. walking). For both, the end-to-end regression and the multi-stage method, different algorithms are tested including artificial neural networks, random forests and ridge regression. The quantitative comparison is conducted based on the datasets presented in Chapter 4.

The proposed methods are designed to perform without information about the global position and orientation of the human body. This property makes them applicable to

1 The mentioned properties are model-dependent and do not apply to all machine learning methods without restrictions.

results of 3D pose reconstruction algorithms. The application to reconstructed 3D motions is demonstrated using results generated by a structure from motion approach [148]. The motions were reconstructed from 2D gait patterns of the CMU database [18].

An obvious disadvantage of learning the dynamics from a training set is the dependency on the available data. The application of the resulting models is limited to data that lies within the parameter space of the learned dynamics, e. g. in terms of body and motion type. But exactly this property can be exploited for the detection of abnormal motion patterns, i. e. motions that are atypical with respect to the used training dataset. This is shown by means of an asymmetrical gait pattern which was reconstructed from inertial sensor data [88].

The contributions of this chapter can be summarized as follows:

1. Learning-based methods for solving the inverse dynamics problem of human motion.
2. A multi-stage approach including regression of missing motion input, gait phase classification, contact feature extraction and control regression is presented.
3. The multi-stage methods and the end-to-end regressions are quantitatively evaluated for walking and running sequences and compared to data-driven inverse dynamics optimization.
4. The applicability of the proposed methods to reconstructed motions from the CMU database is investigated.
5. Abnormal gait detection is exemplified using a sequence reconstructed in 3D from IMU data.

In the following Section 5.1 the proposed methods are described. In Section 5.2 the experimental evaluation is presented and Section 5.3 discusses the main results and concludes this chapter.

5.1 METHODOLOGY

To realize supervised learning of the inverse dynamics problem at hand, the models must match the available data scope and data properties. The data is characterized by its sequential nature, high correlations between motions of the same subject, varying parameter scales, and a nonlinear mapping (given by the EOM) to be learned. To facilitate the regression problem, especially in view of the comparably small training sets, a polynomial approximation of the input and output parameters is used to capture temporal context in advance. As described in Chapter 4, control parameters $\boldsymbol{\alpha}_u$ will be regressed from motion parameters $\boldsymbol{\alpha}_x$ (or $\boldsymbol{\alpha}_{\hat{x}}$ that exclude global information). These parameter vectors

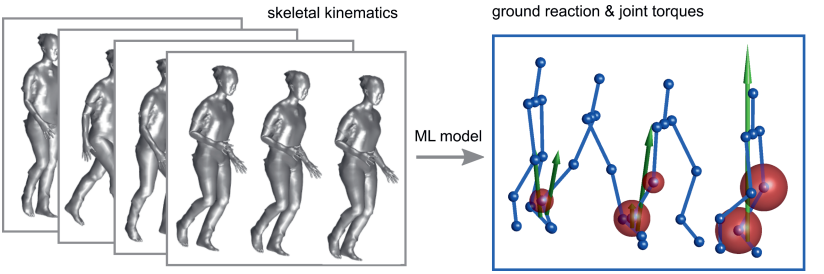


Figure 19: Illustration of the regression task. The GRF/M and the joint torques effecting the modelled human body are predicted by a machine learning (ML) model based on the skeletal kinematics during short temporal windows. The human shapes were generated using SMPL [84].

encode the controls $\boldsymbol{u}(t)$, consisting of joint torques and GRF/M, and the motion states $\boldsymbol{x}(t)$ which contain the generalized coordinates and their first derivatives. Each parameter vector describes a short temporal window of the associated curve. Therefore, the trained regression models are applied to a motion sequence by dividing it into corresponding windows with overlapping frames. The overlap reduces the change of successive input and output parameter pairs and allows finer discretization than given by the used window length.

Varying distributions of individual components are compensated by subtracting the mean and normalizing to the standard deviation. This is done in addition to the normalization of forces and moments using the subject's mass as described in Section 4.3 which is necessary to allow cross-subject predictions. Further measures that help not to overfit on person-specific details include the use of a simplified skeletal model, the split into training, validation and test set according to subject identification and the consideration of small temporal windows.

A main feature of the data is introduced by the ground contact or more precisely by the lack thereof. While motion states and joint torques are progressing smoothly over the course of a gait period, the GRF/M are characterized by sudden changes and are frequently identical with zero. This type of behavior is difficult to learn within a regression task, because it requires the output of absolute zeros. This fact motivates the use of a multi-stage approach including a classification of the contact state which narrows down the regression to the relevant subset and greatly facilitates the output of vanishing forces. In this sense, the method uses prior knowledge about the data which is a reasonable approach given the typically small dataset sizes. In the following sections, the regression methods are described in detail.

5.1.1 *End-to-End Regression*

As a baseline, a direct regression of control coefficients $\boldsymbol{\alpha}_u$ from motion coefficients $\boldsymbol{\alpha}_{\hat{x}}$ is investigated. The implemented methods are a neural network (NN), a random forest (RF) and a linear ridge regression. Their operating principles have been presented in Chapter 3.

The RF is generated by bagging of decision trees. The number of trees and the minimal number of examples in the leaves are determined for each training set separately by comparing the performance on the corresponding validation set (specified in Table 3) using grid search. The same validation procedure is applied to the ridge regression, in order to identify the weighting parameter of the regularization term.

The NN is a fully connected feed-forward network. Here, the determined hyper parameters include the architecture, the activation function, the batch size and the number of epochs. The considered networks have 1 to 3 hidden layers with sizes between 50 and 200 neurons. The training of the NN is done using the Adam optimization algorithm [62]. The three approaches are termed as *end-to-end regressions*.

5.1.2 *Multi-Stage Regression*

The multi-stage approach consists of inference of the missing global information, feature extraction, classification and finally regression of control coefficients. The individual steps are now described separately:

1. **Root Regression**. If the root coordinates, i. e. the global orientation and the linear and angular velocities of the root joint[2], are not available as input, they are estimated by regression methods. The missing coordinates of the root joint $\boldsymbol{\alpha}_{\text{root}}$ are inferred from the partial parameterization $\boldsymbol{\alpha}_{\hat{x}}$ by a mapping

$$\boldsymbol{\alpha}_{\text{root}} = \boldsymbol{f}_{\text{root}}(\boldsymbol{\alpha}_{\hat{x}})\,. \tag{149}$$

   The global information is needed for the calculation of additional contact features, as will be described subsequently. For the root regression the same methods as for the end-to-end regression are applied and compared. The root coefficients $\boldsymbol{\alpha}_{\text{root}}$ are inserted into $\boldsymbol{\alpha}_x$ and the full states $\boldsymbol{x}(t)$ can be obtained using Eq. (145).

2. **Contact Feature Extraction**: In the second step, additional features that are supposed to predominantly characterize the contact state of the model are calculated. These features are the absolute velocities $\boldsymbol{v}_c$ of the joints at the model's feet, more

2 The global position at the initial frame is not included, since it has been set to zero to align the data points.

precisely at ankle and toe joints. The three dimensional linear velocity $\boldsymbol{v}_i$ of each foot joint position $i = 1, 2, 3, 4$ (left ankle, left toe, right ankle and right toe) is computed using the corresponding submatrix $\boldsymbol{T}_{v_i}$ (including only the rows that transform to linear velocities) of the Jacobian. Then the L_2-norms are determined and averaged over the time span of the window to obtain $\boldsymbol{v}_c$:

$$\begin{aligned} \boldsymbol{v}_i(t) &= \boldsymbol{T}_{v_i}(\boldsymbol{q}(t))\dot{\boldsymbol{q}}(t)\,, \\ \boldsymbol{v}_c &= \frac{1}{T}\sum_{t=1}\begin{bmatrix} \|\boldsymbol{v}_1(t)\|_2 & \|\boldsymbol{v}_2(t)\|_2 & \|\boldsymbol{v}_3(t)\|_2 & \|\boldsymbol{v}_4(t)\|_2 \end{bmatrix}. \end{aligned} \tag{150}$$

The modified feature vector is

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\alpha}_x^T & \boldsymbol{v}_c \end{bmatrix}. \tag{151}$$

3. **Gait Phase Classification**: In the third part, a class label c is assigned to the considered feature vector. Each motion type is divided into several phases and a classification task is performed based on $\boldsymbol{\theta}$:

$$l_c = f_{\text{phase}}(\boldsymbol{\theta})\,. \tag{152}$$

A walking cycle is divided into *double support left*, *single support left*, *double support right* and *single support right* and a running cycle is divided into *support left*, *flight left*, *support right* and *flight right*[3]. The methods implementing f_{phase} are a support vector classifier with a radial basis function kernel, an RF and an NN. In the case of the support vector machine (SVM) and the RF, class weighting is applied according to the reciprocal sample numbers to balance the classifiers. In RF, the weights adjust the impurity score used to find training set splits in favour of the minority classes. The SVM is modified by multiplying the softness parameter of class specific margins with the corresponding weights. Both approaches reduce the penalty for false positives of minority classes. The NN on the other hand, receives a training set in which the underrepresented classes are oversampled.

4. **Control Regression**: The final part of the proposed approach is the regression of control coefficients $\boldsymbol{\alpha}_u$ from $\boldsymbol{\alpha}_{\hat{x}}$ given a class label l_c. One model f_{u,l_c} for each class l_c is trained using the respective subset of motion coefficients as predictors and the subset of control coefficients as responses resulting in

$$\boldsymbol{\alpha}_u = \boldsymbol{f}_{u,l_c}(\boldsymbol{\alpha}_{\hat{x}})\,. \tag{153}$$

3 In case of double support and flight, the designations *left* and *right* indicate the leading foot.

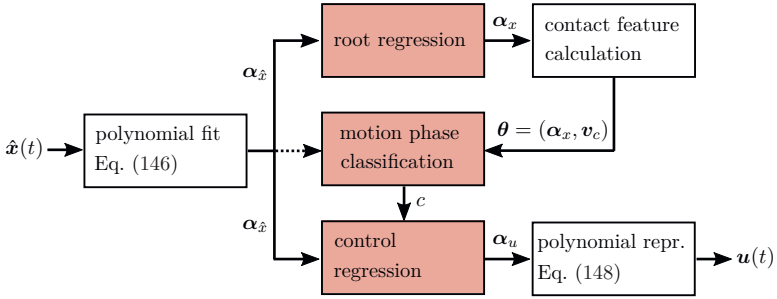


Figure 20: Schematic illustration of the multi-stage regression method. The process steps in the colored boxes are implemented by different learning methods. The dashed arrow represents a version of the method investigated within the context of an ablation study (cf. Table 8).

Only if the global coordinates are given from the beginning, they are used as input here. The control regression is not performed on estimated $\boldsymbol{\alpha}_x$, but on the original $\boldsymbol{\alpha}_{\hat{x}}$, to avoid the multiplication of uncertainties. The task is solved with the same three regression methods as before.

The actual temporal progressions of joint torques and GRF/M can be computed from $\boldsymbol{\alpha}_u$ using the polynomial approximation in Eq. (148). The method is referred to as *multi-stage* regression and the related process is shown in Figure 20.

5.2 EXPERIMENTAL EVALUATION

In this section the proposed learning-based inverse dynamics methods are evaluated. For quantitative evaluation, the recorded laboratory data is used in the form of the PD-set (cf. Section 4.6). The performance of the regression algorithms is evaluated using the following error measures. Predicted quantities $g(t)$ at discrete times t are compared to the target values $h(t)$ in terms of relative root mean squared error (rRMSE) ϵ:

$$\epsilon = \frac{RMSE}{\frac{1}{N}\sum_{i=1}^{N}(\max h_i - \min h_i)}\,, \tag{154}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(g(t) - h(t))^2}\,. \tag{155}$$

The rRMSE is normalized to the average range of the target value in the training set with N samples. In contrast to the root mean squared error (RMSE), the relative measure normalizes the deviation of components according to their average range in the training set.

Thus, components with generally small absolute values are relevant as well. The rRMSE is applied to predicted GRF, GRM, joint torques and the concatenated vector of controls $\boldsymbol{u}$. Furthermore, RMSE values and Pearson's correlation coefficients

$$\rho = \frac{\sum_t (g(t) - \bar{g})(h(t) - \bar{h})}{\sqrt{\sum_t (g(t) - \bar{g})^2} \sqrt{\sum_t (h(t) - \bar{h})^2}}\,, \;\; \bar{g} = \frac{1}{T} \sum_{t=1}^{T} g(t)\,, \tag{156}$$

for the main comparison of the different methods are given in Appendix A.1.

Several algorithms for the end-to-end regression and for individual steps of the multi-stage method, i.e. for the root regression, the gait phase classification and the control regression are compared. The influence of the gait phase classification and the use of additional contact features is investigated in an ablation study. In addition, the end-to-end regression methods are contrasted using the Fukuchi-set. This set is only used for the end-to-end approach, because the contact features $\boldsymbol{v}_c$ cannot be calculated based on the provided data.

Furthermore, the performance on reconstructed gait motions taken from the CMU database is investigated qualitatively. On this dataset, a quantitative evaluation is not possible due to the lack of GRF/M and a consequential lack of joint torques. In a final experiment the application of the learning-based approach is tested as a tool for the detection of abnormal gait based on inertial measurements.

5.2.1 *Predictive Dynamics Dataset*

The proposed methods are evaluated on the laboratory data using the test sets, listed in Table 3. For this first experiment, the entire kinematics are used as input, i. e. also the global root orientation. The difference between predicted and target curves (predictive dynamics optimization results) is quantified using the rRMSE presented in Eq (154). Table 4 lists rRMSEs ϵ_{f_r}, ϵ_{m_r}, ϵ_τ and ϵ_u of the GRF, GRM, joint torques and all controls $\boldsymbol{u}$. The collective measure ϵ_u is included to assess the overall performance of a method. The presented learning-based algorithms are compared to a data-driven inverse dynamics approach [85] as well. This method incorporates physical modeling into a maximum a posteriori framework. Implementation details can be found in Appendix A.2. In the case of the end-to-end and multi-stage regressions, only the best performing implementations are listed in Table 4. In addition to the rRMSE, the table also includes mean per frame computation times.

The proposed regression methods achieve lower error values than the data-driven optimization, while reducing computation times by two orders of magnitude. The computation times include the initial optimization of motion coefficients $\boldsymbol{\alpha}_x$. The values are obtained

Table 4: Comparison between the presented regression methods and a data-driven optimization [85]. The table shows rRMSE values of predicted GRF/M, joint torques and the entire controls $\boldsymbol{u}$. The last column lists computation times per frame. Standard deviations are indicated in parentheses.

data	method	ϵ_{f_r} [%] ↓	ϵ_{m_r} [%] ↓	ϵ_τ [%] ↓	ϵ_u [%] ↓	comp. time [s]
walk	[85]	12.8 (5.5)	18.6 (6.3)	16.4 (5.0)	16.1 (4.6)	3.246 (0.265)
	end-to-end	8.7 (3.0)	**16.1** (6.1)	13.3 (3.6)	12.9 (3.3)	**0.049** (0.003)
	multi-stage	**7.4** (4.2)	17.7 (5.6)	**12.1** (4.2)	**12.3** (4.0)	0.061 (0.011)
run	[85]	17.2 (4.2)	18.4 (5.7)	19.7 (3.8)	19.0 (3.7)	3.368 (0.300)
	end-to-end	**12.6** (4.0)	**14.4** (4.8)	14.8 (3.2)	**14.3** (3.0)	**0.048** (0.002)
	multi-stage	13.1 (4.4)	14.7 (5.3)	**14.6** (2.9)	**14.3** (3.0)	0.084 (0.013)

using unoptimized python code without parallelization, run on an Intel(R) processor with 3.50 GHz. On the walking data, the multi-stage approach outperforms the end-to-end regression with respect to the overall prediction capability measured by ϵ_u. However, a clear drop in performance of the regression methods can be observed for the GRM. This is due to a high variability of the vertical and the anterior-posterior components which fluctuate around zero. These components are small compared to the medio-lateral moment and are strongly influenced by the quality of the inverse kinematics result, in particular the estimated position of foot joints relative to the COP of the GRF. Unlike the running set, the walking set is composed of data from two recording sessions in which different inverse kinematics methods were used. This results in two modes that are particularly pronounced in the GRM and affect their regression because the model must either learn to distinguish between the two modes based on subtle differences in the motion representation or perform some kind of averaging between them. The first scenario is not desirable, as it represents a form of overfitting. However, it is unlikely due to the low complexity of the models. On the running data, the overall performance of end-to-end and multi-stage regression is similar.

To illustrate the mean performance of the different methods, Figure 21 shows multi-stage regression results for walking and running. The figure showcases the major joint torques active in human locomotion (sagittal ankle, knee and hip torques) as well as vertical and anterior-posterior GRF and medio-lateral GRM. These are the components with the largest absolute values of the entire control parameters. It can be seen that the multi-stage model can reliably predict control values during frames without ground contact due to the additional gait phase classification. Better agreement between the distribution of targets and estimates is obtained when ridge regression is used to infer controls in a given class (cf. walking) than when RF is used instead (cf. running). The RF appears to predominantly approximate the average values of the set instead of the features of the individual data points.

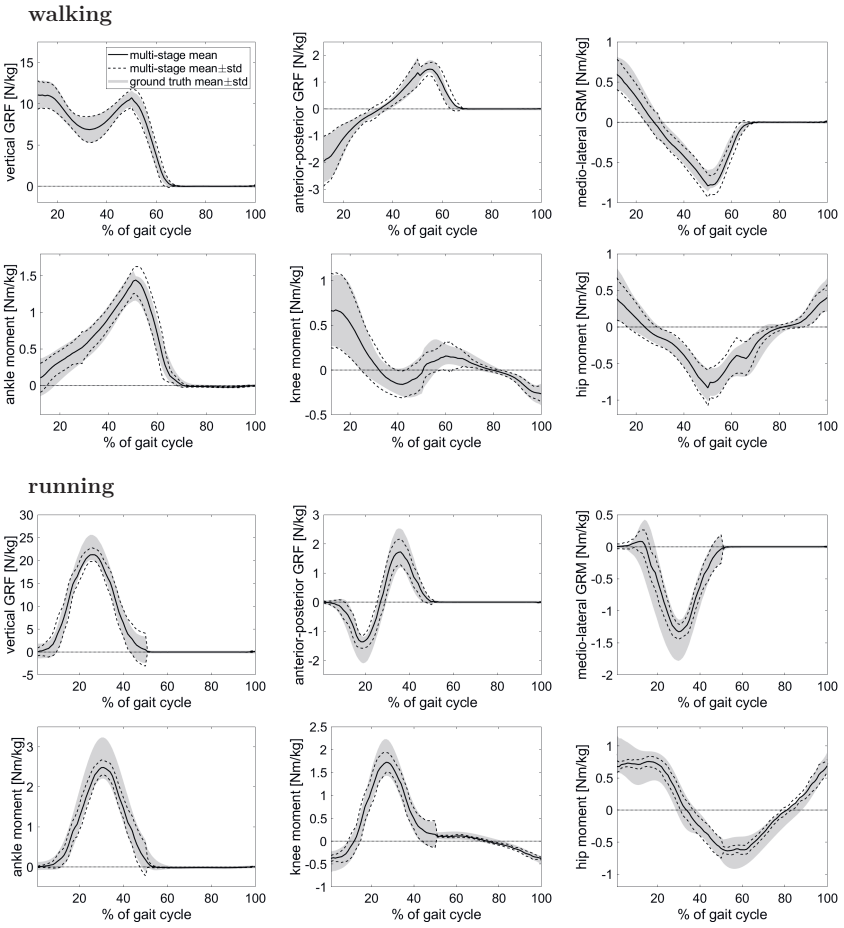


Figure 21: Averaged predicted GRF/M and joint torque components. The figure shows multi-stage results generated without global coordinates as input. The top rows display results for walking with RF root regression, NN gait phase classification and ridge control regression. The estimates for running (bottom) were produced with ridge root regression, RF classification and RF control regression. The bold line represents mean predictions and the dashed lines the related standard deviation. The grey area illustrates mean ± standard deviation of the ground truth.

Table 5: **End-to-end** control regression results. The table shows rRMSE values of predicted GRF/M and joint torques for a regression with and without global root coordinates as input.

data	input	method	ϵ_{f_r} [%] ↓	ϵ_{m_r} [%] ↓	ϵ_τ [%] ↓	ϵ_u [%] ↓
walk	$\boldsymbol{\alpha}_x$	Ridge	10.5 (2.9)	18.8 (5.7)	14.7 (4.6)	14.7 (4.0)
		RF	**8.7** (3.0)	**16.1** (6.1)	13.3 (3.6)	**12.9** (3.3)
		NN	9.5 (3.1)	18.2 (5.0)	**12.8** (3.3)	13.2 (3.1)
	$\boldsymbol{\alpha}_{\hat{x}}$	Ridge	10.6 (3.0)	18.7 (6.2)	15.0 (4.7)	14.8 (4.2)
		RF	**8.7** (2.9)	**16.4** (6.1)	13.3 (3.6)	**13.0** (3.3)
		NN	9.6 (2.9)	17.7 (5.3)	**13.0** (3.4)	13.2 (3.2)
run	$\boldsymbol{\alpha}_x$	Ridge	18.8 (5.3)	24.8 (7.0)	20.8 (5.7)	21.2 (5.1)
		RF	**12.6** (4.0)	**14.4** (4.8)	**14.8** (3.2)	**14.3** (3.0)
		NN	14.4 (3.8)	20.1 (6.0)	15.3 (3.1)	16.1 (3.3)
	$\boldsymbol{\alpha}_{\hat{x}}$	Ridge	18.7 (5.1)	25.2 (7.1)	21.6 (5.6)	21.7 (4.9)
		RF	**12.8** (4.1)	**14.3** (4.8)	**14.8** (3.2)	**14.3** (3.0)
		NN	15.3 (4.4)	19.0 (5.8)	17.6 (4.4)	17.4 (3.7)

Note that for walking, the curves do not span over a full gait period, since the experiments have been done using the optimized input motion states which were generated by PDO. For this algorithm complete force plate information is required. However, the given lab setup only included two force plates, resulting in valid information for one single support, one double support and a second single support. Thus, a second double support is missing to form a full gait period.

Figure 22 shows example regression results for walking and running, generated using the best multi-stage methods for each data type. The chosen sequences have error values close to the mean of the respective data type and regression method.

Comparison of Different Implementations

In the following, different implementations of the proposed methods are evaluated. Table 5 shows end-to-end regression results based on complete kinematic input $\boldsymbol{\alpha}_x$ and partial input $\boldsymbol{\alpha}_{\hat{x}}$. The multi-stage regression results for walking and running are summarized in Table 6. Here, only the best performing combinations of methods (in terms of ϵ_u) are shown together with average performances. For walking, the best performing end-to-end regression method is an RF with $\epsilon_u = 12.9\,\%$. The best multi-stage approach achieves $\epsilon_u = 12.3\,\%$ using an RF as classifier and a ridge regression to estimate the controls. These are the results including global root coordinates as input. If the global information is left out, the corresponding values are $\epsilon_u = 13.0\,\%$ for end-to-end and $\epsilon_u = 12.2\,\%$ for multi-stage. In the latter case, this performance is achieved by an RF root regression, an

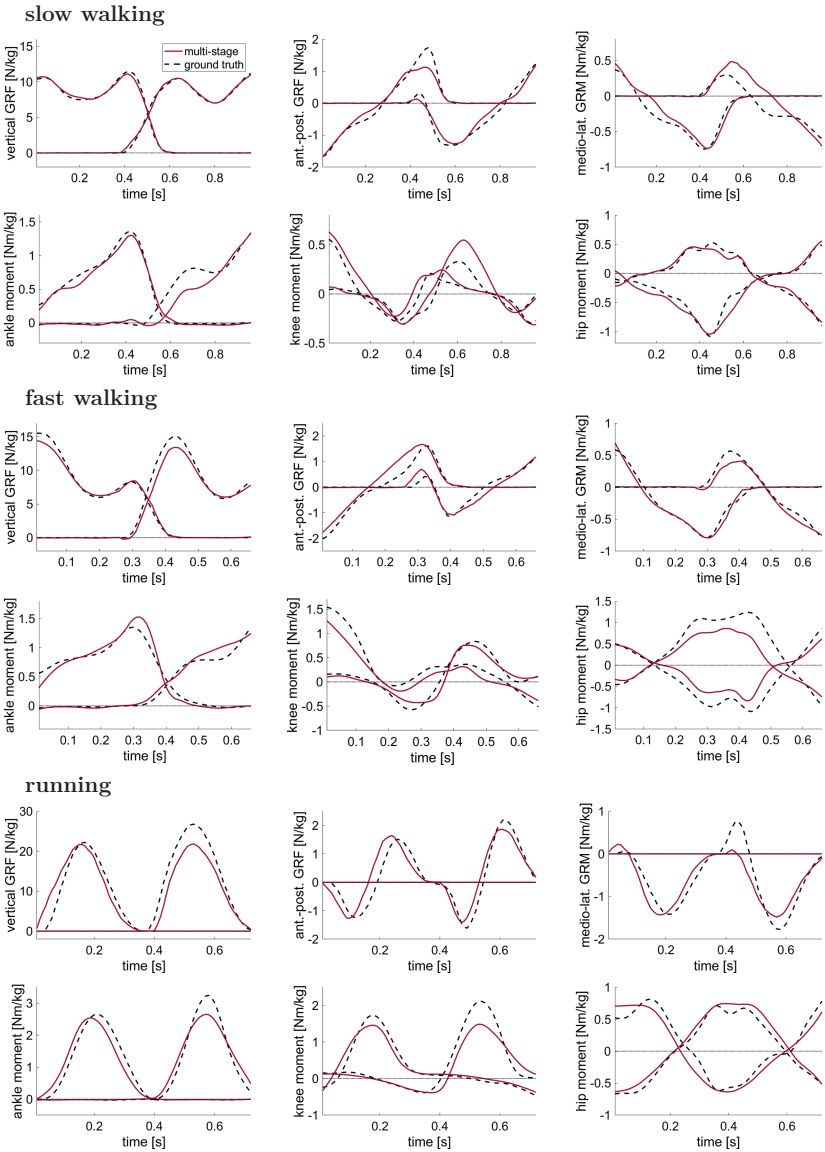


Figure 22: Examples of predicted GRF/M and joint torques. The figure shows multi-stage results using an RF root regression, an NN classification and a ridge control regression for walking and a ridge root regression, an RF classification and an RF control regression for running. The control rRMSEs are $\epsilon_u = 12.1\,\%$, $12.3\,\%$, $14.3\,\%$ (top to bottom).

Table 6: **Multi-stage** regression results. The table shows rRMSEs of the estimated GRF/M and joint torques. The upper part lists the results given complete input $\boldsymbol{\alpha}_x$ and the lower part lists the results with incomplete input $\boldsymbol{\alpha}_{\hat{x}}$ (missing global information). Averaging over methods is indicated by the character Ø.

walking							
input	root reg.	class.	control reg.	ϵ_{f_r} [%] ↓	ϵ_{m_r} [%] ↓	ϵ_τ [%] ↓	ϵ_u [%] ↓
$\boldsymbol{\alpha}_x$	-	RF	Ridge	7.4 (4.2)	17.7 (5.6)	12.1 (4.2)	12.3 (4.0)
	-	SVM	Ø	8.3 (4.0)	16.8 (7.0)	12.8 (4.4)	12.7 (4.4)
	-	RF	Ø	**8.2** (3.4)	**16.6** (5.7)	**12.7** (3.8)	**12.6** (3.5)
	-	NN	Ø	**8.2** (3.4)	**16.6** (5.7)	**12.7** (3.7)	**12.6** (3.5)
	-	Ø	Ridge	**7.5** (4.5)	17.9 (6.7)	**12.2** (4.7)	**12.4** (4.6)
	-	Ø	RF	8.3 (3.2)	**15.8** (6.3)	13.2 (3.9)	12.7 (3.6)
	-	Ø	NN	8.9 (2.8)	16.3 (5.2)	12.8 (3.2)	12.8 (3.1)
$\boldsymbol{\alpha}_{\hat{x}}$	RF	NN	Ridge	7.5 (4.0)	17.4 (6.2)	12.1 (4.3)	12.2 (4.2)
	Ridge	Ø	Ø	8.6 (3.7)	16.8 (6.7)	13.2 (4.3)	13.0 (4.1)
	RF	Ø	Ø	**8.5** (3.7)	16.8 (6.7)	13.2 (4.3)	13.0 (4.1)
	NN	Ø	Ø	**8.5** (3.7)	16.8 (6.7)	13.2 (4.3)	13.0 (4.1)
	Ø	SVM	Ø	8.6 (4.0)	16.9 (7.5)	13.2 (4.8)	13.0 (4.7)
	Ø	RF	Ø	**8.5** (3.5)	**16.8** (6.3)	13.2 (4.0)	13.0 (3.8)
	Ø	NN	Ø	**8.5** (3.5)	**16.8** (6.3)	13.2 (4.1)	13.0 (3.8)
	Ø	Ø	Ridge	**7.5** (4.1)	17.7 (7.4)	**12.2** (4.8)	**12.3** (4.8)
	Ø	Ø	RF	8.4 (3.3)	**16.1** (6.4)	13.4 (4.3)	13.0 (4.0)
	Ø	Ø	NN	9.7 (3.2)	16.7 (6.2)	13.9 (3.5)	13.6 (3.3)
running							
$\boldsymbol{\alpha}_x$	-	RF	RF	13.1 (4.4)	14.7 (5.3)	14.6 (2.9)	14.3 (3.0)
	-	SVM	Ø	14.6 (4.6)	18.9 (7.9)	16.6 (4.8)	16.7 (4.5)
	-	RF	Ø	**13.9** (4.6)	**18.2** (7.5)	**16.1** (4.8)	**16.1** (4.5)
	-	NN	Ø	14.2 (4.8)	18.6 (7.8)	16.4 (4.8)	16.4 (4.6)
	-	Ø	Ridge	14.9 (4.6)	23.5 (8.5)	17.0 (5.8)	17.9 (5.5)
	-	Ø	RF	**13.3** (4.4)	**14.8** (5.2)	**14.8** (2.9)	**14.5** (3.0)
	-	Ø	NN	14.6 (4.9)	17.4 (6.2)	17.4 (4.8)	16.8 (4.0)
$\boldsymbol{\alpha}_{\hat{x}}$	Ridge	RF	RF	13.4 (4.4)	14.8 (5.4)	14.6 (2.9)	14.4 (3.0)
	Ridge	Ø	Ø	**14.7** (5.1)	**19.5** (9.0)	**16.3** (4.6)	**16.6** (4.8)
	RF	Ø	Ø	15.1 (5.1)	20.1 (9.3)	16.7 (4.6)	17.1 (4.9)
	NN	Ø	Ø	14.9 (5.1)	19.8 (9.2)	16.5 (4.6)	16.9 (4.9)
	Ø	SVM	Ø	15.4 (5.0)	20.4 (9.5)	16.9 (4.6)	17.3 (4.9)
	Ø	RF	Ø	**14.4** (5.1)	**19.1** (8.8)	**16.1** (4.6)	**16.3** (4.8)
	Ø	NN	Ø	14.9 (5.1)	19.9 (9.2)	16.5 (4.6)	16.9 (4.9)
	Ø	Ø	Ridge	16.4 (5.1)	27.0 (10.3)	17.9 (5.8)	19.4 (6.0)
	Ø	Ø	RF	**13.6** (4.3)	**15.0** (5.3)	**14.9** (2.9)	**14.7** (3.0)
	Ø	Ø	NN	14.7 (5.4)	17.5 (6.2)	16.7 (4.2)	16.5 (4.0)

Table 7: **Gait phase classification** results for running. The upper part lists the results based on complete input information and the lower part the results without global root coordinates. In the latter case, the values are averaged over the three tested root regression methods. The class labels $l_c = 0, 1, 2, 3$ represent double support left, single support left, double support right and single support right.

		precision ↑				recall ↑			
root reg.	class.	$l_c = 0$	$l_c = 1$	$l_c = 2$	$l_c = 3$	$l_c = 0$	$l_c = 1$	$l_c = 2$	$l_c = 3$
-	SVM	0.83	0.84	0.79	0.81	0.37	0.96	0.42	0.97
-	RF	**0.90**	**0.90**	**0.88**	**0.87**	**0.60**	**0.97**	**0.67**	0.97
-	NN	0.84	0.86	0.84	0.82	0.44	0.96	0.52	0.97
Ø	SVM	0.63	0.80	0.72	0.77	0.20	**0.98**	0.21	**0.98**
Ø	RF	**0.89**	**0.90**	**0.88**	**0.87**	**0.60**	0.97	**0.68**	0.97
Ø	NN	0.85	0.84	0.84	0.81	0.38	0.97	0.41	0.97

NN classification and a ridge control regression. Looking at the averaged results, we can see that the choice of root regression method has no influence on the outcome in the case of walking. For the gait phase classification, the RF and the NN give the best results and the final control regression is best implemented using a ridge regression.

A more detailed evaluation of gait phase classification based on accuracy and recall scores is presented in Table 7. The upper part includes values for classification based on the complete input and the lower part includes the averaged values with prior root regression. It can be seen that the RF outperforms the other classification methods. In particular, the recall of the double support classes c_0 and c_2 is significantly higher, allowing for an accurate control regression during these frames as well.

In contrast to the results of the walking set, the results for running show a clear superiority of the RF, both, for the end-to-end regression and as control regressor in the multi-stage approach. Similar to the walking dataset, the RF as phase classifier is still the best choice and different root regression methods only slightly influence the performance. The ridge regression is not suited as control regressor for this motion type probably due to a larger variability of the data. In general, the errors of the model approximation are amplified by the larger accelerations during running, resulting in a higher uncertainty of the predictive dynamics ground truth compared to walking. This feature combined with the smaller size of the training sets most likely causes the superiority of the RF. The theory will be further explored in an experiment presented in Chapter 6 in which the size of the training set is gradually reduced.

To investigate the influence of the gait phase classification and the contact features, a variant of the method that executes the classification on the pure motion coefficient (without $\boldsymbol{v}_c$) is tested for the walking data. The associated process flow is shown in Figure

Table 8: Ablation study for walking. The multi-stage approach is evaluated without calculation of contact features and compared to the end-to-end and standard multi-stage approach. The upper part lists the results with global input coordinates and the lower part lists the corresponding results without the global information.

$\boldsymbol{v}_c$	root reg.	class.	control reg.	ϵ_{f_r} [%] ↓	ϵ_{m_r} [%] ↓	ϵ_τ [%] ↓	ϵ_u [%] ↓
yes	-	RF	Ridge	**7.4** (4.2)	17.7 (5.6)	**12.1** (4.2)	**12.3** (4.0)
no	-	RF	Ridge	7.5 (4.2)	17.8 (5.5)	12.3 (4.3)	12.4 (4.0)
no	-	-	RF	8.7 (3.0)	**16.1** (6.1)	13.3 (3.6)	12.9 (3.3)
yes	RF	NN	Ridge	**7.5** (4.0)	17.4 (6.2)	**12.1** (4.3)	**12.2** (4.2)
no	-	RF	Ridge	7.6 (4.0)	17.6 (6.2)	12.3 (4.3)	12.4 (4.2)
no	-	-	RF	8.7 (2.9)	**16.4** (6.1)	13.3 (3.6)	13.0 (3.3)

Table 9: Evaluation of the end-to-end regressions using the public dataset by Fukuchi et al. [39]. The table lists rRMSEs of the predicted GRF and joint torques.

control reg.	ϵ_{f_r} [%] ↓	ϵ_τ [%] ↓
Ridge	13.7 (2.7)	16.1 (3.1)
RF	8.8 (2.8)	12.5 (3.2)
NN	9.6 (2.3)	13.9 (3.0)

20 by the dashed arrow. The results of this version are compared to the end-to-end and the standard multi-stage approach in Table 8. The influence of the gait phase classification dominates compared to the inclusion of the contact feature.

5.2.2 *Public Dataset*

The public dataset by Fukuchi et al. [39] includes 44 subjects performing level ground and treadmill walking at various gait speeds. The set encompasses 308 pre-processed sequences with kinematics and dynamics, i. e. joint angles, joint torques and GRF. This data is averaged over several gait cycles. The joint angles are treated as input states and the joint torques and GRF as output controls. After division into 3-frame windows and polynomial fit, the end-to-end regressions are applied to the resulting coefficients and analyzed regarding rRMSE of joint torques and GRF. Due to the larger size of this dataset, a 5-fold cross-validation can be used. The results are listed in Table 9 and predicted curves are shown in Figure 23. The comparison of the methods yields a similar rating as for the PD-set: The RF outperforms the other regression methods, especially ridge regression, whose linear mapping seems to be insufficient for the end-to-end regression task.

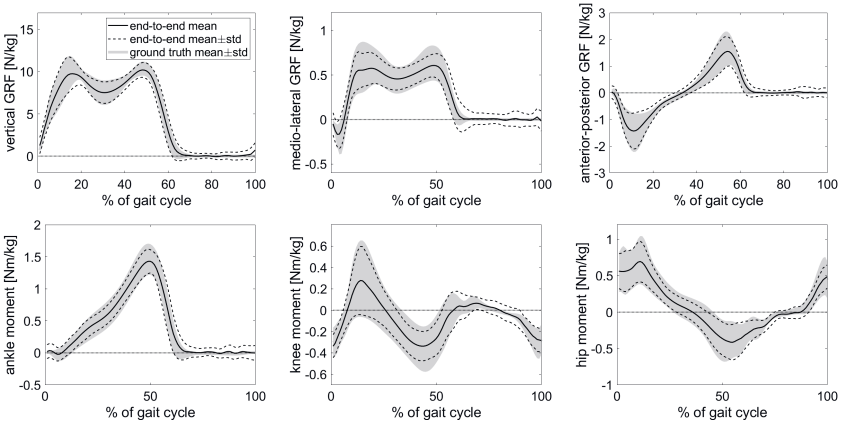


Figure 23: Mean predicted GRF and joint torque curves of the Fukuchi-set [39] using an NN end-to-end regression.

5.2.3 *Application to Reconstructed Motions*

In this Section the proposed methods are tested on reconstructed 3D motion states from 2D joint positions and IMU data. In a first experiment, 19 walking sequences of different subjects taken from the CMU database are considered. The 3D motion states were reconstructed using a non-rigid structure from motion approach by Wandt et al. [148] based on 2D joint positions. Mean regression results are depicted in Figure 24. The figure compares RF end-to-end regression results to multi-stage results implemented with an RF gait phase classification and a ridge control regression. In general, the estimates of the RF are closer to the laboratory data. Both methods are able to produce realistic curve progressions for GRF/M, ankle and hip torques. The multi-stage predictions of the sagittal knee moments, however, deviate significantly from the laboratory data. It should be noted that the distribution of the PD-set is used only as an indication of forces and moments of the same type of motion.

In a second experiment the proposed method is tested as a detector for abnormal gait patterns. The considered sequence is an IMU-based 3D reconstruction result [88]. The recorded movement displays a strong asymmetry: The right leg is kept very stiff, especially at the knee joint, and is dragged behind so to speak. Figure 25 shows frames of the animated regression results together with the corresponding image frames. The abnormality of the gait pattern is analyzed by means of the distribution of mean absolute torques among the limb joints. The torques are averaged over single support phases and depicted in Figure 26. The left side shows the comparison for the left leg and the right

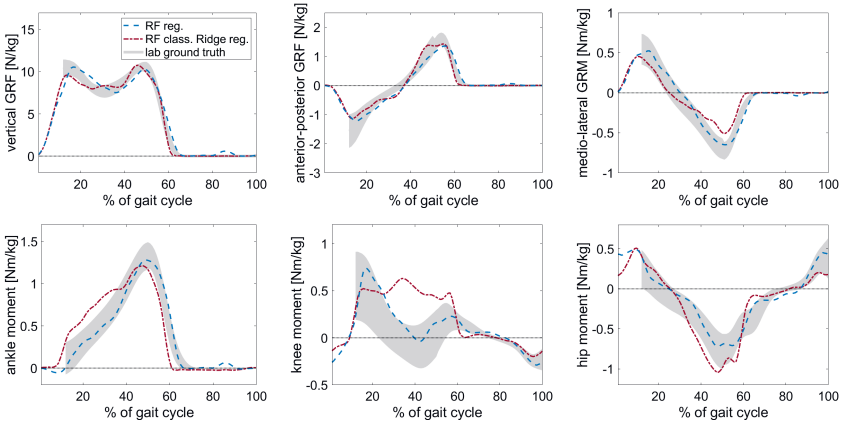


Figure 24: Predicted GRF/M and joint torque components of the reconstructed sequences [148] from the CMU database [18]. The mean predictions are shown as red (multi-stage with RF class. and ridge control reg.) and blue (RF end-to-end reg.) lines. Since there are no ground truth forces included in the CMU data, the distribution of slow walks from the PD-set is used as a reference for realistic profiles (grey area).

side the same comparison for the right leg. The results of the considered sequence are compared to the torque distributions of the PD-set, more precisely, to the predictive dynamics results (target) and the predictions using the multi-stage and the end-to-end approach, respectively. For better comparison, all torque values are normalized to the target values of the PD-set of the left leg (depicted in the left graph on the very left). The torques related to the examined gait sequence exhibit a clear asymmetry between the left and the right leg and thus can be classified as abnormal. Note that the used regression was trained exclusively on healthy, symmetric gait data, which is also reflected in the equality of the graphs in Figure 26. Nevertheless, asymmetric joint moments could be predicted, in part, due to the independent consideration of short time windows.

5.3 DISCUSSION

The comparison between end-to-end regressions and multi-stage approaches shows that the inference of controls is supported by a prior classification into gait phases. This conclusion can be drawn from the quantitative evaluation on the PD-set of walking. With the additional information of the contact state, the multi-stage methods achieve satisfying results and outperform the end-to-end regressions as well as the data-driven optimization method. Applied to 3D reconstructions, that generally lack global coordinates,

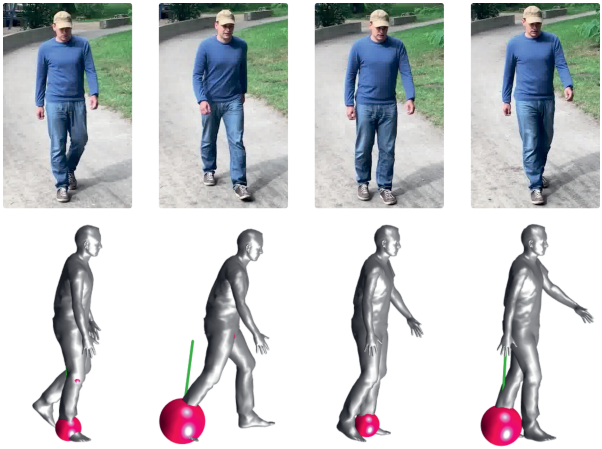


Figure 25: Images and animation frames, generated with SMPL [84], of the asymmetric gait sequence reconstructed based on IMU data [88]. The GRF is illustrated as a green arrow and the joint torques as red spheres. The arrow length and the sphere radii are proportional to the respective predicted values.

the root regression allows for a calculation of global contact features like the foot velocity which further supports the classification of gait phases during walking. The gait phase classification achieves the best results with the RF. This superior performance is probably caused by the higher recall of double support classes, enabling an accurate prediction of the related controls. It can be assumed that the RF can better handle the imbalance of the dataset than the SVM and the NN. In terms of control regressor, the RF yields consistently good results, while the ridge regression dominates on the walking set but fails on the running set. This outcome can most likely be explained by the different dataset sizes. For the small running set (15 subjects performing 132 trials, incl. augmentation, divided into a total of 7156 window samples), the RF clearly outperforms the other tested models in all tasks. Here, the difference between an end-to-end RF and a multi-stage method consisting exclusively of RFs is insignificant. This can be explained by the operating principle of decision trees which in itself performs a cascade of classifications, so that the primary gait phase classification is redundant.

For the walking set, the regression of controls, in particular of GRF and joint torques, is well implemented by a linear ridge regression if the parameters are constrained to a gait phase subset. This implies that, given a contact state, the mapping between the considered motion and force parameters is approximately linear for this dataset. In the case of the GRF, this is easy to understand, since the inverse dynamics calculation is linearly depending on the acceleration of the model's center of mass. The center of mass

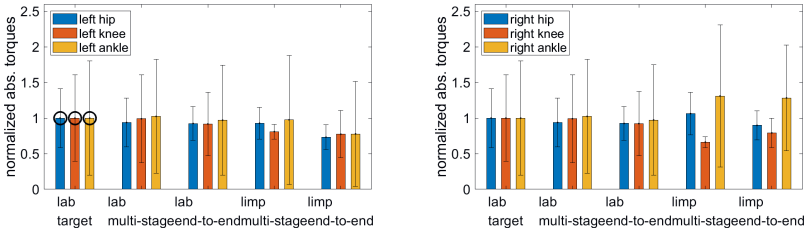


Figure 26: Comparison of mean absolute torques during single support phases. The plot shows sagittal hip, knee and ankle torques of the left leg (left graph) and the right leg (right graph). The analyzed sets are (from left to right) the ground truth of the test lab data, predictions of the test lab data by multi-stage and end-to-end methods and the predictions for the asymmetric gait. Torques are normalized to the corresponding ground truth component of the left leg (indicated by the black circles).

in turn can be approximated by the root of the model whose coordinates and derivatives are included in the input motion parameterization, resulting in an overall linear mapping. Regarding the estimation of joint torques, the high performance of the linear model is surprising since the associated EOM contains nonlinearities in all summands. However, for the comparatively small forces acting during walking movements, these nonlinear effects appear to be negligible. Compared to this, running is a more dynamic form of movement, which contains higher velocities and accelerations, and thus the non-linear relation becomes more apparent. In addition, the errors caused by measurement noise and model simplifications are amplified by the higher values. Therefore, the RF and NN significantly outperform the ridge regression on this dataset.

The end-to-end regressions perform well on the public dataset. Both, the RF and the NN can reliably predict the joint torques and GRF. In contrast to the experiments on the PD-set, the full available motion information was used and the kinematics and dynamics were averaged over several gait periods for each subject prior to regression.

The drop in performance of the multi-stage approach (with ridge control regression) tested on the CMU dataset is due to the inability to bridge the domain gap between the training and the test set. The test set differs from the training set in terms of increased noise caused by prior 3D reconstruction and in terms of average skeletal posture. The latter is an inherent difference found in the 3D poses of both datasets and is probably caused by pre-processing steps, such as marker placement and inverse kinematics as well as the 3D reconstruction. The end-to-end approach implemented by RF predicts curves that are closer to the laboratory data. This is due to the fact that the RF does not extrapolate, so that input deviations caused by the domain gap do not lead to unexpected results.

However, since ground truth forces do not exist for the CMU reconstructions, it remains questionable how close these estimates are to the true forces and moments.

The abnormal gait detection results demonstrate how learning-based inverse dynamics could be utilized to achieve gait analysis in the wild. The deviation of the torque distribution could be used to give a first indication of the evoking impairment. The conducted analysis is only based on the measurement of six IMUs by [88]. Therefore, it could offer a fast and practical procedure to aid in diagnostics.

Although the proposed multi-stage approach as well as some of the end-to-end regression methods yield promising results for the considered PD-set, the performance naturally decreases when the models are confronted with a significant domain gap, as is the case for the 3D reconstructions of the CMU data. In order to improve the generalizability of the regression methods, a larger and broader dataset in terms of motion styles and subject characteristics would be necessary. But as addressed before, dynamic datasets are few and usually restricted to a small number of subjects. Recent research in artificial intelligence investigates the use of self-supervision to make up for a lack of training data. The corresponding approaches involve sophisticated loss functions that guide the optimization of neural networks without the need for excessive training data, as described in Section 2.3. The following chapter presents the realization of self-supervised neural network training using physics-based loss layers. In contrast to the multi-stage approach, described in this chapter, the following method will consist of a single neural network trainable in a practical end-to-end manner with and without joint torque and GRF/M data.

6

SELF-SUPERVISION BY DYNAMICS-BASED LAYERS

This chapter presents self-supervised learning for inverse dynamics of human motion. It is based on an earlier publication from which some text passages and images were taken [172]. As addressed above, the complexity of the recording and pre-processing of human dynamics data leads to a lack of suitably large datasets. This represents a limiting factor for the application of machine learning models in inverse dynamics of human motion. Suitable sets should include multiple subjects and various motion types in order to allow training and testing of generalizable models. In the broader field of machine learning, the problem of missing labeled training data has led to an increasing number of approaches being proposed that use few or weakly labeled data points or even no labeled data at all. Two major categories in this context are semi-supervised and self-supervised learning, as described in Chapter 2. The method presented in this chapter has aspects of both categories, as will become clear after the presentation of the model itself.

The proposed approach is called **Dynamics Network**: A neural network that predicts control coefficients based on motion coefficients, similar to the previous end-to-end approaches, is extended by two physics-based loss layers, the *forward layer* and the *inverse layer*. The network together with the forward layer realizes a full dynamics cycle, as depicted in Figure 27. As before, the inverse dynamics step is implemented as a learnable model, a neural network, that regresses the GRF/M and the joint torques from motion. Based on this prediction and the initial kinematic state of the human model, a simulated motion is generated by integration of the EOM. This step is implemented by the forward layer. The simulated motion can be compared to the input motion in terms of a cyclic loss that is completely independent from measured forces and moments. In addition to this cycle, the inverse layer enables a separate consideration of GRF/M. It executes a bottom-up inverse dynamics calculation starting at the model's feet and ending at the upper-most segment, the end-effector. If the predicted GRF/M perfectly match the input accelerations, the residual force and moment at the end-effector are equal zero. Accordingly, the associated loss function contains the squared values of the residuals. Including both physics-based layers into the model allows for a decoupled control of GRF/M and joint torques during training. The combined loss utilizes the two major approaches to dynamics. Thus, it determines whether the predicted control results from the observed motion and, conversely, whether the observed motion results from the predicted control.

The proposed Dynamics Network is used to realize semi-supervised and completely self-supervised learning. The term semi-supervision means in this context that the number

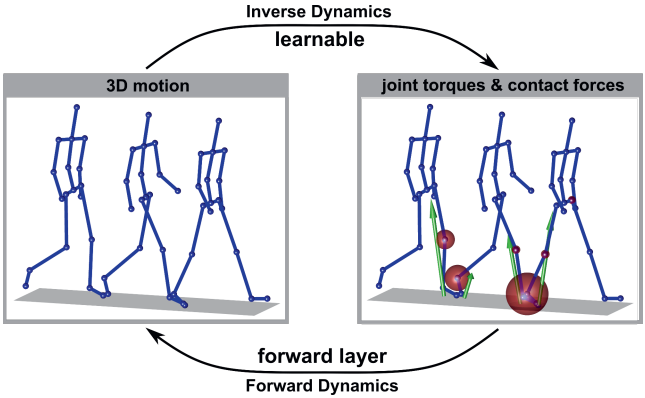


Figure 27: Dynamics Network realizes a cycle consisting of inverse and forward dynamics. The learnable part is represented by a neural network and the forward dynamics simulation is included in a differentiable loss layer.

of labeled training samples (including GRF/M and joint torques) is reduced and extended by unlabeled pure motion samples. While the labeled samples are assessed by means of a standard mean squared error (MSE), the additional motion samples are evaluated by the physics-based loss layers. A gradual reduction of the number of labeled samples shows the benefit of the dataset extension and the proposed loss layers: In contrast to a supervised baseline, the Dynamics Network yields stable results with substantially reduced labeled training set (20 % of the originally included subjects) and can still predict realistic GRF even under complete self-supervision, i. e. without seeing any force data during training. In this case, however, a binary contact loss is necessary to penalize invalid non-zero reaction forces. The generation of contact labels is less complex and expansive compared to a measurement of GRF/M. Automatic gait event detection based on kinematic data for normal walking and running is a largely solved problem [27, 72, 87, 105, 175]. With the application of deep learning and semi-automatic generation of contact labels, ground contact detection is also achieved for arbitrary movement [128, 179].

Furthermore, Dynamics Network is used to realize self-supervised transfer learning between different walking speeds, different motion types (walking and running) and between datasets with varying skeletal characteristics (PD-set and 3D reconstructed CMU sequences). These experiments show how the self-supervision can be leveraged to bridge domain gaps possible in real world scenarios.

Apart from the built-in dynamics layers, the neural network has a modified input layer that distinguishes it from the end-to-end neural network of Chapter 5. It receives a larger input vector including joint and center of gravity trajectories in addition to the motion

states $\boldsymbol{x}$. Since these representations are redundant, an L_1 loss is applied to the weights of the input layer, promoting sparse linkage. Thus, the model is given the freedom to find an optimal mapping from a more variable input space to the control parameters, while the resulting model size is moderated by the L_1 loss. This modification significantly improves the performance of the neural network especially on the more challenging running data. The corresponding evaluation is presented in Section 6.3.4. The resulting network without physics-based loss layers acts as a new baseline in the following.

Finally, this chapter presents a noise experiment that is used to demonstrate the robustness of the proposed Dynamics Network to perturbed motion input signals. Although the network layers introduce dynamics calculations into the training, the models are just as robust as conventional neural networks possessing the same noise-canceling behaviour. In summary, the contributions presented in this chapter are:

1. A new baseline neural network that profits from a variable, high-dimensional input motion parameterization in combination with a L_1 loss penalizing dense input layer connection.

2. A novel forward dynamics layer that numerically integrates the equations of motion and evaluates the deviation of the simulation from the input motion.

3. A novel inverse dynamics layer that propagates reaction forces and moments along the kinematic chain to measure the correspondence between segment accelerations and GRF/M.

4. Since the dynamics layers allow for training on pure motion information, this capacity is used to realize semi-supervised learning and self-supervised domain transfer.

5. The robustness of the proposed model is evaluated with respect to noisy motion input.

6.1 DATASETS

The Dynamics Network is evaluated using the self-recorded predictive dynamics dataset (PD-set) described in Chapter 4 and already used for the quantitative evaluation of the supervised methods of Chapter 5. In contrast to the previous methods, the physics-based layers allow the use of pure motion samples. Therefore, the kinematics of frames that were discarded during the generation of the PD-set, due to non-existent GRF/M information or insufficient convergence of the optimization algorithm, can now be included. The motion states off all sequences and time windows constitute the *motion set*. It includes kinematics $\boldsymbol{x}(t)$, $\boldsymbol{x}_s(t)$ and $\boldsymbol{x}_j(t)$ represented by the polynomial coefficients $\boldsymbol{\alpha}_x$, $\boldsymbol{\alpha}_{x_s}$ and

$\boldsymbol{\alpha}_{x_j}$, respectively, as well as subject specific segment lengths $\boldsymbol{l}$. The additional kinematic representations are the positions and velocities of centers of mass, denoted by $\boldsymbol{x}_s$, and of joints, denoted by $\boldsymbol{x}_j$.

The output of the predictive dynamics optimization is referred to as the *force set* which, additionally to the kinematic information, contains GRF/M and joint torques represented by the associated coefficients $\boldsymbol{\alpha}_u$. Because of the restrictions introduced by the localized force plate measurements, only a fraction of the whole recorded data contains GRF/M, so that the force set is approximately half the size of the motion set regarding sample numbers. Furthermore, a *contact set* is defined which contains the same information as the motion set and additionally includes binary information about the ground contact, i. e. which foot is in contact with the ground. The presented datasets are used in different training modes that represent various levels of supervision. In each case, the used data is shifted to the mean value and normalized by division with the standard deviation.

6.2 DYNAMICS NETWORK

The structure of Dynamics Network is presented in Figure 28. A fully connected neural network executes the inverse dynamics task from motion to joint torques and GRF/M. More precisely, it realizes a function $\boldsymbol{f}_{\text{net}}()$ from an input vector $\boldsymbol{\theta} = [\boldsymbol{\alpha}_x, \boldsymbol{\alpha}_{x_s}, \boldsymbol{\alpha}_{x_j}, \boldsymbol{l}]$ (consisting of motion coefficients and segment lengths) to the control coefficients $\boldsymbol{\alpha}_u$:

$$\boldsymbol{\alpha}_u = \boldsymbol{f}_{\text{net}}(\boldsymbol{\theta})\,. \tag{157}$$

In contrast to the prior methods, the input vector includes coefficients $\boldsymbol{\alpha}_{x_s}$ for segment center of mass positions and velocities and coefficients $\boldsymbol{\alpha}_{x_j}$ for joint positions and velocities on top of the generalized coordinates. The representations of $\boldsymbol{x}$, $\boldsymbol{x}_s$ and $\boldsymbol{x}_j$ are redundant regarding the definition of kinematics, but contain information that facilitates the inference of forces. For example, a mapping from the motion of the centers of gravity to GRF is considerably less complex than a mapping from joint angle kinematics to GRF. In order to reduce the redundancy a L_1 loss is applied to the weights of the input layer, which favors sparse linkage. In Section 6.3.4 it is shown that a network with this input structure performs better than, both, a network that only uses generalized motion states and a network that receives all motion coefficients without L_1 penalization. Following the input layer, the network consists of two fully connected layers of 200 neurons for walking and 120 neurons for running (to account for the smaller number of training examples). Leaky-ReLu activations [86] are used in the hidden layers. The described network, trained in a supervised manner using the MSE of forces and moments, acts as a baseline in this chapter.

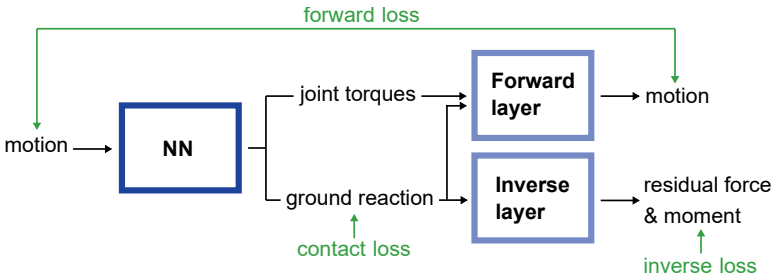


Figure 28: Schematic structure of Dynamics Network. The output of the neural network is processed using forward and inverse layer to achieve training independent from GRF/M and joint torque data. The contact loss can be additionally applied to the GRF/M to penalize invalid external forces.

The integration of EOM is implemented as a differentiable function and referred to as the forward layer. It yields simulated motion states based on the network output. A detailed description is given in Section 6.2.1. The combination of network and the forward layer build a cycle which enables the minimization of a loss between kinematic states. The additional inverse layer evaluates the consistency of the input accelerations with the predicted GRF/M by calculating force and moment transfer along the kinematic chain. The residuals at the last segment yield the corresponding loss. A complete description follows in Section 6.2.2.

Furthermore, the network can be trained in a supervised manner, using an MSE

$$L_{\text{MSE}} = \left\| \boldsymbol{\alpha}_u - \boldsymbol{\alpha}_u^{\text{true}} \right\|_2^2 \tag{158}$$

of the predicted control coefficients referred to as *MSE loss*. This loss will be used in a baseline network that is trained completely supervised and for semi-supervised training of Dynamics Network, which includes labeled examples as well. To gradually reduce the level of supervision, a contact loss is implemented, that penalizes GRF/M during time frames with no ground contact. This loss only requires binary information instead of full ground reactions. It is described in Section 6.2.3.

In summary, the method operates with four different loss functions that can be activated separately or in combination depending on the nature of the current training sample. On this basis, different training modes are defined in Section 6.2.4. These modes implement various levels of supervision and work with different subsets of the data. First, however, the dynamics layers will be presented in detail.

6.2.1 *Forward Layer*

In this section, the forward dynamics simulation and the implementation as a neural network layer is described. A simple skeletal model and a basic numerical integration technique are chosen in order to maintain relatively low computational complexity. This is necessary to facilitate the integration in neural network training. The skeletal leg model has already been presented in Chapter 4. As before, the EOM is formulated by means of the TMT-method and results in

$$\boldsymbol{\mathcal{M}}(\boldsymbol{q}(t),\boldsymbol{l})\ddot{\boldsymbol{q}}(t) = \boldsymbol{\mathcal{F}}(\boldsymbol{q}(t),\dot{\boldsymbol{q}}(t),\boldsymbol{\tau}(t),\boldsymbol{f}_c(t),\boldsymbol{l})\,. \tag{159}$$

It can be rewritten as a 1st order differential equation for the state vector $\boldsymbol{x}(t) = [\boldsymbol{q}(t),\dot{\boldsymbol{q}}(t)]$:

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{\boldsymbol{q}}(t) \\ \boldsymbol{\mathcal{M}}(\boldsymbol{q}(t))^{+}\boldsymbol{\mathcal{F}}(\boldsymbol{x}(t)) \end{bmatrix} =: \boldsymbol{f}(\boldsymbol{x}(t))\,, \tag{160}$$

with the Moore-Penrose inverse $(\cdot)^{+}$. For the sake of clarity, only the dependence on coordinates $\boldsymbol{q}$ and states $\boldsymbol{x}$ is indicated here. Starting at the initial state $\boldsymbol{x}_0 = \boldsymbol{x}(t_0)$, the forward dynamics solution at time t is

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{f}(\boldsymbol{x}(t))dt'\,. \tag{161}$$

This initial value problem can be solved by numerical integration. During this process, an acceleration, force or moment error propagates with the squared integration time. In order to reduce this high sensitivity for neural network training, a damping factor is applied to $\boldsymbol{f}(\boldsymbol{x})$. The damping starts at a threshold value which is based on the standard deviation $\boldsymbol{\sigma}_{\dot{x}}$ of the absolute velocities and accelerations $|\dot{\boldsymbol{x}}(t)|$ contained in the training set and on their maximum value $\boldsymbol{m}_{\dot{x}}$ in the current input sample. Each component f_j of Eq. (160) is damped by

$$d_j(\boldsymbol{x}) = d_{\min} + (1-d_{\min})\exp\left\{-\max\left(\frac{|f_j(\boldsymbol{x})| - m_{\dot{x},j} - k\sigma_{\dot{x},j}}{k\sigma_{\dot{x},j}}, 0\right)\right\}\,. \tag{162}$$

If the absolute value $|f_j|$ exceeds the threshold $m_{\dot{x},j} + k\sigma_{\dot{x},j}$, the value of d_j starts to decrease from 1 to $d_{\min}$ with exponential progression. The vector $\boldsymbol{d}$ is included into the EOM by a Hadamard product:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) \odot \boldsymbol{d}(\boldsymbol{x}(t))\,. \tag{163}$$

The factor k in Eq. (162) controls the slope of the curve and the threshold for the decrease from one. In simple terms, it extends the range of acceptable velocities/accelerations to k times the standard deviation. The value is set to $k = 10$. The minimum value indicates the maximum amount of damping. To obtain non-vanishing derivatives, a value of $d_{\min} = 0.2$ is selected. These settings result in stable simulations that can still be optimized during training.

In order to keep the computation time as low as possible, Euler's method with constant step size Δt is chosen for numerical integration. In the following, the discrete time points are specified by the indexing $(\cdot)_t$ and the composition of discrete signals of several frames is denoted by $(\cdot)_{i\ldots j} = [(\cdot)_i, (\cdot)_{i+1}, \ldots, (\cdot)_{j-1}, (\cdot)_j]$. With this notation, the function realized by the forward layer can be written as

$$\boldsymbol{x}_{1\ldots n} = FD(\boldsymbol{x}_0, \boldsymbol{l}, \boldsymbol{f}_{c0\ldots n-1}, \boldsymbol{\tau}_{0\ldots n-1}, \boldsymbol{m}_{\dot{x}}) \,. \tag{164}$$

It executes $n = T - 1 =$ (window size $-$ 1) Euler steps

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \Delta t \Big[\boldsymbol{f}(\boldsymbol{x}_{t-1}) \odot \boldsymbol{d}(\boldsymbol{x}_{t-1}) \Big] \,. \tag{165}$$

Based on the resulting states, the forward loss is defined as

$$L_{\text{forward}} = \frac{1}{n} \sum_{t=1}^{n} \left(\left\| \boldsymbol{x}_t - \boldsymbol{x}_t^{\text{true}} \right\|_2^2 + w \left\| \boldsymbol{d}(\boldsymbol{x}_{t-1}) - \boldsymbol{1} \right\|_2^2 \right) \tag{166}$$

with an additional term to penalize damping vectors with components smaller than one and a weighting factor $w = 100$.

For the backpropagation during training, the partial derivatives $\frac{\partial L_{\text{forward}}}{\partial \boldsymbol{\alpha}_u}$ of the loss with respect to the neural network output $\boldsymbol{\alpha}_u$ need to be known. This can be achieved either by automatic differentiation, which is commonly included in deep learning frameworks, or by explicit computation using sensitivity analysis as described below. Automatic procedures allow differentiation of non-continuous functions, like the maximum function used in Eq. 162, by treating them piecewise analytically [74].

An explicit calculation of the partial derivatives for the forward layer is based on the application of the chain rule yielding

$$\frac{\partial L_{\text{forward}}}{\partial \boldsymbol{\alpha}_u} = \sum_{t=1}^{n} \frac{\partial L_{\text{forward}}}{\partial \boldsymbol{x}_t} \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\alpha}_u} \,. \tag{167}$$

The derivation of the first factor and the last factor of this expression is straightforward. The second factor $\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{u}}$ includes the gradients of simulated states. For its calculation, the

EOM in Eq. (163) is partially differentiated by the controls $\boldsymbol{u}$, resulting in a second differential equation,

$$\frac{\partial \dot{\boldsymbol{x}}(t)}{\partial \boldsymbol{u}} = \frac{d}{dt}\frac{\partial \boldsymbol{x}(t)}{\partial \boldsymbol{u}} = \frac{\partial}{\partial \boldsymbol{u}}\Big[\boldsymbol{f}(\boldsymbol{x}(t)) \odot \boldsymbol{d}(\boldsymbol{x}(t))\Big] \tag{168}$$

for the matrix $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}}$. Numerical integration of this differential equation supplies the desired partial derivatives for the backpropagation through the forward layer.

Therefore, the forward pass through the forward layer involves numerical integration of Eq. (163) and Eq. (168) with storage of the intermediate results $\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{u}}$. During the backward pass, the gradient of the loss is calculated by substituting the stored factors into the chain rule of Eq. (167).

6.2.2 *Inverse Layer*

The inverse layer receives the input motion, which is also fed into the network, and the predicted GRF/M. It propagates forces and moments along the chains of the kinematic tree in a bottom-up manner. For this purpose, each segment is considered in a free-body diagram where the sum of all acting forces and moments must explain the observed linear and angular accelerations of the segment. Thus, successively, the forces/moments at the proximal joint can be determined from the previously calculated forces/moments at the distal joint. In this process, the GRF/M are applied to the centers of gravity of the model's feet. The formulas result from the Newton-Euler equation [33]. For segment s the force $\boldsymbol{F}_p$ acting on the proximal joint is given by

$$\boldsymbol{F}_p = m_s(\boldsymbol{a}_s - \boldsymbol{g}) - \boldsymbol{F}_d \tag{169}$$

with the distal force $\boldsymbol{F}_d$, the segment mass m_s, the segment acceleration $\boldsymbol{a}_s$ and the gravitational acceleration $\boldsymbol{g}$. The moment $\boldsymbol{M}_p$ exerted at the proximal joint can be calculated in a similar way:

$$\boldsymbol{M}_p = \boldsymbol{I}_s\boldsymbol{\alpha}_s + \boldsymbol{\omega}_s \times (\boldsymbol{I}_s\boldsymbol{\omega}_s) - \boldsymbol{M}_d - \sum_{j=p,d} \boldsymbol{r}_j \times \boldsymbol{F}_j \tag{170}$$

where $\boldsymbol{M}_d$ denotes the distal moment, $\boldsymbol{I}_s$ is the tensor of inertia for the considered segment and $\boldsymbol{\omega}_s$ and $\boldsymbol{\alpha}_s$ are its angular velocity and acceleration. The cross products account for the moments resulting from the forces acting on the joints. Here, $\boldsymbol{r}_j$ is the lever arm from the center of mass of the segment to the position of joint j.

Repeated application of these equations finally yields a residual force $\boldsymbol{F}_{\text{res}}$ and a residual moment $\boldsymbol{M}_{\text{res}}$ which remain at the end of the kinematic tree, the center of mass of the

upper body. To summarize, the inverse layer realizes a function $ID()$ from input motion and predicted GRF/M to residual forces and moments:

$$(\boldsymbol{F}_{\text{res}_{1\dots n}}, \boldsymbol{M}_{\text{res}_{1\dots n}}) = ID(\boldsymbol{x}_{1\dots n}, \boldsymbol{l}, \boldsymbol{f}_{c_{1\dots n}})\,. \tag{171}$$

If the accelerations of the input motion perfectly match the GRF/M, these residuals are zero. Accordingly, the inverse loss is defined as

$$L_{\text{inverse}} = \frac{1}{n}\sum_{t=1}^{n}\left(|\boldsymbol{F}_{\text{res}_t}\|_2^2 + \|\boldsymbol{M}_{\text{res}_t}\|_2^2\right)\,. \tag{172}$$

The calculation of gradients for the inverse layer is straight forward, since there is no dependence between the residuals of different time points:

$$\frac{\partial L_{\text{inverse}}}{\partial \boldsymbol{\alpha}_u} = \frac{2}{n}\sum_{t=1}^{n}\left(\boldsymbol{F}_{\text{res}_t}^T\frac{\partial \boldsymbol{F}_{\text{res}_t}}{\partial \boldsymbol{f}_{c_t}} + \boldsymbol{M}_{\text{res}_t}^T\frac{\partial \boldsymbol{M}_{\text{res}_t}}{\partial \boldsymbol{f}_{c_t}}\right)\frac{\partial \boldsymbol{f}_{c_t}}{\partial \boldsymbol{\alpha}_u}\,. \tag{173}$$

6.2.3 *Contact Loss*

The dynamics layers, described above, ensure that the predicted forces together with the input motion fulfill the model EOM. However, this alone is not sufficient for realistic predictions because there is no automatic contact detection to constrain GRF/M. In order to warrant valid GRF/M that are only greater than zero if ground contact exists, a contact loss is introduced. This loss function only requires binary information about the contact state at each foot. This kind of label is a lot easier to acquire than full 3D GRF/M, e. g. by estimation from kinematic features like foot velocities and knee angle curve progressions [105, 163].

Let $c_{i,t} \in \{0, 1\}$ be the contact state of foot i at time frame t with $c_{i,t} = 0$ if the foot touches the ground and $c_{i,t} = 1$ otherwise. The counterintuitive definition is chosen to allow a simple formulation of the contact loss as

$$L_{\text{contact}} = \frac{1}{n}\sum_{i=1}^{2}\sum_{t=1}^{n}\|c_{i,t}\boldsymbol{f}_{c_{i,t}}\|_2^2\,. \tag{174}$$

The function penalizes contact forces and moments during swing/flight phases.

6.2.4 *Training Modes*

As described above, the model allows the use of various loss functions. During training of the network, the application of the MSE loss, the forward loss and the inverse loss is done

alternatingly. The L_1 loss of the input layer is added to all three alternating functions, in contrast to the contact loss, which is only added to the forward and inverse loss. To balance the effect of the different loss functions, weight factors are dynamically adapted. The forward loss is used as a reference and all remaining losses L_i are multiplied by weights resulting from the respective average ratio L_{forward}/L_i during the last epoch.

Network parameters are optimized using SGD [9] with the momentum set to zero and a batch size of one. These settings were chosen to account for the sensitivity of the forward simulation and their suitability was confirmed by validation. Further stabilizing measures are the damping already presented in Section 6.2.1 and gradient clipping. The initial learning rate is set to 10^{-2} if only the MSE loss is used and to 10^{-3} if the physics-based layers are included. It is reduced by 0.1 every 4th epoch. Moreover, during optimization of the forward loss, the learning rate of the output layer that generates GRF/M is 0.1 times smaller than the learning rate of the joint torque output layer. This distinction is made to counteract the tendency of the model to primarily adjust the GRF/M in order to produce balanced simulated motions, a behavior, in part caused by the normalization parameters from the training data. The multiplication with larger standard deviations amplifies the value of exterior forces and the non-zero mean introduces a corresponding bias. By reducing the related learning rate, the forward layer is mainly used to optimize joint torques while GRF/M are predominantly controlled by the inverse layer in combination with the contact loss.

Based on the general training scheme presented above, the following individual training modes and resulting models are defined:

1. **Baseline**: The network is optimized using the MSE loss and the L_1 loss to encourage a sparse input layer connection. The model is trained on the force set.

2. **F-net**: This model extends the baseline by including the forward loss and applying it to samples of the unlabeled motion set. The corresponding training mode is categorized as semi-supervised.

3. **cFI-net**: This model is trained using all presented loss functions either in a semi-supervised or a completely self-supervised manner, i. e. with or without inclusion of the MSE loss.

6.3 EXPERIMENTAL EVALUATION

In this section, the proposed methods are evaluated regarding their capability to learn exterior GRF/M and interior joint torques from motion. In particular, semi-supervised learning on small labeled training sets, completely self-supervised learning and domain

Table 10: Quantitative evaluation of supervised dynamics learning for the gait dataset. The table shows rRMSE values ϵ_{f_r}, ϵ_{m_r}, ϵ_τ and ϵ_u related to the GRF, GRM, the joint torques and the whole controls u as well as the EOM error e_{EOM}.

motion	method	ϵ_{f_r} [%] ↓	ϵ_{m_r} [%] ↓	ϵ_τ [%] ↓	ϵ_u [%] ↓	e_{EOM} $[Nm/kg]$ ↓
walk	Lv et al. [85]	12.8 (5.5)	18.6 (6.3)	16.4 (5.0)	16.1 (4.6)	0.258 (0.144)
	Multi-stage	7.4 (4.2)	17.7 (5.6)	12.1 (4.2)	12.3 (4.0)	0.128 (0.040)
	Baseline net	7.8 (2.0)	**16.1** (4.5)	12.3 (3.1)	12.2 (2.6)	0.116 (0.029)
	F-net	7.6 (2.1)	17.7 (5.3)	**11.9** (3.1)	12.2 (3.0)	0.108 (0.030)
	cFI-net	**7.0** (1.9)	16.9 (4.7)	**11.9** (3.0)	**11.9** (2.7)	**0.099** (0.024)
run	Lv et al. [85]	17.2 (4.2)	18.4 (5.7)	19.7 (3.8)	19.0 (3.7)	0.433 (0.153)
	Multi-stage	13.1 (4.4)	14.7 (5.3)	14.6 (2.9)	14.3 (3.0)	0.241 (0.030)
	Baseline net	**10.5** (3.6)	**14.2** (4.2)	**13.0** (2.1)	**12.7** (2.0)	0.250 (0.031)
	F-net	12.2 (3.1)	19.3 (5.8)	14.2 (3.2)	14.8 (3.3)	0.251 (0.036)
	cFI-net	10.9 (2.9)	16.9 (5.1)	13.5 (2.8)	13.7 (2.9)	**0.231** (0.032)

transfer are investigated. To quantify performance, the error measures introduced in 6.3 and dataset splits (into training, validation, and test sets) listed in Table 3 are used (similar to the evaluation of the previous chapter). In addition, the violation of the EOM by the estimated controls $\boldsymbol{u}_t$ with $t = 1, \ldots, T$ is assessed by the error measure

$$e_{\text{EOM}} = \frac{1}{T} \sum_{t=1}^{T} \| \boldsymbol{\mathcal{M}} \ddot{\boldsymbol{q}}_t^{\text{true}} - \boldsymbol{\mathcal{F}}(\boldsymbol{x}_t^{\text{true}}, \boldsymbol{u}_t) \|_2 \tag{175}$$

which is closely related to the energy function E_{EOM} used in the predictive dynamics optimization formulation of Eq. (143).

6.3.1 *Comparison in the Supervised Setting*

The proposed models are compared to the supervised learning methods presented in the previous chapter, more precisely, to the data-driven optimization [85] and the best performing multi-stage approach. In this experiment F-net and cFI-net are trained with both labeled and unlabeled data, i. e. alternating between supervised learning on the force set and self-supervised learning on the motion set. Table 10 lists the corresponding results including the violation of EOM in terms of e_{EOM}.

For walking, even when the complete available labeled data is used to train the networks in a supervised manner, the addition of the forward and inverse loss improves the overall performance (cf. ϵ_u of all controls). In the case of running, the proposed baseline network yields the lowest rRMSEs of the predicted components. In contrast to the supervised

methods of Chapter 5, the baseline network receives a high-dimensional, redundant motion input and additionally optimizes the L_1 loss of the input layer weights which significantly boosts the performance. In terms of EOM compliance, the inclusion of both dynamics layers in the training (cFI net) provides a significant improvement over the other approaches for both motion types: The EOM error e_{EOM} is reduced by 15 % for walking and 8 % for running compared to the baseline network.

6.3.2 *Semi-Supervision with Small Labeled Datasets*

One benefit of the proposed Dynamics Networks, F-net and cFI-net, is to realize learning of human locomotion dynamics on small datasets with lower risk of overfitting to the training examples. The models introduced in Section 6.2.4 are now compared to each other regarding their capability to operate on a training set with decreasing number of labeled data points from the force set. The reduction of the force set is subject-wise: If a subject ID is excluded, this applies to all movement trials performed by this subject. The models F-net and cFI-net can be trained in a semi-supervised manner using training sets that are composed of examples from the force set and the motion set (as well as the contact set in the case of cFI-net). The motion and contact training sets are always included to their full extend, so that the dynamics-based models receive data of all subjects included in the original training set while the fully supervised baseline network and the multi-stage method only receive data from the reduced set of subjects. This way, the effect of the higher training data variability and the benefit of the dynamics layers are evaluated. For a fair comparison, the multi-stage method is implemented using only RF for all steps, since it can handle small training sets better than the other models that were tested in Chapter 5.

Figure 29 shows the performance drop with decreasing size of the included force set in terms of rRMSE values of all controls $\boldsymbol{u}$ and of GRF, GRM and joint torques separately. The x-axes display the fractions of the used force set related to the original training set regarding number of subjects. For example, a fraction of 0.2 refers to the inclusion of 3 of 13 subjects for walking and 2 of 9 subjects for running. A corresponding visualization of the mean regressed GRF/M and joint torque curves at this force set fraction can be seen in Figure 30. The upper part shows the results for walking and the lower part the results for running. To assess the validity of the estimated components with respect to the physical model, Figure 31 compares the violation of the EOM. The plots include a *ground truth* value which refers to the window-wise PDO results.

In the case of walking, the Dynamics Networks consistently outperform the baseline network and the multi-stage approach regarding the combined consideration of all controls. Especially cFI-net yields very stable results and can predict the GRF with high accuracy

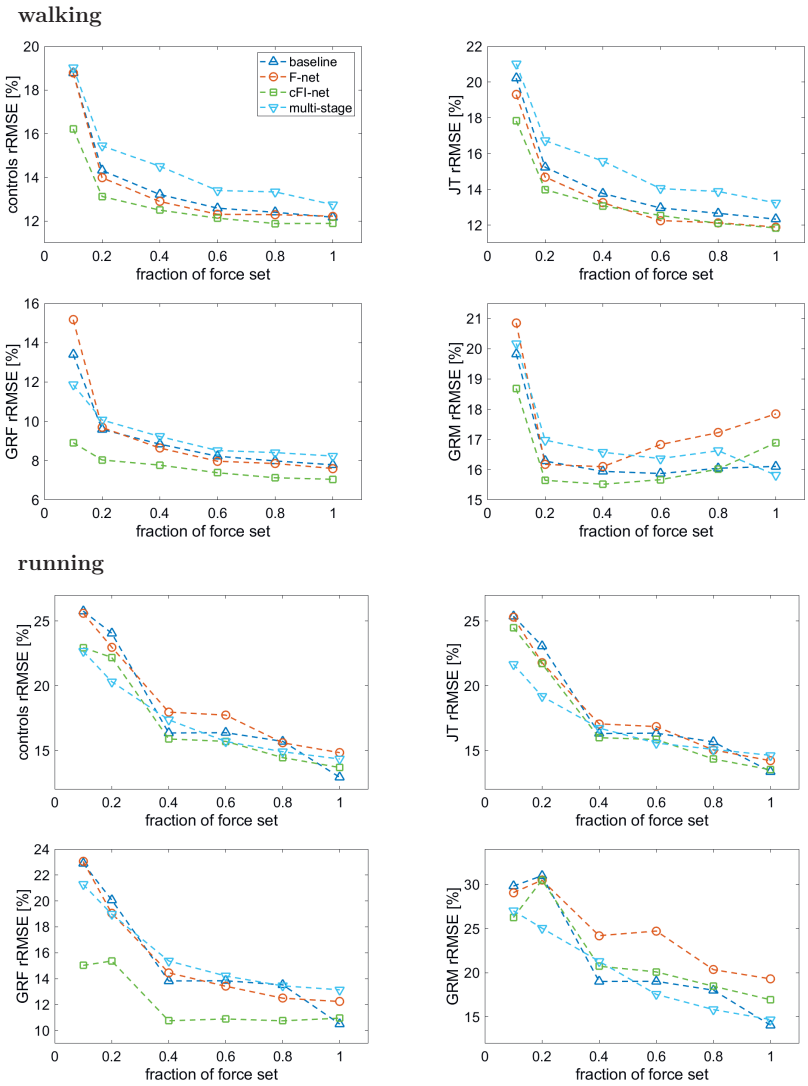


Figure 29: Reduction of the labeled training set. The figure includes rRMSEs of all controls and of joint torques, GRF and GRM separately. The error measures are plotted against the proportion of subjects still contained in the labeled training set in relation to the original number.

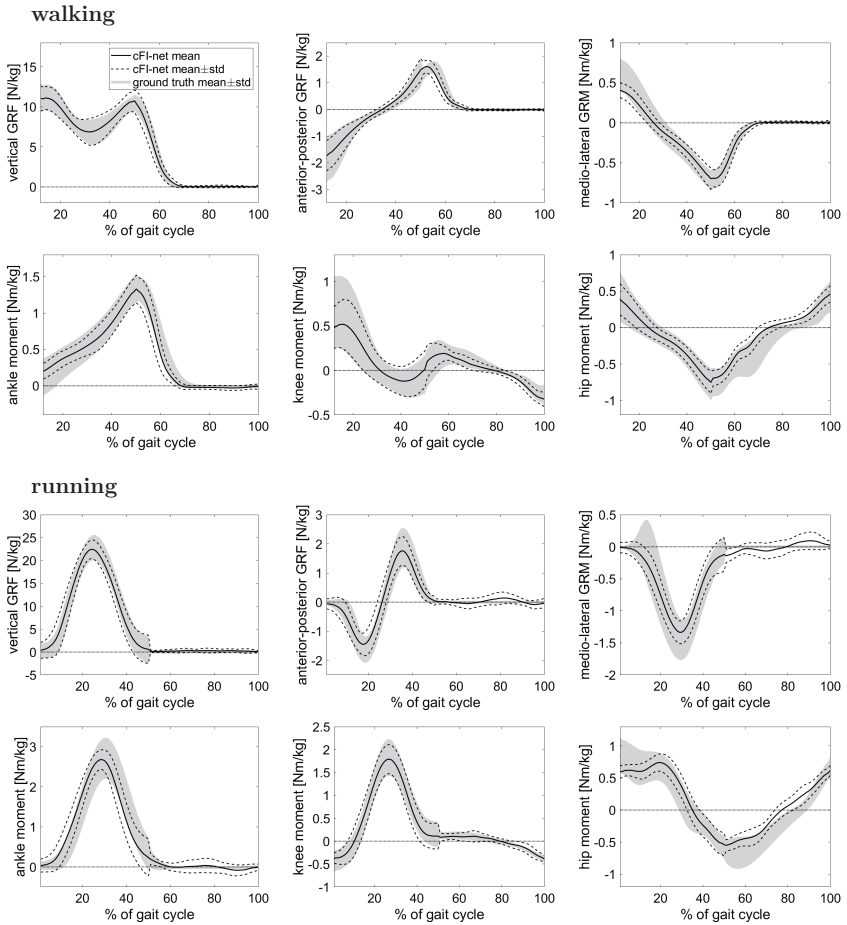


Figure 30: Average estimated control curves for walking (top rows) and running (bottom rows). The used model is cFI-net trained with 20 % of the force set. The grey area shows the distribution of the ground truth test set. The black line indicates the mean regression results and the dashed lines the related standard deviation.

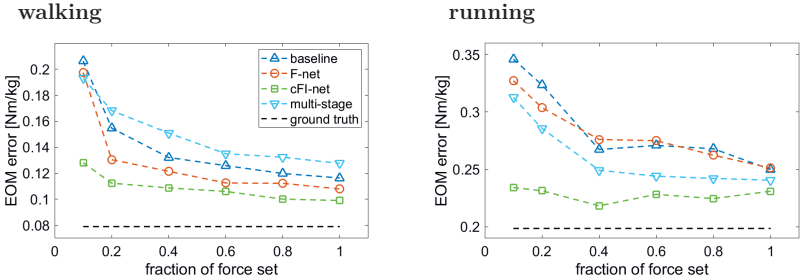


Figure 31: The EOM error e_{EOM} that results when estimated controls and input motion states are inserted into the EOM. The regression results are compared to the window-wise predictive dynamics optimization results as *ground truth*.

even under substantial reduction of the force set. Note that the multi-stage method used for comparison includes RF gait phase classification and RF control regression, since the best method under complete supervision (given a complete force set) uses Ridge control regression which is not suitable for such small training sets. For running, the performance evaluation is less clear. The RF multi-stage approach is still the leading method regarding the stability of the average rRMSE among all control components. It yields the lowest errors at the smallest force set sizes. However, there exists a range (between fractions of 0.4 to 0.8), where cFI-net produces slightly better estimates and in terms of GRF it clearly outperforms the other methods.

With respect to EOM error (cf. Figure 31), cFI-net, which incorporates an inverse dynamics analysis into the training process, is the superior method for both types of movement. F-net, on the other hand, which uses only the forward layer, improves the satisfaction of EOM (compared to the baseline) on the walking data but not on the running data.

6.3.3 *Domain Adaptation*

In addition to the stable performance on small datasets, the applicability to data that systematically deviates from the training data is of interest. Such a scenario is subject of domain adaptation where the training set represents the *source domain* and the deviating test set the *target domain*. The two domains may differ in terms of the input/feature space, the output space and/or the related marginal probability distributions of the data. The difference is generally referred to as domain gap.

The independence of the proposed model from exhaustive dynamic data provides tremendous opportunities for bridging domain gaps that would otherwise require cumbersome

data recording and pre-processing. In this section, the Dynamics Network is applied to realize self-supervised domain expansion and transfer. The considered domain gaps include different walking speeds, different movement types and the deviation of 3D reconstructed motions from the kinematic characteristics of the PD-set.

Domain Expansion to Different Walking Speeds

During the recording of the walking data, the subjects were asked to walk slowly at first and then quickly, which resulted in the dataset being roughly divided into two classes. This circumstance is now used to evaluate domain expansion. The data at one speed level acts as source domain and is learned with supervision, while the other speed level represents the target domain and is included by self-supervision. Similar to the previous experiment, the different loss functions are applied alternately during training: Samples of the source domain are evaluated using the MSE loss and samples of the target domain are processed by the physics-based loss layers. In this way, the model is expanded to cover the dynamics of the target domain without requiring ground truth of forces and moments.

The domain expansion is implemented using cFI-net for both directions, i. e. from slow walking to fast walking and the other way around. The resulting models are evaluated on, both, the target domain and the source domain and compared to the related baseline networks that are only trained with the MSE loss. The additional evaluation on the source domain is performed to check if the performance is degraded by the inclusion of the target samples. The results are listed in Table 11.

In both directions, the self-supervised inclusion of samples from the target domain clearly improves performance in this domain and thus helps to bridge the domain gap. Considering the evaluation in the source domain, an expansion from slow walking to fast walking leads to a slight loss of performance. In the inverse direction, however, the expansion to slow walking even improves the predictions on the fast walking test sequences compared to the baseline exclusively trained on this domain. This difference might be due to the smaller size of the fast walking subset and the fact that it contains less double support frames since walking at a higher speed leads to larger steps and makes it difficult to hit both force plates in succession. Therefore, the model can still benefit from the extension to the comparatively information-rich dataset of slow walking.

An exemplary, visual comparison is given in Figure 32. The left-hand side depicts an example of slow walking and the right-hand side an example of fast walking. The individual plots contain the predictions of the baseline network trained on the other domain, respectively, and the predictions of cFI-net that realizes domain expansion by self-supervision. The corresponding control rRMSE values achieved with cFI-net are $\epsilon_u = 13.0\,\%$ for the slow walk and $\epsilon_u = 19.5\,\%$ for the fast walk.

Table 11: Domain extension results of slow and fast walking in terms of rRMSE ϵ_{f_r}, ϵ_{m_r}, ϵ_τ and ϵ_u for predicted GRF/M, joint torques and all control parameters combined. In addition, the violation of EOM is listed in terms of e_{EOM}.

	ϵ_{f_r} [%] ↓	ϵ_{m_r} [%] ↓	ϵ_τ [%] ↓	ϵ_u [%] ↓	e_{EOM} [$\frac{Nm}{kg}$] ↓
test domain: slow walk					
superv. slow	**7.3** (1.6)	16.6 (3.8)	**12.2** (2.5)	**12.1** (2.1)	**0.246** (0.092)
superv. slow + self-superv. fast	7.9 (2.0)	**16.3** (5.4)	12.7 (2.8)	12.5 (2.7)	0.248 (0.084)
superv. fast	11.3 (2.4)	18.5 (4.1)	15.0 (2.5)	15.0 (2.1)	0.289 (0.077)
superv. fast + self-superv. slow	**7.7** (1.8)	**17.9** (4.3)	**13.4** (2.8)	**13.2** (2.4)	**0.242** (0.081)
test domain: fast walk					
superv. fast	9.2 (1.9)	**13.8** (5.1)	15.5 (3.2)	13.9 (2.7)	0.509 (0.063)
superv. fast + self-superv. slow	**7.5** (1.8)	13.9 (4.6)	**14.5** (3.1)	**13.0** (2.7)	**0.478** (0.064)
superv. slow	10.0 (1.9)	16.1 (5.4)	16.2 (3.3)	14.9 (3.0)	0.532 (0.066)
superv. slow + self-superv. fast	**8.6** (2.2)	**14.2** (5.1)	**15.3** (3.0)	**13.7** (2.7)	**0.488** (0.059)

Complete Self-Supervision and Transfer to New Movement Types

Technically, the training of cFI-net can be done completely self-supervised without the force set. Then the model relies entirely on the dynamics layers in combination with the contact loss. The latter, however, is crucial in order to enforce single support, since both layers consistently assume EOM for double support without explicit modelling of dynamic contact (such as initial contact detection and activation of contact constraints or reaction and friction forces). Furthermore, the normalization scheme has to be adapted due to the lack of ground truth forces and moments. Instead of normalizing all controls to the statistics of the training set, only the largest control component, the vertical GRF, is normalized by $F_y \leftarrow (F_y - 5)/5$. This way the interval $[0, 10]$ is mapped to $[-1, 1]$. The value of 10 is close to the normalized reaction force at rest which corresponds to the gravitational acceleration $9.81\,N/kg$. This training mode is referred to as *from scratch* in Table 12.

The proposed Dynamics Network also enables self-supervised domain adaptation between movement types. Here, the transfer between walking and running is considered with ground truth forces and moments existing for the source domain but not for the target domain. The ground truth data is used to initialize the network parameters by means of supervised learning with the MSE loss. The model is then trained using the motion and contact set

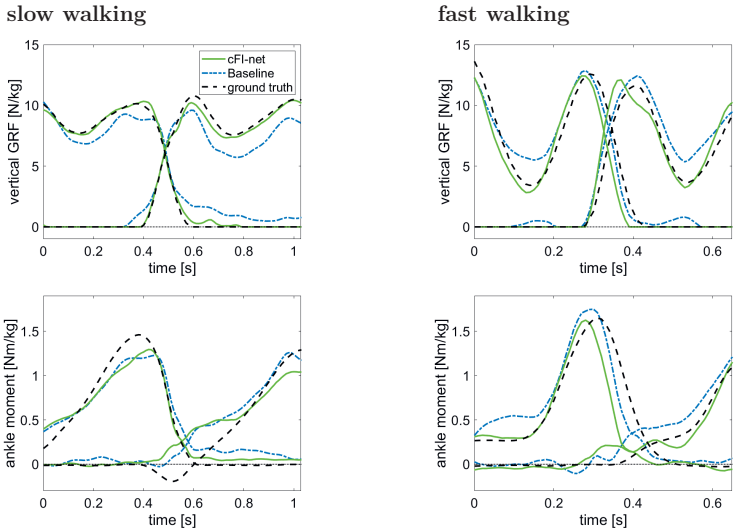


Figure 32: Including target domains by self-supervision. The left side shows example predictions belonging to a slow walking sequence with $\epsilon_u = 13.0\,\%$ and the right side displays the corresponding curves for a fast walking sequence with $\epsilon_u = 19.5\,\%$. The plots compare the performance on the target domain with and without self-supervised training.

of the target domain in a self-supervised manner. In this case, the statistics of the source domain force set are used for normalization. To investigate the effect of the supervised initialization, one test run uses the source domain only for normalization, but not for pre-training. In Table 12 the results of self-supervised learning from scratch, with source domain normalization and with source domain pre-training are compared to each other. In addition, Figure 33 visualizes the self-supervised transfer results of walking, pre-trained on running (top rows) and running, pre-trained on walking (bottom rows).

The quantitative evaluation and the visualization show that the proposed self-supervised model is able to accurately predict GRF. The GRM and consequently the joint torques, on the other hand, are more challenging to learn without explicit target data. Regarding the violation of EOM, the results are close to the corresponding values achieved with supervision. Overall, pre-training and/or normalization using the source domain support the performance of the neural network compared to completely self-supervised learning from scratch.

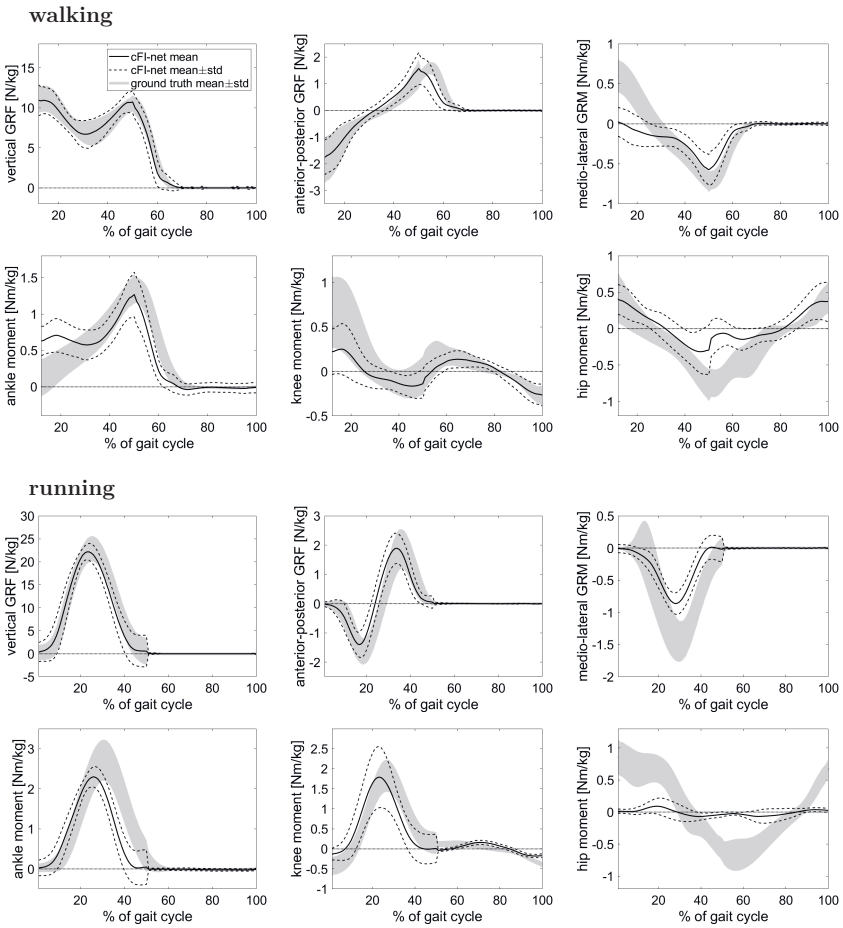


Figure 33: Transfer learning results using cFI-net pre-trained on the source domain and transferred by self-supervised learning on the target domain. The two top rows show controls during walking and the two bottom rows the corresponding controls for running. Grey areas represent the ground truth distribution of the test set and black lines the average regression results with the dashed lines indicating the area of one standard deviation.

Table 12: Domain transfer results for a transfer between walking and running in terms of rRMSE ϵ_{f_r}, ϵ_{m_r}, ϵ_τ and ϵ_u for predicted GRF/M, joint torques and all control parameters combined. The violation of EOM is listed as in terms of e_{EOM}.

motion	scenario	ϵ_{f_r} [%] ↓	ϵ_{m_r} [%] ↓	ϵ_τ [%] ↓	ϵ_u [%] ↓	e_{EOM} $[\frac{Nm}{kg}]$ ↓
gait	from scratch	10.4 (3.4)	27.9 (6.3)	27.1 (4.1)	23.9 (3.8)	0.198 (0.030)
	norm. of run	**9.5** (2.7)	**21.7** (6.3)	23.5 (4.3)	**20.4** (3.6)	**0.151** (0.021)
	transfer	11.6 (3.1)	23.6 (5.8)	**23.4** (3.6)	21.1 (3.2)	0.188 (0.035)
run	from scratch	14.0 (4.4)	34.2 (6.7)	30.6 (3.6)	28.0 (3.2)	0.249 (0.023)
	norm. of walk	13.1 (3.5)	**27.9** (6.9)	28.8 (3.3)	25.5 (3.2)	0.261 (0.027)
	transfer	**12.4** (2.3)	28.4 (6.5)	**26.5** (4.0)	**24.1** (3.5)	**0.229** (0.026)

Reconstructed CMU Sequences

In this section, semi-supervised dynamics learning is implemented using the 3D reconstruction results already introduced in the experimental evaluation of the supervised methods (cf. 5.2.3). The considered CMU gait sequences were reconstructed by a non-rigid structure from motion approach [148]. The goal of this experiment is to investigate whether additional self-supervised training on a subset of the reconstructed movements leads to more realistic predictions and thus helps to bridge the domain gap between structurally different kinematics of the same motion type. For this purpose, the model is trained in a supervised manner on the force set of the laboratory data while 9 of the 18 reconstructed sequences are included into the training and are processed by the forward and the inverse layer of cFI-net. The remaining sequences are used for testing. Since there are no ground truth forces available in this test set, only a qualitative assessment is possible. Although the datasets include the same type of motion (walking) they have been generated in different ways resulting in structural differences. First of all, the 3D reconstruction leads to a lack of global motion, i. e. the root joint of every single pose is aligned. Furthermore, the original CMU data and the self-recorded set were generated using different pre-processing steps and inverse kinematics algorithms which may result in differently biased joint positions and joint angles, respectively.

To make the forward layer applicable to the non-moving root, the global components are multiplied by zero in the calculation of the forward loss. This way, only the angular joint motions of simulation and input are compared. Figure 34 shows mean estimated controls using the baseline network and cFI-net. The ground truth distribution of the PD-set can be seen in the background in grey. This distribution only includes slow walking sequences, since they are more similar to the considered CMU gait patterns than the faster walking motions. However, the distribution shown is not a real *ground truth*, because it does not belong to the test set under consideration. It merely provides an indication of the realism

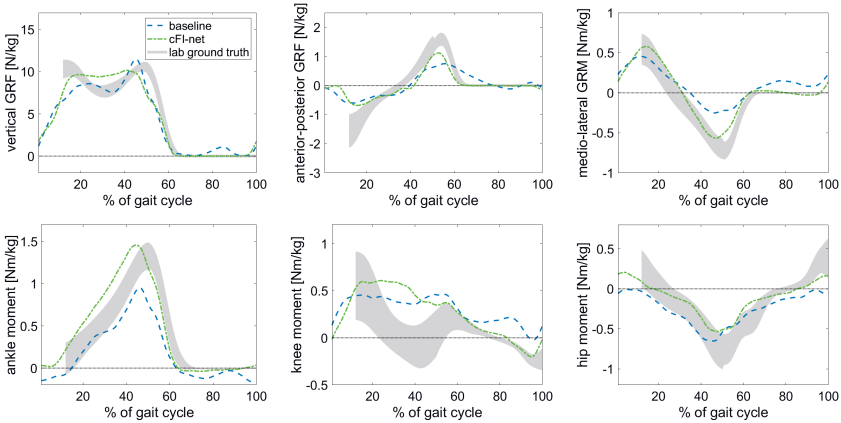


Figure 34: Application to 3D reconstructions [148] of gait sequences from the CMU dataset [18]. The baseline network is only trained on the force set of the self-recorded walking motions. Since the dynamics layers of cFI-net allow self-supervised learning with pure motion samples the model is additionally trained using a subset of the 3D reconstructions.

of the predictions. It can be seen that the predictions by cFI-net are closer to the values of the PD-set, in particular at extreme points, resulting in more realistic shapes.

6.3.4 *Ablation of Input Structure*

This section addresses an ablation study of the input layer to assess the benefit of different design decisions. In particular, the effect of the larger but redundant kinematic input, the polynomial approximation of motion and controls and of the L_1 loss applied to the input layer weights are evaluated. The ablation is performed for the supervised setting without inclusion of the dynamics layers. The following settings are tested:

a) Generalized motion states to controls: $\boldsymbol{u}_{1\dots n} = \boldsymbol{f}_{\text{net}}(\boldsymbol{x}_{1\dots n}, \boldsymbol{l})$.

b) Polynomial approximation: $\boldsymbol{\alpha}_u = \boldsymbol{f}_{\text{net}}(\boldsymbol{\alpha}_x, \boldsymbol{l})$.

c) Motion representation including joint and segment positions: $\boldsymbol{\alpha}_u = \boldsymbol{f}_{\text{net}}(\boldsymbol{\theta})$.

d) Same motion representation as in c) with additional L_1 loss moderating the input layer weights. This setting is used in the baseline method.

Table 13: Comparison of different input and output structures of the baseline network (cf. Section 6.3.4) in terms of rRMSE of GRF, GRM, joint torques and all controls $\boldsymbol{u}$.

motion	setting	ϵ_{f_r} [%] ↓	ϵ_{m_r} [%] ↓	ϵ_τ [%] ↓	ϵ_u [%] ↓
walk	a	10.5 (3.9)	18.7 (5.9)	13.0 (4.1)	13.6 (3.6)
	b	9.4 (3.0)	16.8 (5.1)	12.9 (3.3)	13.0 (3.0)
	c	8.0 (2.0)	**16.1** (4.5)	12.4 (3.0)	12.3 (2.7)
	d	**7.8** (2.0)	**16.1** (4.5)	**12.3** (3.1)	**12.2** (2.6)
run	a	14.0 (4.0)	17.7 (6.7)	14.9 (2.9)	15.3 (3.4)
	b	13.3 (4.6)	18.3 (5.4)	15.3 (2.9)	15.7 (3.2)
	c	12.4 (3.6)	16.1 (5.3)	14.4 (3.4)	14.4 (3.4)
	d	**10.5** (3.6)	**14.2** (4.2)	**13.0** (2.1)	**12.7** (2.0)

The performance of these networks is compared in Table 13. The results clearly show that the network benefits from a wide, information-rich input parameter space, especially when combined with the L_1 loss that causes sparse linkage of the input layer.

6.3.5 *Effect of Noise*

In real world applications, the input motion data can be affected by various error sources, such as measurement noise, detection errors and uncertainties of the lifting approach (if the 3D motions were reconstructed from 2D keypoint detections). Therefore, in order to test the robustness of the proposed models against noisy motion input, zero mean Gaussian noise with increasing standard deviation is added to the generalized coordinates of the motion training set and the test sequences of the walking data. The resulting rRMSE of the estimated controls is depicted in Figure 35. In the case of cFI-net, noisy contact detections are simulated in addition to the coordinate noise. For this purpose, 10 % of the contact labels are randomly chosen and switched with a probability of 50 %. The plot includes results of a network that is trained using the raw motion states and controls as inputs and targets, respectively, without polynomial approximation (*setting a)* of Section 6.3.4). This comparison is done to assess the influence of the polynomials which cause a prior smoothing of the input signal due to the low order.

The experiment shows that the proposed models are very robust to this type of noise, especially when a linear approximation of the input trajectories is used. Furthermore, the dynamics layers can operate on noisy kinematic input without losing their positive influence on the regression results on average.

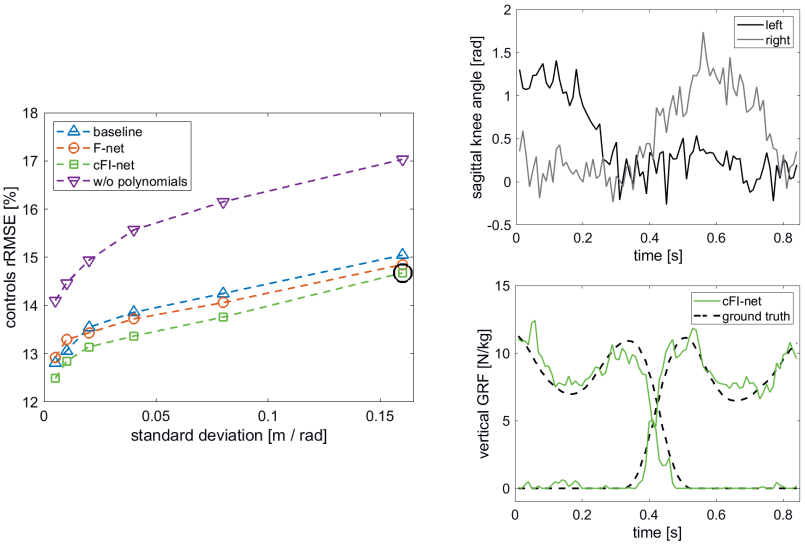


Figure 35: Influence of noisy motion input. The left side shows regression errors ϵ_u for all control parameters given kinematic data in training and test set perturbed by white Gaussian noise with increasing standard deviation. The black circle indicates the measurement point which is exemplified on the right side of the figure by the sagittal knee angle (which is part of the input signal) and the regressed vertical GRF.

6.4 DISCUSSION

In this section the presented methods of supervised and semi-supervised inverse dynamics learning are contrasted and main results as well as opportunities and limitations are discussed. Fully supervised learning was realized by means of different end-to-end regression models and multi-stage methods presented in Chapter 5. For the implementation of both approaches, random forests (RFs), artificial neural networks (NNs) and linear ridge regressions were tested as mappings from motion to control parameters. The multi-stage approach includes a gait phase classification (realized by an RF, an NN or a support vector machine (SVM)) to constrain the parameter space to valid contact states prior to control regression. A further fully supervised method is the baseline network presented in this chapter. In contrast to the end-to-end NN of Chapter 5, the baseline network has a larger input layer that receives all motion representations, i. e. generalized motion states as well as joint and centroid trajectories, and whose weights are penalized with an L_1 loss to encourage sparse linkage. Thus, when training the model a motion input representation, which is optimal for the posed regression task, can be automatically chosen. Of all presented

fully supervised methods, the baseline network yields the best results on both motion types. Compared to the end-to-end NN of Chapter 5, the extended variable input parameter space has led to an improvement of ϵ_u by 7.6 % for walking and considerable 21.1 % for running. However, when comparing to the best multi-stage method, it is important to note that the baseline network requires significantly more hyper parameter tuning to reach the stated performance. For example, the best performing multi-stage approach on the walking dataset, consisting of RF classification and ridge control regression, only involves adjustment of the regularization weight of the ridge regression. The hyper parameters of the RF classification, i. e. number of trees, splitting criterion, minimum number of samples in a leave, have negligible effect on the performance. In contrast to this, the prediction capability of the baseline network is noticeably influenced by parameters such as network depth, width, activation function, optimization algorithm and learning rate. Finally, a fundamental advantage of all presented end-to-end trainable methods is, of course, the less complex training process and shorter computation times for the application.

As addressed earlier and also shown by means of the qualitative evaluation on the reconstructed CMU sequences, the presented supervised methods are limited to motions that are close to the training data and their prediction capability is strongly depending on the number of training examples. To facilitate training on pure motion data and thus increase the usable data pool, self-supervised learning for inverse dynamics was introduced and referred to as Dynamics Network. Here, a distinction is made between a model that uses forward dynamics simulation (F-net) and a model that additionally performs an inverse dynamics calculation including contact states as input (cFI-net). The dynamics procedures are implemented as neural network loss layers. Both methods improve the performance regarding prediction error of GRF/M and joint torques compared to the baseline network (only trained with supervision using an MSE loss) on the walking data. As a further measure for the validity of prediction results, the satisfaction of the EOM are considered. This error measure is substantially improved by the proposed Dynamics Networks.

Overall, the positive effect of the physics-based losses increases with decreasing number of labeled training examples as demonstrated by a corresponding experiment (cf. 6.3.2). While the labeled set is reduced, the size of the unlabeled set that is used by the dynamics layers is kept constant. In general, it can be observed that the error measures describing the closeness to the measured quantities grow with decreasing labeled training set as expected. Considered in detail, the results vary for both forms of movement: For walking, both physics-based models, cFI-net as well as F-net, improve the performance at almost all force set sizes (F-net performs similar to the baseline using the full labeled training set). For running, cFI-net outperforms the other methods using force set fractions of 40 % to 80 % of the total number of subjects. The stronger improvement in the gait dataset

may be due to the fact that many samples of double support that could not be used in the supervised training, due to one missing force plate recording, could be reintroduced with the dynamics layers. Accordingly, the inclusion of the motion set is associated with a greater gain in information than is the case with the running data. In terms of GRF rRMSE, cFI-net achieves excellent results even with very few subjects left in the force set for both motion types.

The GRM are more challenging to learn than the GRF in, both, the supervised and the semi-supervised setting. This can be explained by the comparatively low quality of motion capture results at the foot joints, which are strongly influenced by marker placement: The positions of heel and toe, as endpoints of the kinematic chain, cannot be additionally constrained by determining rotational axes, as is possible for key points at joints. This leads to uncertainties in the fitted 3D motion of the feet and thus the placement of the force application point on the modeled segments. By forming the cross product between the lever arm and the GRF vector, the distance error is amplified and passed on to the GRM. This error propagation leads to a higher uncertainty for the measured GRM than for the GRF, which introduces noise into the training examples and complicates supervised learning of the GRM. Regarding the semi-supervised procedure incorporating the inverse layer a similar drop in performance can be observed between prediction of GRF and GRM. The agreement of the observed motions with Euler's equations for rotational dynamics is affected by the approximation of moments of inertia by simple geometric bodies. This is not the case for linear dynamics described by Newton's equations. The deviation of the GRM from the ground truth is passed on to the joint torques via application of the forward layer. Thus, while GRF estimates supported by the dynamics layers are very robust to a decrease in labeled training data, the related GRM and joint torque predictions deteriorate at a similar rate as the baseline estimates.

It is noteworthy that for running, the multi-stage approach with RF classification and control regression still provides the lowest rRMSE of predicted quantities when the labeled training set is reduced to 20 % and even to 10 % of the total subjects, corresponding to 3 and 1 subjects, respectively. But in terms of compliance with the EOM, incorporating dynamics into the training process leads to significantly improved results for this motion type as well. This outcome is of great interest, since machine learning methods and particularly neural networks are often criticized for their lack of interpretability regarding extracted features and their lack of transparency regarding the regression process [17]. The application of the proposed dynamics loss layers not only improves the model consistency of the predicted quantities, but also enables the quantification of the deviation: The violation of EOM e_{EOM} is in fact a variant of the inverse loss function. Therefore, a low error value in the corresponding criterion is a direct consequence of the training process of cFI-net. The forward loss, on the other hand, is closely related to e_{EOM} but not synonymous with it,

although the same EOM are used. The simulation of motion states involves temporal correlation and error propagation as well as interdependence between GRF/M and joint moments. In contrast, the inverse loss and the error measure e_{eom} consider the satisfaction of the EOM for each time point separately. Temporal correlation is introduced only between three consecutive points by determining the acceleration. In addition, joint torques are ignored in the inverse layer, so the compatibility of GRF/M with input motion is evaluated in isolation. This approach, together with the reduction of the learning rate of the GRF/M output layer during backpropagation of the forward loss leads to a partial decoupling of the optimization of the output parameters, GRF/M and joint torques. In other words it allows that the GRF/M represent the actual response to the observed segment accelerations and that the joint torques are adjusted to produce a motion close to the target motion when applied in a simulation.

Together with the contact loss, the physics-based layers enable training without any forces and torques as ground truth. This capacity can be used to increase the variability of the training set and the generalization capability of the resulting model, as described above, but also to extend the training set with unlabeled samples of a new domain. In this way, domain expansion to different walking speeds was performed. The experiment shows that the Dynamics Network can be adapted to the target domain without significant loss of performance in the source domain.

A related experiment has been conducted to investigate self-supervised domain transfer between locomotion types and self-supervised learning from scratch. In contrast to the previous experiment, the model is fitted to the target domain rather than extended to cover both domains. This is done due to the larger domain gap between walking and running. The model generated by transfer learning can estimate the GRF with excellent accuracy, but the errors of GRM and joint torques increase significantly compared to supervised learning. However, the EOM are well-satisfied by the estimated quantities (low e_{EOM}). Overall, the behavior of the models is similar to the experiment of labeled dataset reduction and can be reasoned in the same way.

The semi-supervised domain expansion to CMU kinematics demonstrates a further use case of the proposed model. The inclusion of pure motion samples from the target set, that can be processed by the dynamics layers, helps to bridge the domain gap between skeleton characteristics. This conclusion can be drawn from a qualitative comparison: The generated curves without domain expansion are considerably less realistic regarding absolute values and curve progressions. It is worth addressing that the flat shape of the estimated vertical GRF is a consequence of the lacking global movement, which is on the one hand a limit of the proposed self-supervision approach, but which shows, on the other hand, that the model is able to capture the characteristics of the tested sequences.

Table 14: Comparison of inverse dynamics methods addressed and proposed in this work. The table includes (from left to right) Predictive dynamics optimization (PDO), data-driven optimization by Lv et al. [85], a bagged random forest (RF), the multi-stage approach (implemented with RF gait phase classification and Ridge control regression), the baseline neural network (Baseline) and the semi-supervised models F-net and cFI-net.

	optimization		**learning**				
	PDO	Lv [85]	RF	Multi-Stage	Baseline	F-net	cFI-net
generation							
self-supervision			-	-	-	+	+
handling small sets			+	-	-	-	+
few hyper param.	-	+	+	+	-	-	-
fast generation	+	+	+	+	+	-	-
application							
w/o GRF/M input	-	+	+	+	+	+	+
w/o global input	-	+	+	+	+	+	-
fast application	-	-	+	+	+	+	+

To summarize the proposed methods of supervised and semi-supervised inverse dynamics learning, Table 14 contrasts all presented approaches listing their conditions, strengths and weaknesses. The comparison is intended to provide a rough classification and recommendation for use. For simplicity, the individual criteria are labeled with + and - indicating *true* and *false* or *strong* and *weak* performance (compared to the average of the included methods), depending on the category. In some cases, a more detailed rating would have been possible, but was omitted for clarity.

Limitations

To conclude this chapter, some limitations of the proposed Dynamics Network will be addressed. One fundamental limitation is that both dynamics layers only constrain the sum of forces and moments, respectively. In combination with the contact loss, single support can be enforced. However, the dynamics layers do not solve the double support ambiguity, i. e. the overdetermination of EOM which is an inherent problem of the equations and can only be resolved by measurement of GRF/M. This limit is not an issue for running, since this motion type does not include double support. In the case of walking, the network converges to the simple solution of modeling a nearly linear progression between the known data points. Here, this behavior leads to good predictions. For more complex motion types, that include long periods of double support, possibly with irregular weight shifting, a

completely self-supervised training of the proposed model will lead to over-smoothing and false equal distribution of GRF.

Furthermore, for computational efficiency, the skeletal model is kept simple with literature values for body segment inertial parameters and average scale factors of the dataset. Therefore, the current implementation of the inverse and the forward layer is only applicable to subjects with average body types. In principle, a corresponding generalization is possible but is not investigated in this work.

Generally, learning-based approaches fail when confronted with data that is too abnormal, i. e. not covered by the training set. This limitation also applies to the proposed learning-based regressions. In principle, however, the self-supervision allows adaptation to deviating data independent from force measurements if the used physical model is still appropriate for the considered motion and double support is frequently interrupted.

Regarding the applicability of the estimated controls in a forward simulation, the usage of the forward loss will decrease the deviation of simulated motion states from the targets. However, since the model does not include a feedback control scheme, the controls cannot be adjusted and thus stable simulations cannot be produced over time. Therefore, a feedback loop could enhance the training process of the proposed model: During the validation of each epoch necessary motion state adjustments could be computed to keep the simulation close to the target motion and the resulting samples could be included in the motion set during the next epoch. This is an interesting topic for future research aiming at automatic balance.

7

CONCLUSIONS

In this work, machine learning models for inverse dynamics of human locomotion are developed and compared. The suitability of the presented learning-based methods for solving the considered task is investigated with respect to predictive dynamics optimization results as a gold standard. To quantify the suitability, deviations of model predictions from this gold standard are considered. This is done not only in a direct sense, but also indirectly by measuring the violation of the equations of motion. Based on the quantitative evaluation it can be concluded that regression methods like random forests and artificial neural networks are able to predict joint torques, contact forces and moments from three dimensional kinematics with satisfactory accuracy. Compared to predictive dynamics optimization, computation times are substantially reduced by two orders of magnitude (from about 3 s to 50 ms per frame).

In developing appropriate regression methods for the inverse dynamics problem, dealing with small datasets has become a focus of this work due to the lack of appropriate large public datasets. Several steps were taken to achieve a reliable regression despite limited data availability. These include, on the one hand, design decisions that allow the use of comparably small regression models: The input and output parameter spaces are kept low-dimensional by using a simplified kinematic model that approximates the upper body with one segment. Moreover, the trajectories of all relevant variables are divided into short temporal windows and linear fits are performed so that the obtained coefficients can be used as input and output signals of the regression. On this basis, relatively shallow fully-connected neural networks, random forests and even linear ridge regression can be applied to estimate forces and moments from motion as described in Chapter 5. On the other hand, the space of usable data is increased to pure motion samples by means of self-supervised learning with differentiable physics-based loss layers which can be incorporated into neural network training. This approach is subject of Chapter 6. Based on these concepts, fully supervised and semi-supervised procedures are implemented and evaluated.

SUPERVISED LEARNING OF INVERSE DYNAMICS

In human locomotion, alternating ground contact results in non-continuous GRF/M curves with values equal to zero during the swing/flight phases. This behaviour is difficult to capture with a regression model and motivated the design of a multi-stage approach that includes a gait phase classification as well as an optional regression of global root

coordinates and calculation of additional manually designed contact features. The method outperforms direct end-to-end regression in terms of prediction quality. It allows the use of a linear control regression which is faster to train, requires less hyperparameter tuning than an artificial neural network and is less prone to overfitting.

Different machine learning models were compared to each other regarding their suitability for the individual tasks of the multi-stage method. The evaluation shows that with moderate hyper-parameter tuning it is advantageous to use linear ridge regression after gait phase classification because this model is not as vulnerable to overfitting as a more powerful, higher-dimensional neural network. Furthermore, on small datasets (like the investigated running set consisting of 66 sequences of 15 subjects) a random forest yields excellent results without having to adjust hyper parameters at all.

The proposed learning based inverse dynamics methods (end-to-end as well as multi-stage approaches) are robust to noisy and incomplete motion representations, which makes them applicable to motion patterns reconstructed from 2D. Such data is characterized, among other things, by the lack of global coordinates and increased noise in the joint trajectories. In the case of the multi-stage approach, an initial estimation of the global coordinates allows for calculation of absolute foot velocities which in turn support the classification of contact states.

SELF-SUPERVISION BY DYNAMICS-BASED LAYERS

The proposed Dynamics Network realizes a complete dynamics cycle consisting of a fully-connected neural network which implements the inverse dynamics step (from kinematics to exterior forces, moments and acting joint torques) and the forward layer that simulates a motion based on the network output. This structure enables minimization of a cyclic loss function depending only on the kinematics. The forward layer solves an initial value problem, given by the equations of motion and the initial motion state, by means of numerical integration. For isolating the optimization of the ground reaction predictions from the estimations of joint torques, an additional inverse layer is proposed which matches exterior forces and moments to the input motion. In combination with a contact loss the model can be trained without any ground truth samples of forces and moments. Only binary labels are still necessary to prevent non-zero exterior forces when there is no contact to the ground. Thus the model combines forward and inverse dynamics loss functions to achieve best possible constraining of the neural network output without needing explicit target forces and moments.

The model maintains stable performance even with very limited labeled training data (consisting of sequences from only a few subjects) by learning generalization ability from a larger, more variable unlabeled motion set. This capability is demonstrated by gradually

reducing the size of the data that includes ground truth forces. With complete reduction, i. e. entirely self-supervised, the ground reaction forces can still be learned accurately. However, the related GRM, determined by the inverse layer, differ from the measured quantities, which is a result of model simplifications and inherent limitations (double support ambiguity). As a consequence, the joint torques, constrained by the forward layer, also deviate from the predictive dynamics results. Without consideration of any measured forces, the training procedure is solely determined by the modelled dynamics which is why control parameters learned with self-supervision match the input motion in terms of the satisfaction of the equations of motion.

In addition to self-supervised learning from scratch, domain expansion and transfer between different walking speeds and motion types (walking and running) is investigated. The model can be extended to slow and fast walking, respectively, without requiring ground truth forces and moments of the new domain. In the case of domain transfer, pre-training as well as normalization using the source domain supports the prediction capability of the neural network trained with self-supervision on the target domain. Furthermore, self-supervised domain expansion to data with differing skeletal characteristics is evaluated qualitatively using the example of 3D reconstructed gait sequences from the CMU database. The experiment shows that the inclusion of motion samples from the target set into the self-supervised training helps to bridge the domain gap between the datasets.

FUTURE WORK

As discussed in Section 6.4, a main limitation of the method that interferes with fully self-supervised learning of GRM and joint torques is the discrepancy between model and reality. Accordingly, possible future steps that directly tie in with the developed model are the usage of a more complex foot model including automatic adjustment of dynamics (i. e. heel strike and toe-off detection and change of the EOM according to the new contact state) and the inclusion of a motion correction framework prior to the self-supervised learning. Correcting the input motions for balance and valid ground contact would reduce motion capture errors that are passed on to the network's internal dynamics processing. Such methods, which also obtain 3D reconstruction from image data, already exist [128]. This idea includes an important step towards fully automated motion analysis which is the combination of inverse dynamics with pose estimation. The presented regression models are designed to handle incomplete and noisy input and to enable fine-tuning without force measurement and thus yield a promising foundation for the realization of vision based dynamics analysis. The implementation of a joint model that simultaneously derives physically plausible 3D trajectories and accurate joint moments from image data remains a goal of future research worth striving for.

With regards to a major limitation introduced by the EOM, the double support ambiguity, it would be interesting to reduce the measured data to this unsolvable subspace, i. e. to record the GRF distribution among the feet and the associated COP for a broad spectrum of motions that include double support. Based on such data a model could be trained specifically for the task of resolving the ambiguity. The derivation of the remaining variables, the total ground reaction force and the joint torques, is covered by the presented physics-based loss layers.

Another potential application worth investigating is the self-supervised fine-tuning of a regression model to subject-specific features. The presented method can be used to fit a pre-trained model to the motion patterns of a specific person. This is interesting, e. g. for the monitoring of rehabilitation or workout progress. In this context, adaptation of the modeled body type should be considered leading to the question whether the body type, i. e. the mass distribution between the body segments, can also be learned from motion patterns.

A

APPENDIX

A.1 EVALUATION BASED ON ADDITIONAL METRICS

This appendix lists additional results belonging to the experiments presented in Section 5.2.1 and Section 6.3.1. The performance of all discussed approaches is evaluated by means of the RMSE and Pearson's correlations coefficient in Table 15. The tables contain the error values of predicted GRF $\boldsymbol{f}_r$, GRM $\boldsymbol{m}_r$ and joint torques $\boldsymbol{\tau}$ compared to the optimization results. The models evaluated were built using the full labeled training datasets and, in the case of the self-supervised approaches F-net and cFI-net, also the unlabeled datasets.

A.2 DATA-DRIVEN INVERSE DYNAMICS OPTIMIZATION

In Chapter 5, the performance of the learning-based inverse dynamics methods is compared to a data-driven maximum a posteriori approach by [85]. For this purpose the referenced method is implemented with a few modifications to allow a fair comparison between the methods. These modifications were made to facilitate the use of the self-recorded data and the presented physical model. The following description is drawn from a previous publication [171].

Opposed to [85], the center of pressure on the foot sole and the torsional torque are modeled using the six dimensional GRM applied to each foot. As a result the state and control parameters become

$$\boldsymbol{z}(t) = (\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \boldsymbol{f}_c(t), \boldsymbol{\tau}(t)) \tag{176}$$

To find $z(t)$ at each frame, the following sum of energy terms is minimized:

$$E(\boldsymbol{z}(t)) = \lambda_1 E_{\text{physical}} + \lambda_2 E_{\text{prior}} + \lambda_3 E_{\text{data}} + \lambda_4 E_{\text{smooth}}\,, \tag{177}$$

with the weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (2, 2, 100, 1)$. In consistence with the used physical model, the friction term E_{friction} is omitted. Instead, friction is captured by the horizontal components of the measured GRF.

Similar to [85], principle component analysis is used to linearize the local parameter space at each frame. The local environment is built of the 200 next neighbours of $\boldsymbol{z}(t)$. Only the scores $\boldsymbol{s}$ of the first n principle components stacked in the matrix $\boldsymbol{K}$ are optimized. These n components constitute 95 % of the overall variability of the local data. This way

Table 15: Performance comparison in terms of RMSE and the correlation coefficient ρ. The models were generated with supervision using the full labeled training sets.

motion	method	RMSE_{f_r} $\left[\frac{N}{kg}\right]$ ↓	RMSE_{m_r} $\left[\frac{Nm}{kg}\right]$ ↓	RMSE_{τ} $\left[\frac{Nm}{kg}\right]$ ↓	ρ_{f_r} ↑	ρ_{m_r} ↑	ρ_{τ} ↑
walk	Lv et al. [85]	0.567 (0.310)	0.077 (0.023)	0.123 (0.039)	0.80 (0.17)	0.73 (0.15)	0.72 (0.22)
	end-to-end	0.295 (0.144)	**0.060** (0.020)	0.095 (0.027)	0.88 (0.11)	0.80 (0.11)	0.81 (0.14)
	multi-stage	**0.188** (0.091)	0.062 (0.016)	**0.084** (0.026)	0.89 (0.11)	0.78 (0.11)	**0.86** (0.11)
	Baseline net	0.251 (0.081)	**0.060** (0.015)	0.088 (0.022)	0.90 (0.11)	**0.83** (0.12)	0.81 (0.14)
	F-net	0.242 (0.086)	0.064 (0.016)	0.085 (0.022)	**0.91** (0.10)	0.79 (0.12)	0.84 (0.12)
	cFI-net	0.210 (0.068)	0.061 (0.014)	0.085 (0.022)	**0.91** (0.11)	0.80 (0.13)	0.82 (0.13)
run	Lv et al. [85]	1.094 (0.342)	0.116 (0.045)	0.174 (0.036)	0.67 (0.15)	0.57 (0.24)	0.73 (0.10)
	end-to-end	0.593 (0.215)	0.083 (0.038)	0.130 (0.037)	0.78 (0.14)	0.75 (0.16)	0.82 (0.08)
	multi-stage	0.572 (0.198)	0.085 (0.044)	0.127 (0.034)	0.75 (0.14)	0.73 (0.16)	**0.84** (0.06)
	Baseline net	**0.553** (0.187)	**0.076** (0.027)	**0.117** (0.022)	**0.91** (0.05)	0.71 (0.19)	0.82 (0.07)
	F-net	0.699 (0.275)	0.104 (0.042)	0.129 (0.035)	0.89 (0.09)	0.75 (0.14)	0.82 (0.08)
	cFI-net	0.599 (0.219)	0.096 (0.042)	0.125 (0.033)	**0.91** (0.08)	**0.77** (0.13)	0.82 (0.08)

the number of optimization variables is drastically reduced. The optimization problem becomes

$$\min_{\boldsymbol{s}}\{E(\boldsymbol{\mu}+\boldsymbol{K}\boldsymbol{s}))\}\,, \tag{178}$$

with the mean $\boldsymbol{\mu}$ of the neighbouring parameter vectors.

As indicated before, $\boldsymbol{z}$ is adapted to fit the physical model, which has an immediate effect on the physical term E_{phys}. This term describes the deviation of the kinematic state $(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t))$ given by $\boldsymbol{z}(t)$ from the kinematics arising from the acting forces and torques via the EOM. Similar to the definition used in the predictive dynamics optimization presented in Eq. (144) the energy term is

$$E_{\text{physical}} = \|\boldsymbol{\mathcal{M}}\ddot{\boldsymbol{q}} - \boldsymbol{\mathcal{F}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{f}_c, \boldsymbol{\tau})\|_2^2\,. \tag{179}$$

The changed physical model further necessitates a slight modification of the smoothness term:

$$E_{\text{smooth}} = \|\boldsymbol{f}_c(t-1) - 2\boldsymbol{f}_c(t) + \boldsymbol{f}_c(t+1)\|_2^2\,. \tag{180}$$

The remaining energy terms E_{prior} and E_{data} can be employed without adaptation.

BIBLIOGRAPHY

[1] Sivan Almosnino, David Kingston, and Ryan B. Graham. “Three-Dimensional Knee Joint Moments During Performance of the Bodyweight Squat: Effects of Stance Width and Foot Rotation.” In: *Journal of Applied Biomechanics* 29.1 (2013), 33–43.

[2] Jorge Angeles. *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms (Mechanical Engineering Series)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[3] Marzieh Ardestani, Xuan Zhang, Ling Wang, Qin Lian, Yaxiong Liu, Jiankang He, Dichen Li, and Zhongmin Jin. “Human lower extremity joint moment prediction: A wavelet neural network approach.” In: *Expert Syst. Appl.* 41 (2014), pp. 4422–4433.

[4] Rezaul Begg and Joarder Kamruzzaman. “A machine learning approach for automated recognition of movement patterns using basic, kinetic and kinematic gait data.” In: *Journal of Biomechanics* 38.3 (2005), pp. 401–408.

[5] Rezaul Begg and Joarder Kamruzzaman. “Neural networks for detection and classification of walking pattern changes due to ageing.” In: *Australasian Physics Engineering Sciences in Medicine* 29.2 (2006).

[6] Filipe de A. Belbute-Peres, Kevin A. Smith, Kelsey R. Allen, Joshua B. Tenenbaum, and J. Zico Kolter. “End-to-End Differentiable Physics for Learning and Control.” In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2018, pp. 7178–7189.

[7] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. “LoopReg: Self-supervised Learning of Implicit Surface Correspondences, Pose and Shape for 3D Human Mesh Registration.” In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] Léon Bottou. “Large-scale machine learning with stochastic gradient descent.” In: *in COMPSTAT*. 2010.

[10] Biagio Brattoli, Uta Büchler, Anna-Sophia Wahl, Martin E. Schwab, and Björn Ommer. “LSTM Self-Supervision for Detailed Behavior Analysis.” In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3747–3756.

[11] Leo Breiman. “Bagging Predictors.” In: *Machine Learning* 24.2 (1996), pp. 123–140.

[12] Leo Breiman. “Random Forests.” In: *Machine Learning* 45.1 (2001), pp. 5–32.

[13] Leo Breiman, Jerome H. Friedman, R. A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[14] Charles G. Broyden. “The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations.” In: *IMA Journal of Applied Mathematics* 6.1 (1970), pp. 76–90.

[15] Marcus A. Brubaker and David J. Fleet. “The Kneed Walker for human pose tracking.” In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 2008, pp. 1–8.

[16] Marcus A. Brubaker, Leonid Sigal, and David J. Fleet. “Estimating contact dynamics.” In: *Computer Vision, 2009 IEEE 12th International Conference on*. 2009, pp. 2389–2396.

[17] Jenna Burrell. “How the machine ‘thinks’: Understanding opacity in machine learning algorithms.” In: *Big Data & Society* 3.1 (2016).

[18] CMU. *Human motion capture database*. 2014. URL: `http://mocap.cs.cmu.edu/` (visited on 06/04/2020).

[19] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. “Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[20] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. “Neural Ordinary Differential Equations.” In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[21] Xipeng Chen, Kwan-Yee Lin, Wentao Liu, Chen Qian, and Liang Lin. “Weakly-Supervised Discovery of Geometry-Aware Representation for 3D Human Pose Estimation.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[22] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. “HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[23] Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. “Physics-Based Motion Capture Imitation with Deep Reinforcement Learning.” In: *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*. MIG ’18. Association for Computing Machinery, 2018.

[24] Jonas Degrave, Michiel Hermans, Joni Dambre, and Francis wyffels. “A Differentiable Physics Engine for Deep Learning in Robotics.” In: *Frontiers in Neurorobotics* 13 (2019), p. 6.

[25] Scott L. Delp, Frank C. Anderson, Allison S. Arnold, Peter Loan, Ayman Habib, Chand T. John, Eran Guendelman, and Darryl G. Thelen. “OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement.” In: *IEEE Transactions on Biomedical Engineering* 54.11 (2007), pp. 1940–1950.

[26] Jacques Denavit and Richard Scheunemann Hartenberg. “Kinematic synthesis of linkages.” In: *McGraw-Hill series in mechanical engineering. New York: McGraw-Hill* (1965), p. 435.

[27] Eric Desailly, Yepremian Daniel, Philippe Sardain, and Patrick Lacouture. “Foot contact event detection using kinematic data in cerebral palsy children and normal adults gait.” In: *Gait & Posture* 29.1 (2009), pp. 76–80.

[28] Carl Doersch and Andrew Zisserman. “Sim2real transfer learning for 3D human pose estimation: motion to the rescue.” In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[29] Rudolfs Drillis, Renato Contini, and Maurice Bluestein. “Body Segment Parameters; a Survey of Measurement Techniques.” In: *Artificial limbs* 8 (1964), 44—66.

[30] Mahdokht Ezati, Peter Brown, Borna Ghannadi, and John McPhee. “Comparison of Direct Collocation Optimal Control to Trajectory Optimization for Parameter Identification of an Ellipsoidal Foot–Ground Contact Model.” In: *Multibody System Dynamics* (2020).

[31] Mahdokht Ezati, Borna Ghannadi, and John McPhee. “A Review of Simulation Methods for Human Movement Dynamics with Emphasis on Gait.” In: *Multibody System Dynamics* (2019).

[32] Herre Faber, Arthur J. van Soest, and Dinant A. Kistemaker. “Inverse dynamics of mechanical multibody systems: An improved algorithm that ensures consistency between kinematics and external forces.” In: *PLOS ONE* 13.9 (Sept. 2018), pp. 1–16.

[33] Roy Featherstone. *Rigid Body Dynamics Algorithms*. New York, NY, USA: Springer Science+Business Media, LLC, 2008.

[34] Martin L. Felis, Katja Mombaur, and Alain Berthoz. “An optimal control approach to reconstruct human gait dynamics from kinematic data.” In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. 2015, pp. 1044–1051.

[35] Florian Fischer, Miroslav Bachinski, Markus Klar, Arthur Fleig, and Jörg Müller. “Reinforcement learning control of a biomechanical model of the upper extremity.” In: *Scientific Reports* 11.1 (2021).

[36] Roger Fletcher. *Practical Methods of Optimization*. Second. New York, NY, USA: John Wiley & Sons, 1987.

[37] Arturo Forner-Cordero, H.J.F.M. Koopman, and F.C.T. van der Helm. "Use of pressure insoles to calculate the complete ground reaction forces." In: *Journal of Biomechanics* 37.9 (2004), pp. 1427–1432.

[38] Arturo Forner-Cordero, H.J.F.M. Koopman, and F.C.T. van der Helm. "Inverse dynamics calculations during gait with restricted ground reaction force information from pressure insoles." In: *Gait & Posture* 23.2 (2006), pp. 189–199.

[39] Claudiane A. Fukuchi, Reginaldo K. Fukuchi, and Marcos Duarte. "A public dataset of overground and treadmill walking kinematics and kinetics in healthy individuals." In: *PeerJ* 6 (2018).

[40] Reginaldo K Fukuchi, Bjoern M Eskofier, Marcos Duarte, and Reed Ferber. "Support vector machines for detecting age-related changes in running kinematics." In: *Journal of Biomechanics* 44.3 (2010), pp. 540–2.

[41] James R. Gage and Tom F. Novacheck. "An update on the treatment of gait problems in cerebral palsy." In: *Journal of Pediatric Orthopaedics B* 10.4 (2001), 265—274.

[42] Sean Gallagher, Christopher A. Hamrick, Kim M. Cornelius, and Mark S. Redfern. "The effects of restricted workspace on lumbar spine loading." In: *Occupational Ergonomics* 2.4 (2001), pp. 201–213.

[43] Yaroslav Ganin, E. Ustinova, Hana Ajakan, Pascal Germain, H. Larochelle, François Laviolette, M. Marchand, and V. Lempitsky. "Domain-Adversarial Training of Neural Networks." In: *J. Mach. Learn. Res.* 17 (2016), 59:1–59:35.

[44] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Perez, and Matthieu Cord. "Boosting Few-Shot Visual Learning With Self-Supervision." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[45] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*. 2010.

[46] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks." In: vol. 15. Proceedings of Machine Learning Research. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[47] *Go Further with Vicon MX T-Series Revision 1.3 August 2010*. `https://docs.vicon.com/display/LegacyCamDoc?preview=/71237793/71237799/T-Series_GoFurther_Rev1.3_2010Aug.pdf`.

[48] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[49] Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Second. USA: Society for Industrial and Applied Mathematics, 2008.

[50] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. "DensePose: Dense Human Pose Estimation in the Wild." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[51] Arthur E. Haas and Terence Verschoyle. *Introduction to theoretical physics*. Constable company ltd, 1928.

[52] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. "DeepCap: Monocular Human Performance Capture Using Weak Supervision." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[53] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. "DeepCap: Monocular Human Performance Capture Using Weak Supervision." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[54] Layla Hashem, Roaa Al-Harakeh, and Ali Cherry. "Human Gait Identification System Based on Transfer Learning." In: *2020 21st International Arab Conference on Information Technology (ACIT)*. 2020, pp. 1–6.

[55] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[56] Milton Roberto Heinen and Fernando Santos Osório. "Gait Control Generation for Physically Based Simulated Robots Using Genetic Algorithms." In: *Advances in Artificial Intelligence - IBERAMIA-SBIA 2006*. Springer Berlin Heidelberg, 2006, pp. 562–571.

[57] Arthur E. Hoerl and Robert W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." In: *Technometrics* 12.1 (1970), pp. 55–67.

[58] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. cite arxiv:1704.04861. 2017. URL: `http://arxiv.org/abs/1704.04861`.

[59] Leif Johnson and Dana H. Ballard. “Efficient Codes for Inverse Dynamics During Walking.” In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 2014, pp. 343–349.

[60] William Johnson, Jacqueline Alderson, David Lloyd, and Ajmal Mian. “Predicting Athlete Ground Reaction Forces and Moments From Spatio-Temporal Driven CNN Models.” In: *IEEE Transactions on Biomedical Engineering* PP (2018), pp. 1–1.

[61] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. “Learning 3D Human Dynamics From Video.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[62] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” In: *3rd International Conference on Learning Representations (ICLR)*. 2015.

[63] Kristof Kipp. “Relative importance of lower extremity net joint moments in relation to bar velocity and acceleration in weightlifting.” In: *Sports Biomechanics* 0.0 (2020), pp. 1–13.

[64] Kristof Kipp, Matthew Giordanelli, and Christopher Geiser. “Predicting net joint moments during a weightlifting exercise with a neural network model.” In: *Journal of Biomechanics* 74 (2018), pp. 225–229.

[65] Reinhard Klette and Garry Tee. “Understanding Human Motion: A Historic Review.” In: *Human Motion: Understanding, Modelling, Capture, and Animation*. Dordrecht: Springer Netherlands, 2008, pp. 1–22.

[66] Bart Koopman, Henk J. Grootenboer, and Henk J. de Jongh. “An inverse dynamics model for the analysis, reconstruction and prediction of bipedal walking.” In: *Journal of Biomechanics* 28.11 (1995), pp. 1369–1376.

[67] Basil Kouvaritakis and Mark Cannon. *Model Predictive Control: Classical, Robust and Stochastic*. Advanced Textbooks in Control and Signal Processing. Springer International Publishing, 2015.

[68] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. “PifPaf: Composite Fields for Human Pose Estimation.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks.” In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105.

[70] Arthur D. Kuo. “A Least-Squares Estimation Approach to Improving the Precision of Inverse Dynamics Computations.” In: *Journal of Biomechanical Engineering* 120.1 (1998), pp. 148–159.

[71] Yuri Kwon, Ji-Won Kim, Jae-Hoon Heo, Hyeong-Min Jeon, Eui-Bum Choi, and Gwang-Moon Eom. “The effect of sitting posture on the loads at cervico-thoracic and lumbosacral joints.” In: *Technology and Health Care* 26.S1 (2018), pp. 409–418.

[72] Stefan Lambrecht, Anna Harutyunyan, Kevin Tanghe, Maarten Afschrift, Joris De Schutter, and Ilse Jonkers. “Real-Time Gait Event Detection Based on Kinematic Data Coupled to a Biomechanical Model †.” In: *Sensors* 17.4 (2017).

[73] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. “Scalable Muscle-Actuated Human Simulation and Control.” In: 38.4 (2019).

[74] Wonyeol Lee, Hangyeol Yu, Xavier Rival, and Hongseok Yang. *On Correctness of Automatic Differentiation for Non-Differentiable Functions*. 2020. arXiv: 2006.06903 [cs.LG].

[75] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, and Mingwu Ren. “Gait Recognition via Semi-supervised Disentangled Representation Learning to Identity and Covariate Features.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[76] Xinwei Li, Su Liu, Ying Chang, Sujiao Li, Yuanjie Fan, and Hongliu Yu. “A Human Joint Torque Estimation Method for Elbow Exoskeleton Control.” In: *International Journal of Humanoid Robotics* 17.03 (2020), p. 1950039.

[77] Yanan Li, Yilong Yin, Lili Liu, Shaohua Pang, and Qiuhong Yu. “Semi-supervised Gait Recognition Based on Self-Training.” In: *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*. 2012, pp. 288–293.

[78] Hyerim Lim, Bumjoon Kim, and Sukyung Park. “Prediction of Lower Limb Kinetics and Kinematics during Walking by a Single IMU on the Lower Back Using Machine Learning.” In: *Sensors* 20.1 (2020).

[79] Yi-Chung Lin, Jonathan P Walter, and Marcus G Pandy. “Predictive Simulations of Neuromuscular Coordination and Joint-Contact Loading in Human Gait.” In: *Annals of biomedical engineering* 46 (2018), pp. 1216–1227.

[80] Marius Lindauer and Frank Hutter. “Best Practices for Scientific Research on Neural Architecture Search.” In: *Journal of Machine Learning Research* 21 (2020), pp. 1–18.

[81] Nan Liu, Liangyu Li, Bing Hao, Liusong Yang, Tonghai Hu, Tao Xue, Shoujun Wang, and Xingmao Shao. “Semiparametric Deep Learning Manipulator Inverse Dynamics Modeling Method for Smart City and Industrial Applications.” In: *Complexity* (2020), p. 11.

[82] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. "Attention Mechanism Exploits Temporal Contexts: Real-Time 3D Human Pose Reconstruction." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[83] Yu Liu, Shi-Min Shih, Shi-Liu Tian, Yun-Jian Zhong, and Li Li. "Lower extremity joint torque predicted by using artificial neural network during vertical jump." In: *Journal of biomechanics* 42.7 (2009), 906—911.

[84] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. "SMPL: A Skinned Multi-Person Linear Model." In: *ACM Trans. Graph.* 34.6 (Oct. 2015).

[85] Xiaolei Lv, Jinxiang Chai, and Shihong Xia. "Data-driven Inverse Dynamics for Human Motion." In: *ACM Trans. Graph.* 35.6 (2016), 163:1–163:12.

[86] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models." In: *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[87] Christian Maiwald, T. Sterzing, T.A. Mayer, and T.L. Milani. "Detecting foot-to-ground contact from kinematic data in running." In: *Footwear Science* 1.2 (2009), pp. 111–118.

[88] Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. "Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs." In: *Computer Graphics Forum 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics)* (2017).

[89] Angel Martínez-González, Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. "Investigating Depth Domain Adaptation for Efficient Human Pose Estimation." In: *2018 European Conference on Computer Vision - Workshops, ECCVW*. 2018.

[90] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. "A simple yet effective baseline for 3d human pose estimation." In: *Proceedings IEEE International Conference on Computer Vision (ICCV)*. 2017.

[91] Warren Mcculloch and Walter Pitts. "A Logical Calculus of Ideas Immanent in Nervous Activity." In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 127–147.

[92] Ali Meghdari, Alaa Abdulrahman, Kamran Iqbal, and Gannon White. "Improving Inverse Dynamics Accuracy in a Planar Walking Model Based on Stable Reference Point." In: *Journal of Robotics* (2014).

[93] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. "XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera." In: *ACM Transactions on Graphics, (Proc. SIGGRAPH)* 39.4 (2020).

[94] Richard H. Middletone and Graham C. Goodwin. "Adaptive computed torque control for rigid link manipulators." In: *1986 25th IEEE Conference on Decision and Control*. 1986, pp. 68–73.

[95] Rick Miranda. *Algebraic Curves and Riemann Surfaces*. Dimacs Series in Discrete Mathematics and Theoretical Comput. American Mathematical Society, 1995.

[96] Rahul Mitra, Nitesh B. Gundavarapu, Abhishek Sharma, and Arjun Jain. "Multiview-Consistent Semi-Supervised Learning for 3D Human Pose Estimation." In: *CVPR*. IEEE, 2020, pp. 6906–6915.

[97] Francisco Mouzo, Urbano Lugris, Rosa Pamies-Vila, and Javier Cuadrado. "Skeletal-level control-based forward dynamic analysis of acquired healthy and assisted gait motion." In: *Multibody System Dynamics* 44.1 (2008), 1–29.

[98] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013.

[99] Natalia Neverova, Christian Wolf, Griffin Lacey, Lex Fridman, Deepak Chandra, Brandon Barbello, and Graham Taylor. "Learning Human Identity From Motion Patterns." In: *IEEE Access* 4 (2016), pp. 1810–1820.

[100] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation." In: *Computer Vision - 14th European Conference, ECCV 2016, Proceedings*. Springer Verlag, 2016, pp. 483–499.

[101] Marlies Nitschke, Eva Dorschky, Dieter Heinrich, Heiko Schlarb, Bjoern M. Eskofier, Anne D. Koelewijn, and Antonie J. van den Bogert. "Efficient trajectory optimization for curved running using a 3D musculoskeletal model with implicit dynamics." In: *Scientific Reports* 10.17655 (2020).

[102] Bhargava T. Nukala, Naohiro Shibuya, Amanda Rodriguez, Jerry Tsay, Jerry Lopez, Tam Nguyen, Steven Zupancic, and Donald Y. Lie. "An Efficient and Robust Fall Detection System Using Wireless Gait Analysis Sensor with Artificial Neural Network (ANN) and Support Vector Machine (SVM) Algorithms." In: *Open Journal of Applied Biosensor* 3.4 (2014).

[103] Seung Eel Oh, Ahnryul Choi, and Joung Hwan Mun. "Prediction of ground reaction forces during gait based on kinematics and a neural network model." In: *Journal of Biomechanics* 46.14 (2013), pp. 2372 –2380.

[104] Yassine Ouali, C. Hudelot, and Myriam Tami. “An Overview of Deep Semi-Supervised Learning.” In: *ArXiv* abs/2006.05278 (2020).

[105] Ciara M. O’Connor, Susannah K. Thorpe, Mark J. O’Malley, and Christopher L. Vaughan. “Automatic detection of gait events using kinematic data.” In: *Gait & Posture* 25.3 (2007), pp. 469–474.

[106] Sinno J. Pan and Qiang Yang. “A Survey on Transfer Learning.” In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359.

[107] Adina M. Panchea, Sylvain Miossec, Olivier Buttelli, Philippe Fraisse, Angèle Van Hamme, Marie-Laure Welter, and Nacim Ramdani. “Gait analysis using optimality criteria imputed from human data.” In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 13510–13515.

[108] Prashant Pandey, Prathosh AP, Manu Kohli, and Josh Pritchard. “Guided Weak Supervision for Action Recognition with Scarce Data to Assess Skills of Children with Autism.” In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01 (2020), pp. 463–470.

[109] Hwayoung Park, Changhong Youm, Minji Son, Meounggon Lee, and Jinhee Kim. “Effects of Freezing of Gait on Spatiotemporal Variables, Ground Reaction Forces, and Joint Moments during Sit-to-walk Task in Parkinson’s Disease.” In: *Korean Journal of Sport Biomechanics* 28.1 (2018), pp. 19–27.

[110] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. “DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills.” In: *ACM Trans. Graph.* 37.4 (2018).

[111] Xue Bin Peng, Glen Berseth, Kangkang Yin, and Michiel Van De Panne. “DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning.” In: *ACM Trans. Graph.* 36.4 (2017).

[112] Xue Bin Peng and Michiel van de Panne. “Learning Locomotion Skills Using DeepRL: Does the Choice of Action Space Matter?” In: *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. SCA ’17. Association for Computing Machinery, 2017.

[113] Lerrel Pinto and Abhinav Gupta. “Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours.” In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 3406–3413.

[114] Dejan B. Popovic and Mirjana B. Popovic. “Design of a Control for a Neural Prosthesis for Walking: Use of Artificial Neural Networks.” In: *2006 8th Seminar on Neural Network Applications in Electrical Engineering*. 2006, pp. 121–128.

[115] Stephen D. Prentice, Aftab E. Patla, and Deborah A. Stacey. “Artificial neural network model for the generation of muscle activation patterns for human locomotion.” In: *Journal of electromyography and kinesiology* 11.1 (2001), pp. 19–30.

[116] Santiago Pujol, Madeline D. Nelson, and J. Paul Smith-Pardo. “Virtual velocity: Connecting the concepts of Kinematics and Virtual Work.” In: *Engineering Structures* 56 (2013), pp. 2249–2252.

[117] Tian Qi, Yinfu Feng, Jun Xiao, Hanzhi Zhang, Yueting Zhuang, Xiaosong Yang, and Jianjun Zhang. “A Human Motion Feature Based on Semi-Supervised Learning of GMM.” In: 23.1 (2017), 85–93.

[118] Manish Raj, Akhilesh Kumar Singh, and Vivek Sharma. “Joint Angle Torque Generation Based on Machine Learning Approaches for Humanoid Locomotion.” In: *International Journal of Advanced Science and Technology* 29.3 (2020), pp. 13509–.

[119] C. David Remy and Darryl G. Thelen. “Optimal estimation of dynamically consistent kinematics and kinetics for forward dynamic simulation of gait.” In: *Journal of Biomechanical Engineering* 131.3 (2009).

[120] Ricardo Riaza. *Differential-Algebraic Systems: Analytical Aspects and Circuit Applications*. World Scientific, 2008.

[121] Raziel Riemer, Elizabeth T. Hsiao-Wecksler, and Xudong Zhang. “Uncertainties in inverse dynamics solutions: A comprehensive analysis and an application to gait.” In: *Gait & Posture* 27.4 (2008), pp. 578–588.

[122] Laura von Rueden, Sebastian Mayer, Rafet Sifa, Christian Bauckhage, and Jochen Garcke. “Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions.” In: *Advances in Intelligent Data Analysis XVIII*. Cham: Springer International Publishing, 2020, pp. 548–560.

[123] Isabel C.N. Sacco, Andreja P. Picon, Diego O. Macedo, Marcos K. Butugan, Ricky Watari, and Cristina D. Sartor. “Alterations in the lower limb joint moments precede the peripheral neuropathy diagnosis in diabetes patients.” In: *Diabetes Technol Ther.* 17.6 (2015), pp. 405–412.

[124] Connor Schenck and Dieter Fox. “SPNets: Differentiable Fluid Dynamics for Deep Neural Networks.” In: *Proceedings of The 2nd Conference on Robot Learning*. Vol. 87. Proceedings of Machine Learning Research. PMLR, 2018, pp. 317–335.

[125] Arend L. Schwab and Guido M. J. Delhaes. *Lecture Notes Multibody Dynamics B, wb1413*. Accessed: 11-24-2020. 2002. URL: `http://bicycle.tudelft.nl/schwab/Publications/LinSch02.pdf`.

[126] Terrence J. Sejnowski. “The unreasonable effectiveness of deep learning in artificial intelligence.” In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30033–30038.

[127] Sharif Shourijeh, Mohammad and McPhee, John. “Foot–ground contact modeling within human gait simulations: from Kelvin–Voigt to hyper-volumetric models.” In: (2015).

[128] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. “PhysCap: Physically Plausible Monocular 3D Motion Capture in Real Time.” In: *ACM Transactions on Graphics* 39.6 (2020).

[129] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” In: *CoRR* abs/1409.1556 (2014).

[130] Aditya Singh and G. C. Nandi. “Machine Learning based Joint Torque calculations of Industrial Robots.” In: *2018 Conference on Information and Communication Technology (CICT)*. 2018, pp. 1–6.

[131] Ghanapriya Singh, Mahesh Chowdhary, Arun Kumar, and Rajendar Bahl. “A Personalized Classifier for Human Motion Activities With Semi-Supervised Learning.” In: *IEEE Transactions on Consumer Electronics* 66.4 (2020), pp. 346–355.

[132] Yaroslav Smirnov, Denis Smirnov, Anton Popov, and Sergiy Yakovenko. “Solving musculoskeletal biomechanics with machine learning.” In: *bioRxiv* (2020).

[133] Seungmoon Song, Łukasz Kidziński, Xue Bin Peng, Carmichael Ong, Jennifer Hicks, Sergey Levine, Christopher G. Atkeson, and Scott L. Delp. “Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation.” In: *Journal of NeuroEngineering and Rehabilitation* 18.1 (2021).

[134] Mark .W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Hoboken, NJ, USA: Wiley, 2005.

[135] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[136] Bernd J. Stetter, Frieder C. Krafft, Steffen Ringhof, Thorsten Stein, and Stefan Sell. “A Machine Learning and Wearable Sensor Based Approach to Estimate External Knee Flexion and Adduction Moments During Various Locomotion Tasks.” In: *Frontiers in Bioengineering and Biotechnology* 8 (2020), p. 9.

[137] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. “Integral Human Pose Regression.” In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[138] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going Deeper with Convolutions." In: *Computer Vision and Pattern Recognition (CVPR)*. 2015.

[139] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. cite arxiv:1905.11946Comment: Published in ICML 2019. 2019. URL: http://arxiv.org/abs/1905.11946.

[140] Darryl G. Thelen and Frank C. Anderson. "Using computed muscle control to generate forward dynamic simulations of human walking from experimental data." In: *Journal of Biomechanics* 39.6 (2006), pp. 1107–1115.

[141] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition*. 3rd. San Diego, CA, USA: Academic Press, An imprint of Elsevier, 2006.

[142] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. "Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms." In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '13. Association for Computing Machinery, 2013, 847–855.

[143] Marc Toussaint, Kelsey R. Allen, Kevin A. Smith, and Joshua B. Tenenbaum. "Differentiable Physics and Stable Modes for Tool-Use and Manipulation Planning –Extended Abstract." In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. International Joint Conferences on Artificial Intelligence, 2019, pp. 6231–6235.

[144] Brian R. Umberger and Ross H. Miller. "Optimal Control Modeling of Human Movement." In: *Handbook of Human Motion*. Cham: Springer International Publishing, 2017, pp. 1–22.

[145] Tanmay Tulsidas Verlekar, Paulo Lobato Correia, and Luís Ducla Soares. "Using transfer learning for classification of gait pathologies." In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2018, pp. 2376–2381.

[146] Anh Ta Vinay Prabhu Stephanie Tietz. *Classifying humans using Deep time-series transfer learning : accelerometric gait-cycles to gyroscopic squats*. 2019.

[147] Housner G. W. *Applied mechanics / by George W. Housner and Donald E. Hudson*. eng. 2d ed. Princeton, N.J: Van Nostrand, 1961.

[148] Bastian Wandt, Hanno Ackermann, and Bodo Rosenhahn. "3D Reconstruction of Human Motion from Monocular Image Sequences." In: *Transactions on Pattern Analysis and Machine Intelligence* 38.8 (2016), pp. 1505–1516.

[149] Bastian Wandt and Bodo Rosenhahn. “RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[150] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin, and Bodo Rosenhahn. “CanonPose: Self-Supervised Monocular 3D Human Pose Estimation in the Wild.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 13294–13304.

[151] Kun Wang, Mridul Aanjaneya, and Kostas Bekris. “A First Principles Approach for Data-Efficient System Identification of Spring-Rod Systems via Differentiable Physics Engines.” In: *Proceedings of the 2nd Conference on Learning for Dynamics and Control*. Vol. 120. Proceedings of Machine Learning Research. PMLR, 2020, pp. 651–665.

[152] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. “UntrimmedNets for Weakly Supervised Action Recognition and Detection.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[153] Paweł Wawrzyński. “Autonomous Reinforcement Learning with Experience Replay for Humanoid Gait Optimization.” In: *Procedia Computer Science* 13 (2012), pp. 205–211.

[154] Wen We, Katherine R. Saul, and He Huang. “Using Reinforcement Learning to Estimate Human Joint Moments From Electromyography or Joint Kinematics: An Alternative Solution to Musculoskeletal-Based Biomechanics.” In: *J Biomech Eng* 143.4 (2021), p. 044502.

[155] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. “A survey of transfer learning.” In: *Journal of Big data* 3.1 (2016), p. 9.

[156] S.E. Williams, Sheila Gibbs, C.B. Meadows, and Rami Abboud. “Classification of the reduced vertical component of the ground reaction force in late stance in cerebral palsy gait.” In: *Gait & Posture* 34.3 (2011), pp. 370–373.

[157] David A. Winter. *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons, Ltd, 2009.

[158] Jianning Wu, Bin Wu, and Kwang Gi Kim. “The Novel Quantitative Technique for Assessment of Gait Symmetry Using Advanced Statistical Learning Algorithm.” In: *BioMed Research International* (2015).

[159] J. Wühr, Ursula Veltmann, L. Linkemeyer, Burkhard Drerup, and Hans H. Wetz. “Influence of Modern Above-Knee Prostheses on the Biomechanics of Gait.” In: *Advances in Medical Engineering*. Springer Berlin Heidelberg, 2007, pp. 267–272.

[160] Yujiang Xiang, Jasbir S. Arora, Salam Rahmatalla, and Karim Abdel-Malek. "Physics-based modeling and simulation of human walking: a review of optimization-based and other approaches." In: *Structural and Multidisciplinary Optimization* 42.1 (2010), pp. 1–23.

[161] Yujiang Xiang, Hyun-Joon Chung, Joo H. Kim, Rajankumar Bhatt, Salam Rahmatalla, Jingzhou Yang, Timothy Marler, Jasbir S. Arora, and Karim Abdel-Malek. "Predictive dynamics: an optimization-based novel approach for human motion simulation." In: *Structural and Multidisciplinary Optimization* 41.3 (2010), pp. 465–479.

[162] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang, and Wenjun Zhang. "Deep Kinematics Analysis for Monocular 3D Human Pose Estimation." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[163] Cheng Yang, Ukadike C. Ugbolue, Andrew Kerr, Vladimir Stankovic, Lina Stankovic, Bruce Carse, Konstantinos T. Kaliarntas, and Philip J. Rowe. "Autonomous gait event detection with portable single-camera gait kinematics analysis system." In: *Journal of Sensors* 2016 (2016).

[164] Jiantao Yang and Yuehong Yin. "Dependent-Gaussian-Process-Based Learning of Joint Torques Using Wearable Smart Shoes for Exoskeleton." In: *Sensors* 20.13 (2020).

[165] Huiying Yu, Murad Alaqtash, Eric Spier, and T. Sarkodie-Gyan. "Analysis of muscle activity during gait cycle using fuzzy rule-based reasoning." In: *Measurement* 43.9 (2010), pp. 1106–1114.

[166] Wenhao Yu, Greg Turk, and C. Karen Liu. "Learning Symmetric and Low-Energy Locomotion." In: *ACM Trans. Graph.* 37.4 (2018).

[167] Hamed Zamani and W. Bruce Croft. "On the Theory of Weak Supervision for Information Retrieval." In: *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. ICTIR '18. Association for Computing Machinery, 2018, 147–154.

[168] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. "Weakly Supervised 3D Human Pose and Shape Reconstruction with Normalizing Flows." In: *Computer Vision – ECCV 2020*. Springer International Publishing, 2020, pp. 465–481.

[169] Petrissa Zell and Bodo Rosenhahn. "A physics-based statistical model for human gait analysis." In: *German Conference on Pattern Recognition (GCPR)*. Oct. 2015.

[170] Petrissa Zell and Bodo Rosenhahn. “Learning-Based Inverse Dynamics of Human Motion.” In: *The IEEE International Conference on Computer Vision (ICCV) Workshops*. Oct. 2017, pp. 842–850.

[171] Petrissa Zell and Bodo Rosenhahn. “Learning inverse dynamics for human locomotion analysis.” In: *Neural Computing and Applications* 32.15 (2020), pp. 11729–11743.

[172] Petrissa Zell, Bodo Rosenhahn, and Bastian Wandt. “Weakly-supervised Learning of Human Dynamics.” In: *European Conference on Computer Vision (ECCV)*. Aug. 2020.

[173] Petrissa Zell, Bastian Wandt, and Bodo Rosenhahn. “Joint 3D Human Motion Capture and Physical Analysis from Monocular Videos.” In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.

[174] Petrissa Zell, Bastian Wandt, and Bodo Rosenhahn. “Physics-Based Models for Human Gait Analysis.” In: *Handbook of Human Motion*. Cham: Springer International Publishing, 2018, pp. 267–292.

[175] Joseph A. Zeni, James G. Richards, and Jill S. Higginson. “Two simple methods for determining gait events during treadmill and overground walking using kinematic data.” In: *Gait & Posture* 27.4 (2008), pp. 710–714.

[176] Xiao-Yu Zhang, Haichao Shi, Changsheng Li, Kai Zheng, Xiaobin Zhu, and Lixin Duan. “Learning Transferable Self-Attentive Representations for Action Recognition in Untrimmed Videos with Weak Supervision.” In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (2019), pp. 9227–9234.

[177] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. “Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach.” In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017.

[178] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks.” In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2242–2251.

[179] Yuliang Zou, Jimei Yang, Duygu Ceylan, Jianming Zhang, Federico Perazzi, and Jia-Bin Huang. “Reducing Footskate in Human Motion Reconstruction with Ground Contact Constraints.” In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2020.

Petrissa Zell

Personal Data

Year of Birth:	1989
Place of Birth:	Langenhagen, Germany
email:	zell@tnt.uni-hannover.de

Work Experience

2014 - *present*	Research assistant at the **Leibniz University Hannover (LUH)**, **Institut für Informationsverarbeitung**, Main Focus: Physical Modelling of Human Motion, Thesis: "Learning-based Inverse Dynamics for Human Motion Analysis" | Advisor: Prof. Dr.-Ing. Bodo Rosenhahn
2012 - 2013	Teaching assistant at the **LUH**, **Institut für Festkörperphysik**, Courses: Experimental Physics 1: Mechanics and Relativity, Seminar about Nobel Laureates
2010 - 2011	Teaching assistant at the **LUH**, **Institut für Quantenoptik**, Courses: First year physics lab

Education

Dec 2013	MSc in Physics at the **LUH**, **Albert Einstein Institut für Gravitationsphysik**, Thesis: "Thesis: Entanglement between near infrared and visible light via frequency up-conversion" | Advisor: Prof. Dr. Roman Schnabel
Oct 2011	BSc in Physics at the **LUH**, **Institut für Festkörperphysik**, Thesis: "Ultrafast spin noise spectroscopy - comparison between simulation and experiment" | Advisor: Prof. Dr. Michael Oestreich

VDI verlag

Alle 23 Reihen der „Fortschritt-Berichte VDI“
in der Übersicht – bequem recherchieren unter:
elibrary.vdi-verlag.de

Und direkt bestellen unter:
www.vdi-nachrichten.com/shop

Reihe 01 Konstruktionstechnik / Maschinenelemente
Reihe 02 Fertigungstechnik
Reihe 03 Verfahrenstechnik
Reihe 04 Bauingenieurwesen
Reihe 05 Grund- und Werkstoffe/Kunststoffe
Reihe 06 Energietechnik
Reihe 07 Strömungstechnik
Reihe 08 Mess-, Steuerungs- und Regelungstechnik
Reihe 09 Elektronik/Mikro- und Nanotechnik
Reihe 10 Informatik/Kommunikation
Reihe 11 Schwingungstechnik
Reihe 12 Verkehrstechnik/Fahrzeugtechnik
Reihe 13 Fördertechnik/Logistik
Reihe 14 Landtechnik/Lebensmitteltechnik
Reihe 15 Umwelttechnik
Reihe 16 Technik und Wirtschaft
Reihe 17 Biotechnik/Medizintechnik
Reihe 18 Mechanik/Bruchmechanik
Reihe 19 Wärmetechnik/Kältetechnik
Reihe 20 Rechnergestützte Verfahren
Reihe 21 Elektrotechnik
Reihe 22 Mensch-Maschine-Systeme
Reihe 23 Technische Gebäudeausrüstung

VDI nachrichten Recruiting Tag und VDI nachrichten Recruiting Tag Online. Deutschlands führende Karrieremessen für Ingenieure und IT-Ingenieure.
Für alle Studierenden der Ingenieurwissenschaften, Absolventen und Young Professionals ein absolutes Muss. Knüpfen Sie Kontakte zu renommierten Unternehmen und sprechen Sie direkt mit Entscheidern aus den Fachabteilungen. Viele Serviceangebote wie Karriereberatung und -vorträge unterstützen Sie bei Ihrem erfolgreichen Einstieg ins Berufsleben.

Jetzt informieren und kostenfrei teilnehmen: www.ingenieur.de/recruitingtag

Mehr Informationen?
Silvia Becker, Telefon: +49 211 6188-170
Franziska Opitz, Telefon: +49 211 6188-377
recruiting@vdi-nachrichten.com

VDI nachrichten
recruiting tag

VDI

REIHE 10
INFORMATIK/
KOMMUNIKATION

NR. 877

ISBN 978-3-18-387710-2
E-ISBN 978-3-18-687710-9

BAND
1|1

VOLUME
1|1