Recovering Optimal Cost Functions for Natural Walking: From Musculoskeletal Simulation to Exoskeleton Control

by

Jiacheng Weng

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2022

Author's Declaration

This thesis consists of materials all of which I authored or co-authored: see the State of Contributions included in this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Jiacheng Weng was the sole author of the Chapters 1, 2, and 6.

Chapter 3 contains excerpts from a publication [97] that was co-authored by Jiacheng Weng, Dr. Arash Arami, and Dr. Ehsan Hashemi in the peer-reviewed journal *IEEE Robotics and Automation Letters*, "Natural Walking With Musculoskeletal Models Using Deep Reinforcement Learning" (March 19, 2021).

Chapter 4 contains excerpts from a paper [98] that was co-authored by Jiacheng Weng, Dr. Arash Arami, and Dr. Ehsan Hashemi prepared for the peer-reviewed journal *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, "Adaptive Reference Inverse Optimal Control for Natural Walking with Musculoskeletal Models".

In Chapter 5, Jiacheng Weng was the sole author for the development of the human-exoskeleton model as well as the pipeline for solving optimal gait trajectories for the human-exoskeleton system. The experiments of adopting optimal gait trajectories to the real exoskeleton were conducted by Jiacheng Weng and Mohammad Shushtari. Jiacheng Weng provided the optimal joint angle and feedforward torque exoskeleton trajectories. Mohammad Shushtari developed the exoskeleton tracking controller.

Abstract

Human movement studies have contributed to our understanding of how the central nervous system's (CNS) interactions with our body result in rich and complex motor behaviours, such as human gait. Such understanding is particularly important for human-centred engineering such as lower-limb exoskeletons. Assuming the emerged natural gait patterns are the result of some optimization done by CNS, researchers modelled the walking simulation problem as an optimization problem that recast the walking task into a cost function. However, accurately capturing the CNS goal within the cost function is challenging. Cost functions in existing studies were often assumed a priori which either did not lead to natural gait behaviour, or were manually tuned based on the researcher's knowledge which is time-consuming. Some studies attempted to tune the cost function algorithmically using inverse optimal control (IOC), but suffered from expensive computation. These limitations hinder the use of IOC for personalized cost function tuning and, by extension, exoskeleton controller design.

To address this issue, computationally efficient tuning methods of the cost function were designed and validated in two optimization frameworks: deep reinforcement learning (DRL) and predictive simulation. For DRL, a novel learning method, which generates a control policy with close-to-natural walking behaviour, was developed. The proposed neuromechanically-inspired cost function contributed to the effective learning of the realistic gait by the DRL agent. The nature-inspired curriculum learning scheme led to efficient convergence to natural and bilateral symmetric gait by adaptively tuning the cost function weights while maintaining the agent's walking capability. To further improve the cost function tuning efficiency, an efficient IOC algorithm named Adaptive Reference IOC (AR-IOC) was proposed that used direct collocation for solving optimal gait trajectories and gradient-based weight optimization. We showcased the efficiency of the proposed algorithm in tuning cost functions and matching gait trajectories using both synthetic data and experimental data which outperformed the Genetic Algorithm by more than 80% in computational time. With the AR-IOC, the correlation between the walking tasks and the cost function weights was studied which revealed a change in cost function compositions with respect to walking speed.

With the efficient AR-IOC algorithm, we explored the potential of using predictive simulation to generate physics-informed reference trajectories for lower-limb exoskeleton tracking controllers. First, an accurate human-exoskeleton system was developed. By combining the optimal human cost function obtained using AR-IOC and the exoskeleton cost function, we obtained the optimal gait trajectories for the human-exoskeleton system which were different from the unassisted natural walking trajectories. These optimal trajectories

were then tested in real exoskeleton systems using a time-dependent proportional-derivative (PD) controller and their performances in reducing muscle activities were compared to the unassisted natural walking trajectories. The likely limitations of the controller design were also discussed.

With the proposed simulation frameworks and the efficient cost function tuning methods, this thesis serves as a catalyst for enabling personalized rehabilitation design based on detailed musculoskeletal simulation. The presented framework, which covers from data collection and post-processing, to simulation and experiments, serves as a guidance and reference to future developments in this field, such as extending the musculoskeletal simulation to impaired subjects with different locomotion tasks, and different control systems for lower-limb rehabilitation.

Acknowledgements

I would like to sincerely thank my supervisor, Dr. Arash Arami, and my co-supervisor, Dr. Ehsan Hashemi for their guidance and support during my master's studies and throughout the development of this thesis.

I would like to thank Dr. John McPhee and Dr. Katja Mombaur for being part of the review committee for my thesis.

I would like to thank my colleagues in the Neuromechanics and Assistive Robotics Laboratory for their support and assistance in my master's studies and research.

I would also like to thank my parents for their support both emotionally and financially, especially during this difficult period of the pandemic.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AR-IOC Adaptive Reference Inverse Optimal Control 3, 27, 32, 34, 36, 38, 43, 44, 49, 52, 55, 65

BF Biceps Femoris 60–62

BFsh Biceps Femoris short head 14, 28, 60

BoS base of support 31, 54

CL curriculum learning 2, 3, 9, 14, 17–21, 27, 64

CNS central nervous system 1, 3, 11, 30, 41, 55, 64

CoM center of mass 15, 16, 30, 31, 38, 42, 49, 51, 54, 55

DC direct collocation 6, 7, 9, 15, 16, 19, 24, 26, 29, 31–36, 42–44, 52, 54, 55, 58, 59, 65

DK direct kinematics 1, 4

DLM Direct loss minimization 10, 27, 32, 34

DoF degrees of freedom 4, 10, 14, 28, 31

DRL deep reinforcement learning 2, 3, 9, 13–19, 24–27, 64, 65

EMG electromyography 4, 40, 45, 46, 59, 60

GA genetic algorithm 3, 10, 43–45, 65

GAS Gastrocnemius 14, 28, 40, 60

GASl Gastrocnemius Lateralis 60–62

GASm Gastrocnemius Medialis 45

GM Gluteus Maximus 14, 28, 45

GRF ground reaction force 30, 33, 38, 41, 44–46, 48, 55, 56, 60

HAM Hamstrings 14, 28, 40, 60

ID inverse dynamics 1, 4

IK inverse kinematics 1, 4

IL Iliopsoas 14, 28

IOC inverse optimal control x, 3, 10, 11, 27, 28, 36, 38–43, 48, 49, 65

IPOPT Interior Point OPTimizer 6

IRL inverse reinforcement learning 10, 11

KKT Karush–Kuhn–Tucker 10

MH metatarsal head 28, 45

MPC model-predictive controller 7

MS multiple shooting 6, 7, 9

NLP non-linear program 6, 7

OC optimal control 2, 5–7, 10, 13, 64

PD proportional-derivative 58

PG policy gradient 8

RF Rectus Femoris 14, 28, 40, 45, 60–62

RL reinforcement learning 5, 7, 8, 10, 11, 14, 64

RMSE root mean square errors 40, 41

SC structured classification 10, 27, 32

SD standard deviation 38

SENIAM Surface Electromyography for the Non-Invasive Assessment of Muscles 45

SI symmetry index 18, 20

SMBO Sequential model-based optimization 11

SOL Soleus 14, 28, 45

TA Tibialis Anterior 14, 28, 40, 45, 60–62

VA Vasti 14, 28, 40, 60

VAl Vastus Lateralis 45

ZMP zero-moment point 16

Chapter 1

Introduction

Human movement studies have contributed to our understanding of how the central nervous system (CNS) in interacting with our body results in rich and complex motor behaviours, such as human gait [100]. Understanding the human control strategies during walking is particularly important for human-centreed engineering such as the design of lower-limb exoskeletons and their controllers[103]. However, performing studies with human participants is often expensive (i.e., instruments for data collection and human labour) and difficult (i.e., complex experimental setup, strict safety measurements, and limited human participants). As many early-phase developments do not necessarily require human involvement, human movement simulation [22] using tools such as MapleSim [80, 81], AnyBody [12], and OpenSim [15] have gained popularity in analyzing walking. With a rich collection of detailed musculoskeletal models, researchers are able to investigate both the kinematic and dynamic characteristics of the human body by recreating the walking motion in simulation [86]. Although conventional approaches such as direct kinematics (DK), inverse kinematics (IK), and inverse dynamics (ID) showed success in gait analysis [32], they fully rely on the experimental data and cannot be used as a predictive simulation platform (i.e., a simulation platform capable of extrapolating gait patterns adaptively based on the dynamics of the system beyond the observed experimental reference) for researchers to test out their robotic devices virtually.

Assuming the emerged natural gait patterns is the result of some optimization done by CNS, researchers modelled the walking simulation problem as an optimization problem [88]. The optimization problem abstracts the walking task into a cost function which allows the researchers to simulate the walking motion without the need for experimental data [16]. However, accurately capturing the human rationale within the cost function has always been a challenge. Cost functions in existing studies are assumed a priori [14] which

often do not lead to natural gait behaviour, or manually tuned based on the researcher's knowledge [24] which is time-consuming. Some studies attempted to tune the cost function algorithmically [67] based on experimentally collected human gait data, but suffered from expensive computation. These limitations hinder the extension of personalized cost function tuning to the design of exoskeleton controllers. Thus, there is a need to develop efficient cost function tuning methods for walking.

1.1 Objectives and Outline

The main goal of this research is to obtain optimal cost functions for natural walking. The means for obtaining the optimal cost functions should be efficient, and the obtained cost functions should enable the predictive simulation platform. As a result, the fulfillment of the main goal is broken down into four major objectives:

1. Construct predictive simulation environment for natural walking using musculoskeletal models.
2. Formulate efficient optimization problems to obtain optimal gait trajectories given cost functions.
3. Develop computationally-efficient inverse optimization techniques to optimize the cost functions for natural walking.
4. Validate the obtained optimal cost functions in assisting lower-limb exoskeleton controller design.

We choose two frameworks as the foundations of the predictive simulation environment: optimal control (OC) and deep reinforcement learning (DRL). In Chapter 2, theoretical background of OC and DRL as well as their limitations in locomotion studies are provided. Existing methods for cost function tuning and their constraints for extending to locomotion studies are also discussed. At the end of the Chapter 2, a brief review of the existing controllers for lower-limb exoskeletons and their issues are provided, which marks the potential for improvement using optimization-based approaches. In Chapter 3, we developed a nature-inspired curriculum learning (CL) scheme for adaptive cost function weight update along with a neuromechanically-inspired cost function based on section 2.2.1 to efficiently train a DRL agent that outperforms the existing DRL walking agent in both learning efficiency and gait naturalness. To further improve the efficiency of cost function

tuning, Chapter 4 covers our novel IOC approach called Adaptive Reference Inverse Optimal Control (AR-IOC) to efficiently recover optimal human cost functions during walking. In Chapter 5, we study the benefit of using the optimal cost function obtained using AR-IOC from Chapter 4 to assist exoskeleton tracking controller design. In Chapter 6, concluding remarks are provided.

1.2 Contributions

The main contribution of this research is summarized as below:

1. (**Chapter 3**) Propose a neuromechanically-inspired cost function for realistic gait pattern generation in the DRL setting.
2. (**Chapter 3**) Propose a nature-inspired CL scheme for efficient DRL convergence for learning-to-walk problems without using any experimental tracking data.
3. (**Chapter 3**) Generate a DRL policy that demonstrates close-to-natural walking gait on musculoskeletal models.
4. (**Chapter 4**) Formulate the AR-IOC: an efficient IOC framework for natural walking in detailed musculoskeletal models with a significant computational efficiency improvement compared to the existing derivative-free optimization method genetic algorithm (GA).
5. (**Chapter 4**) Demonstrate the success of the proposed method in recovering cost functions based on multiple participant-specific reference data.
6. (**Chapter 4**) Investigate the change in optimal cost functions with respect to the walking speed.
7. (**Chapter 5**) Develop a realistic human-exoskeleton system in simulation.
8. (**Chapter 5**) Evaluate the tracking controller performance on a human participant using the optimal reference trajectories obtained from the human-exoskeleton system assuming identical CNS cost function structure for both natural and assistive walking (explained in section 5.2).

Chapter 2

Background

2.1 Gait Analysis

In human movement studies, trajectories of lower-limb joint angles and ground reaction forces are the most common information used to analyze the gait behaviour [54, 32]. In certain cases, surface electromyography (EMG) signals are also collected to identify activated muscles during locomotion [9]. To obtain joint angle trajectories, reflective markers are first attached to anatomical landmarks on human body segment [13]. Motion capture systems (e.g., Vicon Motion Systems, UK) are then used to capture the 3D marker positions. Tools such as conventional gait models [4, 51] are developed that uses DK (i.e., directly calculate kinematic profiles from marker trajectories) to reconstruct the joint angle trajectories. To obtain the gait kinetics such as joint torque and energy, ground reaction forces are collected to enable ID analysis on the gait. Different from DK, IK requires a predefined rigid skeletal model where the degrees of freedom (DoF) of each joint is predefined. The joint angle trajectories are then reconstructed by tracking the marker positions. Detailed and accurate rigid skeletal models for biomechanical analysis are provided by the dynamic simulation software including MapleSim [80, 81], AnyBody [12], and OpenSim (open-source) [15, 79]. These skeletal models also include detailed muscle connections with realistic muscle dynamics which makes analysis of muscle control (e.g., muscle excitation and activation) and energistics (e.g., metabolic energy consumption) possible [87, 86].

2.2 Dynamic Walking Simulation

With the experimentally obtained joint kinematics, ground reaction forces and a predefined rigid skeletal model, researchers are able to replay the recorded locomotion and perform analysis for both normal and pathological gait [41]. However, this simulation method has three major limitations. First, experiment replay in simulation is constrained to the experimental conditions and cannot be easily extrapolated to different skeletal models with different anthropometric parameters or different walking conditions. Second, the human rationale for walking cannot be explained by the experimental data alone as the walking kinematics and kinetics do not directly explain why human converges to such gait patterns. Last, replaying recorded kinematic data does not guarantee the dynamic feasibility in the simulation environment as it ignores the ground contact mechanisms. Simulating ground contact is important as it not only provides hard constraints to the foot positions, but also generates dynamically feasible ground support. Due to the marker position errors, reconstructed foot positions in simulation sometimes fail to maintain contact to the ground during stance which is not possible in reality. To address this issue, some studies performed forward simulations using reflex-based controllers for natural walking [31] and trajectory tracking-based controllers for both natural walking and assistive walking with the exoskeletons [82].

One approach to address the aforementioned limitations is to formulate the walking simulation problem as an optimization problem. An optimization problem tries to find a best state and control sequence (i.e., gait patterns in this case) by minimizing an cost function that serves as a high-level abstraction of the task to be optimized (i.e., human rationale during walking). In this section, we discuss two main frameworks for formulating the optimization problems for walking including OC and reinforcement learning (RL). Challenges of these two frameworks are also discussed.

2.2.1 Optimal Control

A general OC problem can be formulated in the following form:

$$\min_{T,\boldsymbol{x},\boldsymbol{u}} J_b(t_0, T, \boldsymbol{x}(t_0), \boldsymbol{x}(T)) + \int_{t_0}^{T} J_i(t, \boldsymbol{x}(t), \boldsymbol{u}(t))dt, \tag{2.1}$$

$$
\begin{aligned}
\text{subject to: } & \boldsymbol{g}(T, \boldsymbol{x}(t_0), \boldsymbol{x}(T)) \leq \boldsymbol{0}, & & \boldsymbol{h}(T, \boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}, \\
& t_{low} \leq T \leq t_{up}, & & \dot{\boldsymbol{x}}_t = \boldsymbol{f}(t, \boldsymbol{x}_t, \boldsymbol{u}_t), \\
& \boldsymbol{x}_{0,low} \leq \boldsymbol{x}(t_0) \leq \boldsymbol{x}_{\boldsymbol{0,up}}, & & \boldsymbol{x}_{low} \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_{\boldsymbol{up}}, \\
& \boldsymbol{x}_{T,low} \leq \boldsymbol{x}(T) \leq \boldsymbol{x}_{\boldsymbol{T,up}}, & & \boldsymbol{u}_{low} \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_{\boldsymbol{t}}.
\end{aligned}
$$

The goal of the optimization is to find the optimal state $\boldsymbol{x}$, control signals $\boldsymbol{u}$, and final time T such that the cost function is minimized. The cost function consists of boundary cost J_b and integral cost J_i along time t. The optimization is also subjected to constraints including boundary constraints $\boldsymbol{g}$, path constraints $\boldsymbol{h}$, system dynamics, and variable bounds on the states and controls where low, up represents the lower and upper bounds of the variables. This general formulation of the optimal control problems involve differential equations and works in continuous time which is difficult to handle in numerical optimization. To address this, the problems are discredited and transcript to a non-linear program (NLP) that only involves algebraic equations [44]. A general form of the non-linear program can be seen as follows:

$$
\begin{aligned}
& \min_{\boldsymbol{z}} J(\boldsymbol{z}), \\
\text{subject to: } & \boldsymbol{g}(\boldsymbol{z}) \leq \boldsymbol{0}, \\
& \boldsymbol{h}(\boldsymbol{z}) = \boldsymbol{0},
\end{aligned}
\tag{2.2}
$$

where $\boldsymbol{z}$ represents all the variables from after the problem discretization; $J(\boldsymbol{z})$ is the cost function of the discretized variables; $\boldsymbol{g}(\boldsymbol{z})$ and $\boldsymbol{h}(\boldsymbol{z})$ are a set of inequality and equality constraints of the discretized problem. The NLP is then solved using non-linear optimization methods such as the Interior Point OPTimizer (IPOPT) [94]. OC techniques are considered to be model-based as it requires a known dynamic model of the system to perform numerical integration.

For locomotion tasks that generally have high-dimensional state space and complex controls, methods such as multiple shooting (MS) and direct collocation (DC) are used to perform the problem discretization and transcription. For torque-driven humanoid robots, MS has been widely experimented [61, 10, 25]. Although, DC is also used for humanoid robots [18, 36, 35], it is more common for solving trajectory optimization problems with musculoskeletal systems due to its ability to handle non-linear and redundant muscle actuation [93, 64, 90, 2, 67, 23, 24]. Common performance measures (i.e., description of J_i

in (2.1)) include metabolic energy [64, 90, 24], muscle activation [2, 23, 24], joint hyperextension [64, 90, 23, 24], stability [67], foot sliding [64], motion smoothness [67, 24], and norm of tracking error [23]. These cost functions provide an abstraction of the walking task optimality that can be used in different scenarios such as assistive walking with prostheses [68].

With the development of tools such as OpenSim Moco [16], researchers are able to formulate the DC problems for complex musculoskeletal models efficiently. Due to the complexity of the musculoskeletal systems, the transcribed NLP for the OC problem often gets large and takes long time to solve. Thus, existing studies often work with relatively short trajectories and small number of discretized points [90, 2, 67, 23, 24]. In addition, MS and DC only generate open-loop solutions which does not guarantee stability in a control scenario. The long computational time (i.e., much slower than real time) on the musculoskeletal models makes neither MS nor DC suitable as the real-time model-predictive controller (MPC).

2.2.2 Reinforcement Learning

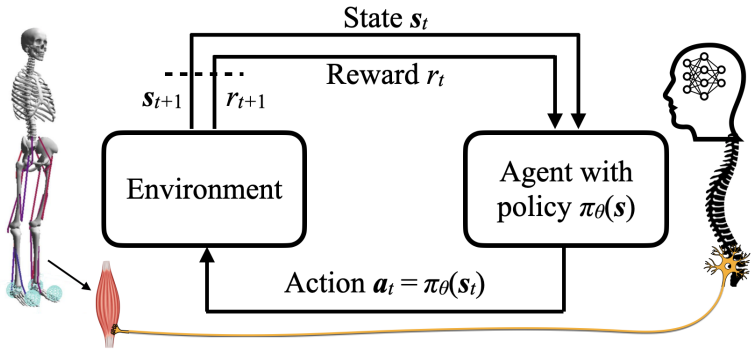

Figure 2.1: Agent-environment interaction in DRL [97]

RL [85], as a subcategory of machine learning, addresses the optimal control problem through agent-environment interaction. A general RL problem consists of an agent and an environment in interaction with each other. Taking musculoskeletal control as an example shown in Fig. 2.1, the environment includes the dynamic description of a musculoskeletal model. The agent represents a neuromuscular controller that has its policy π_θ (i.e., some

function that represents the control strategy) parameterized by θ. Given some state observation $\boldsymbol{s}_t$ (e.g., coordinate position and velocity), the policy generates an action (i.e., a control command such as muscle excitation) $\boldsymbol{a}_t$ that drives the environment to the next state $\boldsymbol{s}_{t+1}$ constrained on the system dynamics $\boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ where f can be stochastic or deterministic. The environment also provides a reward signal $r_t = R(\boldsymbol{s}_t, \boldsymbol{a}_t)$ (opposite to cost) that evaluates the goodness of the previous action $\boldsymbol{a}_t$ at state $\boldsymbol{s}_t$. The goal of RL is to find an optimal policy such that the expected cumulative reward is maximized.

Given a policy π, a sequence of state-action pairs can be obtained. The goodness of the sequence at time t is captured by the discounted accumulative reward

$$G_t = \sum_{k=0}^{\infty} (\gamma^k r_{t+k+1}), \tag{2.3}$$

that sums over individual reward values from time $t+1$ onward to the terminal state, where $\gamma \in [0, 1]$ is a discount factor. If the environment is stochastic (i.e., $\boldsymbol{s}_{t+1}$ has its own probability distribution given $\boldsymbol{s}_t, \boldsymbol{a}_t$), we need to compute the expectation of all possible future sequences which is given by the return

$$Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_{\pi}[G|\boldsymbol{s}, \boldsymbol{a}]. \tag{2.4}$$

If the policy is stochastic (i.e., the agent can select different actions given the action probability distribution), we can then formulate the objective of the reinforcement learning problem to the following:

$$\max_{\pi} J(\theta) = \max_{\pi} \sum_{\boldsymbol{s} \in S} d^{\pi}(\boldsymbol{s}) \sum_{\boldsymbol{a} \in A} \pi_{\theta}(\boldsymbol{a}|\boldsymbol{s}) Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}), \tag{2.5}$$

where S, A are sets of all possible states and actions, respectively; $d^{\pi}(\boldsymbol{s})$ is the distribution of all possible states under π; $\pi_{\theta}(\boldsymbol{a}|\boldsymbol{s})$ is the probability of $\boldsymbol{a}$ given $\boldsymbol{s}$.

One common approach to solve for the optimal policy is called the policy gradient (PG) method [85] which computes the gradient of the objective function $J(\theta)$:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\pi}[Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) \nabla_{\theta} \ln \pi_{\theta}(\boldsymbol{a}|\boldsymbol{s})]. \tag{2.6}$$

However, such gradient estimates experience high variance which can lead to learning inefficiency. Actor-critic algorithm addressed this issue by replacing $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a})$ with the advantage term $A(\boldsymbol{s}, \boldsymbol{a})$ [77, 50]:

$$A(\boldsymbol{s}, \boldsymbol{a}) = Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) - V(\boldsymbol{s}) \tag{2.7}$$

where $V(s)$ is the value function that estimates the expected cumulative return over all possible $\boldsymbol{a}$ at $\boldsymbol{s}$. In practice, $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a})$ is computed from the agent's collected experience, whereas the V is a estimator to be optimized. For complex problems with high dimensional, continuous state-action spaces, representing π and V is difficult. DRL addresses this issue by representing π and V by deep neural networks whose parameters are iteratively updated using the collected batch sequences.

Different from previously mentioned methods (i.e., MS and DC) where only open-loop solutions are generated, DRL aims to learn a feedback policy that gives optimal solutions at any states within the scope of the problem. This makes solving DRL problems difficult. Solving locomotion problems with DRL is particularly challenging due to the high dimensional state-action spaces (i.e., the curse of dimensionality based on realizing the Bellman's optimality using dynamic programming [21]) and expensive simulation of the musculoskeletal systems. The recent advances in DRL algorithms [78, 76, 55, 33, 30] have shown success in controlling complex robotic system such as humanoids. This encouraged researchers to consider learning based methods for studying and simulating natural locomotion behaviour [83]. In computer graphics, reference motion is often used when developing DRL agents [70, 53] since generating natural behaviour is the primary goal. However, the underlying optimal criteria that lead to the desired motion is not considered. In the machine learning research community, DRL has been used to realize high level motor control on accurate musculoskeletal models without reference motion [45, 46, 47, 83] but failed to obtain natural behaviour. Some studies showed realistic walking [14] and running gait [45, 46] learned by DRL agents but with imitation learning or manual agent selection during the learning process. Other studies developed a walking-capable DRL agent [47, 83] but with abnormal gait. In these studies, the reward functions are often simpler than those in optimal control studies with primarily task specific rewards such as target moving speed and directions. Neuromechanical cost terms that contributes to the naturalness of the behaviour are often neglected.

One reason that makes the use of DRL for locomotion difficult is that DRL is prone to convergence to local optima especially in high-dimensional setting. CL aims to improve learning progress by decomposing a challenging objective into multiple intermediate steps that are easier to learn [5]. The environment-based curriculum alters the difficulty of a parameterized environment to achieve better agent performance and generalization over a wide range of conditions [102, 52]. The goal-based curriculum distinguishes intermediate learning steps by providing different reward functions. This strategy has been used in both NeurIPS Learning to Run and AI for Prosthetics challenges [46, 47, 104, 83]. Although these curriculum designs help agent convergence to some walking capability, the locomotion still appears to be unnatural.

2.2.3 Cost Function Tuning

A well-posed cost function (or a reward function for RL) is essential to solve the corresponding problem effectively. However, having such a cost function is challenging for both OC and RL. IOC addresses the problem of recovering the underlying cost function given the desired state trajectories such as human demonstrations [63]. IOC problems are often formulated as bilevel optimization problems with a parametric cost function given a priori. The inner-loop problems find optimal state trajectories given the cost function, whereas the outer-loop problems tune the parameters of the cost function so that the inner-loop solutions resemble the desired state trajectories. For problems that only involve kinematics such as motion path prediction (e.g., obstacle avoidance during walking), Karush–Kuhn–Tucker (KKT) conditions have been imposed to optimize the weights with high computational efficiency [20, 71]. However, for more complex cost functions that involve system dynamics (e.g., joint torques), derivative-free optimization techniques such as COBYLA [11], BOBYQA [62] and CMA-ES [19] are used. These techniques are based on strategic cost weight sampling (local approximation of the optimization problem or population-based search) coupled with internal evaluations of the sampled cost functions. In practice, only IOC problems with efficient internal evaluations and a few cost weights can be solved within a reasonable time frame. However, for locomotion problems that utilize musculoskeletal models, the internal optimization problems (usually in the form of trajectory optimization problems) are challenging to solve as the musculoskeletal models have many DoFs and redundant muscle actuators. In the existing locomotion studies with musculoskeletal models, the cost function weights are often manually tuned based on the researcher's domain knowledge [90, 24, 23]. With a reasonable simplification of the musculoskeletal models by constraining sagittal-plane motion and reducing the number of muscles, researchers formulated the IOC problem for natural walking using GA for optimal cost weight search [67], but it still requires a long computational time. On the other hand, structured classification (SC) treats the IOC problem as a structured prediction problem (i.e., a multi-class classification problem with an extremely large number of classes) and allows gradient-based update to the cost function weights, which can be more efficient than derivative-free methods [73, 74]. Direct loss minimization (DLM), which is the successor to the vanilla structured prediction, achieves better performance by using a loss-adjusted cost function for lower-level optimization (i.e., adding an additional domain-specific loss term to push the optimized trajectories towards the preferred region of search) during weight update [59]. This marks the DLM a good candidate for solving the IOC for complex locomotion problems.

Similar to IOC, inverse reinforcement learning (IRL) tries to recover the expert's reward

function while solving the expert policies in the inner-loop optimization instead of specific trajectories. Maximum margin IRL solves the optimal cost function by increasing the distance between expert policies and all other policies [1, 73, 74]. Maximum entropy IRL [105, 28], on the other hand, finds the optimal cost function that matches the expectation of the expert behaviour while maximizing the entropy of the policy. All these IRL approaches require solving a forward RL problem (i.e., learning an expert policy through the agent's interaction with the environment) either fully or partially which is non-trivial by itself. This makes the extension of these methods to continuous and high-dimensional settings (i.e., muscle actuated human locomotion) difficult [46, 47].

In some applications (e.g., computer graphics) where the control policy is the primary need, imitation learning methods are used to solve the IRL problem where the cost function tuning is absorbed into the policy optimization and not explicitly defined [38]. These methods provide little insight into important factors that shape human locomotion and therefore are unsuitable for this study.

IOC problems with parametric cost functions can also be seen as a hyperparameter tuning problem [6] commonly found in machine learning field [26]. While naive optimization methods, including grid search and random search [7], can be used for IOC with easy parallelization, they usually suffer from high computational costs when the number of weights is large. Sequential model-based optimization (SMBO) is an alternative approach that utilizes a surrogate model (i.e., an approximation model that mimics the observed input-output relationship; often obtained through data-driven approaches) to estimate the relationship between the total cost values and the cost function weights [40]. Bayesian optimization is a common SMBO method that uses probabilistic models (e.g., Gaussian processes) to locate the good weight candidates during weight search [99]. Similar to other derivative-free methods, SMBO can be slow for human locomotion study as the inner-loop optimization problem is difficult to solve.

2.3 Lower-limb Exoskeleton

One main purpose of developing accurate walking simulation is to facilitate the design of assistive robotic devices such as lower-limb exoskeletons. The walking simulation with a cost function that represents CNS role in optimizing the gait (can be obtained by cost function tuning methods mentioned in section 2.2.3) can help design controllers that better align with human intention and perform exoskeleton controllers within the simulation environment without the risk of involving real humans.

In the past two decades, Many lower-limb exoskeleton control strategies for walking have been proposed [103]. Predefined gait trajectory control is a common control method for rehabilitation exoskeleton which often uses experimentally obtained joint angles from healthy subjects as the reference for the feedback controller [95]. As humans modify gait trajectories for different physiological parameters and walking conditions, various gait adaptation methods have been proposed. For example, A data-driven personalized reference trajectory generator was developed for walking without exoskeleton [101] to account for various body geometries and walking speed. However, this approach does not consider the interaction between the human and the exoskeleton which can potentially alter the human gait. Some adaptation methods identify the human intended trajectories directly from the subjects which requires the subject to have partial mobility (i.e. one fully functional leg) [92, 56]. For online adaptation, the healthy reference trajectories are often modified based on optimization criteria such as interaction force [42, 82] to better align with the human intended gait trajectories. However, such methods often apply constraints on the modified trajectories that either maintain the overall gait trajectory shape and only apply scaling and shifting, or apply the gait deviation penalty. These constraints can improve the stability of gait adaptation, but may limit the controller from adapting different but more optimal gait strategies for the human-exoskeleton system. In addition, using sub-optimal initial gait trajectories in the exoskeleton controller may impose safety issues as the conflict between the initial gait trajectories and the human gait can initiate falling [42]. Thus, a close-to-optimal initial gait trajectory for the human-exoskeleton system is preferred.

To obtain an optimal gait trajectory for the human-exoskeleton system, one must consider the human intention, passive dynamics from the exoskeleton, and the exoskeleton assistance. For example, predictive simulation has been used to optimize passive prosthesis [27, 84] to improve performance such as reducing joint force and metabolic energy rate. OpenSim is used to analyze the impact of ideal torque assistance devices to running [89] and walking with heavy loads [17]. It is noted in [89] that the optimal assistive torque may differ from the human's biological torque, and the benefit of assistance can occur at a location different from where the assistance is applied. This indicated the complexity of analyzing assisted human locomotion. For walking, predictive simulation has been used to simulate the impact of ideal ankle exoskeleton [**?**] and knee exoskeleton [49]. However, the passive dynamics of the exoskeleton such as the inertia and joint friction is often ignored, which can greatly reduce the reliability of the simulation results when the exoskeleton is heavy. Predictive simulation studies involving exoskeleton dynamics have been performed for walking [48] to obtain optimal control torques given reference motion data, and other movement tasks (e.g., lifting) [58] to obtain both optimal control torque and lifting motions. A case study from Franks *et al.* [29] also showed the limitations of applying results

from simulated assistive walking to experiments due to inaccurate modelling of the experimental scenario. Therefore, there is a need to improve the predictive simulation by including accurate human cost functions and proper exoskeleton dynamics. Walking gait optimization for the human-exoskeleton system is also desired.

2.4 Concluding remarks

Both OC and DRL suits the purpose of developing an predictive simulation environment for natural walking. These two methods generate optimal gait using cost functions and have the the capability for gait extrapolation in unseen virtual environments. However, obtaining the accurate cost function that reveals human rationale is challenging. Although existing OC studies for locomotion generate realistic open-loop motion trajectories, there is a lack of efficient methods for systematic cost function tuning. Existing DRL studies for locomotion produces more capable feedback policies for control, but lack the focus on naturalness of the motion emerged from the cost function, or directly use imitation learning from reference motion data. Hence, there is a need to develop efficient cost function tuning methods for both OC and DRL to achieve more close-to-natural gait patterns.

Accurate simulation of walking can facilitate the development of human-in-the-loop controllers for assistive robotic devices. Predefined gait trajectory control, being one of the most common control strategies for lower-limb exoskeletons, often fully or partially relies on the healthy gait. This neglects the change in system dynamics after the addition of the exoskeletons. On the other hand, predictive simulations have been used to analyze the optimal gait trajectories for an altered musculoskeletal system by prostheses or exoskeletons, but often lacks the consideration of the optimal human cost function or exoskeleton dynamics. Therefore, a high-fidelity simulation that considers these two aspects is required.

Chapter 3

Simulating Natural Walking Using DRL

This chapter describes a novel reward function tuning methods within the DRL formulation for efficient learning of close-to-natural human walking without reference motion. The content of this chapter is published by J. Weng, E. Hashemi and A. Arami in IEEE Robotics and Automation Letters Vol. 6(2): 4156-4162, 2021 and in the IEEE International Conference on Robotics and Automation, Xi'an, China, 2021. The proposed method aims to bridge the gap between collocation-based analysis and DRL-based locomotion study by applying neuromechanically-inspired reward functions. It also improves the learning progress using our goal-based CL strategy that resembles the human's natural learning progress. We first discuss the agent observation, reward shaping, and network architecture, followed by the goal-based curriculum formulation and training strategy.

3.1 Environment Formulation

The learning environment consists of a musculoskeletal model adopted from the OpenSim RL package used in the NeurIPS Learning to Run challenge [45]. The model includes trunk, pelvis, and leg segments (i.e., femur, tibia and talus-calcaneus-toes segments) with 18 muscles at the lower extremity and 9 DoF which allows only sagittal-plane motion. The 18 muscles (9 for each side) includes Hamstrings (HAM), Biceps Femoris short head (BFsh), Gluteus Maximus (GM), Iliopsoas (IL), Rectus Femoris (RF), Vasti (VA), Gastrocnemius (GAS), Soleus (SOL), and Tibialis Anterior (TA). The 9-DoF musculoskeletal model has

three global coordinates (x translation, y translation, and z rotation at the pelvis) and six local coordinates (one rotation for each hip, knee, and ankle joint). The musculoskeletal model uses a Hill-type muscle model and Hunt Crossley contact spheres at heel and toes for foot-ground interaction.

The state observation of the DRL agent has 33 dimensions: five at pelvis (vertical position and velocity, horizontal velocity, and rotational angle and velocity); 12 at lower limb joints (joint angle and velocity at hip, knee, and ankle); 12 at different body segments (vertical and horizontal position of head, torso, left and right toes, left and right talus relative to pelvis); two for vertical ground reaction forces; and two for center of mass (CoM)'s vertical and horizontal velocity. All velocity related observations are scaled down by 10. The vertical ground reaction forces are scaled down by 1000. However, unlike DC approaches and some DRL-based studies [47, 24, 67, 23], muscle activation is not included to ensure small observation space.

The reward function R in (3.1) consists of seven performance measures including muscle activation penalty R_a, hyperextension penalty R_h, instability penalty R_s, jerk penalty R_j, forward reward R_f, footstep reward R_{fs}, and survival reward terms R_{u_h}, R_{u_o}.

$$\begin{aligned} R =& \omega_a R_a dt + \omega_h R_h dt + \omega_s R_s dt + \omega_j R_j dt + \\ & \omega_f R_f dt + \omega_{fs} R_{fs} + \omega_u (R_{u_h} + R_{u_o}) dt, \end{aligned} \tag{3.1}$$

where the reward/penalty function components are

$$R_a = -0.1 \textstyle\sum_m a_m^3 \tag{3.2}$$

$$R_h = -0.1 \textstyle\sum_j |\tau_p^j| \tag{3.3}$$

$$R_s = -\left(p_{cx} + p_{cy} g^{-1} a_{cx} - p_{fb}\right)^2 \tag{3.4}$$

$$R_j = -10^{-4} \left(\alpha j_x^2 + (1-\alpha) j_y^2\right) \tag{3.5}$$

$$R_f = \begin{cases} -(v_{cx} - v_{refx})^2 + 1, & p_{cx} \leq 1\text{m}, \\ \exp(-8(v_{cx} - v_{refx})^2), & \text{else} \end{cases} \tag{3.6}$$

$$R_{fs} = \text{clip}(t_{fs}, \text{none}, 1) \tag{3.7}$$

$$R_{u_h} = 10^2 \text{clip}(p_{hy} - 1.45, \text{none}, 0.02) \tag{3.8}$$

$$R_{u_o} = 10^2 \text{clip}(-2(p_{hx} - p_{px})^2, -0.02, \text{none}) \tag{3.9}$$

The first four terms in (3.1) are inspired by existing trajectory optimization studies [90, 2, 64, 67, 24, 23]. R_a in (3.2) computes the sum of cubed normalized muscle activation,

denoted by $a \in [0, 1]$. R_h in (3.3) computes the sum of absolute passive joint torque τ_p at hip, knee, and ankle provided by rotational spring-damper actuators in the OpenSim [45]. R_s in (3.4) penalizes the deviation of the actual position of the front foot, p_{fb}, from the expected front foot position, computed using zero-moment point (ZMP) [43]. The expected front foot position is $(p_{cx} + p_{cy}g^{-1}a_{cx})$ where p_{cx} and p_{cy} are the forward and vertical position of the CoM, a_{cx} is the forward acceleration of CoM, g is $-9.81ms^{-2}$. R_j in (3.5) computes the jerk along the x and y directions with weighting factor $\alpha \in [0, 1]$ which is set to 0.11 based on [67].

In almost all reviewed studies that used DC, the walking speed, stride frequency, and gait symmetry or periodicity are enforced through hard constraints [64, 90, 2, 67, 24, 23]. This is usually done by specifying the trajectory time and initial/terminal states. In the DRL setting, these constraints are difficult to impose. Instead, the walking speed constraint is transformed into the forward reward, R_f in (3.6), which is computed using the forward speed of CoM, v_{cx}, and some target forward speed, v_{refx}. We use a quadratic form for R_f during the initial standing-to-walking transition to intensify the forward reward. After stabilizing the forward motion, R_f is evaluated using a Gaussian function with sharper attraction around v_{refx}. No constraint is implemented on the stride frequency and gait symmetry. Instead, a simple footstep reward, R_{fs} in (3.7), is provided based on step time t_{fs} to encourage alternating stepping behaviour during the initial learning phase. Besides, survival reward terms, R_{uh} in (3.8), and R_{uo} in (3.9) are used to maintain height and upward pose, where p_h, p_p are the head and pelvis positions. The clipping operation limits R_{uh} and R_{uo} within the defined boundaries to avoid undesired agent behaviour such as vertical jumping and unwillingness to move forward.

3.2 Training Strategies

Agent-environment interaction for our DRL problem formulation is shown as in Fig. 2.1. The learning-to-walk problem is decomposed into two phases. The first phase aims to efficiently generate a walking-capable agent with two requirements: relatively symmetric gait behaviour and no joint hyperextension. This walking-capable agent is then used as the starting point in the second phase which focuses on gait fine-tuning.

To efficiently generate the walking-capable agent in the first phase, we formulate a curriculum that resembles the natural learning progress of humans: learning to stand, learning to step, and learning to move forward as shown in table 3.1. The changing objectives of these three courses are realized by targeting different reward components in the reward function. During course transition, we maintain the same reward weights

from the previous course, and enable additional reward weights for the next course. Upon completion of each course, the DRL agent is directly used as the initial point of the next course.

Table 3.1: Target reward weights during phase one CL

Courses	**Target Weights**	**Description**
Learn to stand	ω_h, ω_u	survival, hyperextension
Learn to step	$\omega_h, \omega_u + \omega_{fs}$	+footstep
Learn to move	$\omega_h, \omega_{fs}, \omega_u + \omega_f$	+forward

After the first learning phase, we investigate the ideal reward weight composition of the remaining penalty terms, including instability, jerk, and muscle activation. Intuitively, the agent is expected to minimize these three penalty terms while still maintaining forward motion at a desired speed. We formulate the learning process as a constrained penalty maximization problem as shown in Algorithm 1. The learning process follows the goal-based CL scheme with the penalty weights being gradually increased as defined by $\boldsymbol{c}$ between the courses. The gradual changes in penalty weights facilitate a smooth transition between behaviours. After each course, the agent's episodic forward reward $\sum R_f$ is evaluated to ensure no degradation of walking capability beyond p which is set to be 5%. A significant drop in forward reward might indicate the policy convergence back to standing or early termination caused by falling that violates the constraint of maintaining forward motion.

Algorithm 1: Penalty Maximization With Primary Performance Constraints

Initialize: Pretrained agent π, V; target speed; initial weight vector $\boldsymbol{\omega}$; weight steps $\boldsymbol{c}$

for *reward term index* $i \in \{a, s, j\}$ **do**

  log initial forward performance $P_{\text{ini}} = \sum R_f$;

  while *True* **do**

    $\boldsymbol{\omega}_i := \boldsymbol{\omega}_i + \boldsymbol{c}_i$;

    Get π_{new}, V_{new} by optimizing π, V using PPO;

    if *converge* **then**

      log final performance $P_f = \sum R_f$;

      if $P_f > (1-p)P_i$ **then**

        $P_{ini} := P_f$, $\pi := \pi_{\text{new}}$, $V := V_{\text{new}}$;

      else

        $\boldsymbol{\omega}_i := \boldsymbol{\omega}_i - \boldsymbol{c}_i$;

        break;

      end

    end

  end

end

3.3 Gait Validation

Gait bilateral symmetry is important in achieving close-to-natural behaviour, but difficult to learn in the DRL setting [46, 47]. To validate the efficiency of learning bilaterally symmetric behaviour during phase one learning, the symmetry index (SI) is used [75] as shown below:

$$SI = \frac{X_R - X_L}{0.5(X_R + X_L)} \times 100\% \tag{3.10}$$

where X_R, X_L are the spatiotemporal parameters of the gait including step length and step time. In addition, We evaluate the gait symmetry of two additional comparative experiments without curriculum learning. One experiment directly uses the reward function from the last course of the phase one CL. The other experiment further simplifies the reward

by dropping the joint hyperextension penalty and the step reward terms. All the comparative experiments were run for the same number of time steps during all CL courses for consistency.

To evaluate the naturalness of the emerged gait after phase two learning, the agent gait is compared to the experimental overground gait data adopted from [67] and solutions from DC method. The DC method uses the same musculoskeletal model described in section 3.1 and generates optimal trajectory over half a gait cycle by minimizing muscle activation. We formulate the problem constraints to ensure gait symmetry and the same target average walking speed as the DRL agent. Pearson correlation analysis is then used to compare the agent gait, optimal gait from DC, and experimental overground gait data [67]. In addition, spatiotemporal gait parameters including stride length, walking speed, step time, swing time, and stance time are compared between the agent and human experimental data [54]. The walking speed and stride length are normalized as below:

$$\text{Normalized speed} = \frac{\text{speed}}{\sqrt{H * g}} \tag{3.11}$$

$$\text{Normalized stride length} = \frac{\text{stride length}}{H} \tag{3.12}$$

where H is the height of the musculoskeletal model which is set to $1.63m$, and g is $-9.807ms^{-2}$.

3.4 Results and Discussion

In this section, we first show the impact of our goal-based CL on generating symmetric behaviour based on the results from phase one learning. We then validate the agent's behaviour against existing solutions in the running scenario. Last, we discuss the fine-tuned walking gait patterns of the agent after the second phase learning and benchmark it against DC results and experimental human gait data.

3.4.1 Phase one - Initial Gait Learning

During phase one learning, the agent followed the goal-based CL scheme in Table 3.1 with a target speed of $1.2m/s$ and a max simulation time of $3s$ (300 time steps). Detailed weight

values for each course are listed in Table 3.2. All three courses were run for 1.5 million time steps which transferred to about 1.5 hours of training on a Ryzen 3600 6-core CPU with 16GB of RAM. After the first course, the agent successfully learned to stand with knee joint angles at -8.5 ± 10.2 degrees. After the second course, the agent learned to tap both feet alternately at high frequency. After the third course, the agent was able to walk at target speed with ~ 4 steps/s (see the supplementary video). This curriculum formulation significantly reduced the difficulty during initial policy search for close-to-symmetric gait.

Table 3.2: The reward weight values for phase one courses

	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7
Learn to stand	0	1	0	0	0	0	1
Learn to step	0	1	0	0	0	1	1
Learn to move	0	1	0	0	4	1	1

We ran two comparative experiments without CL as stated in section 3.3. For all three experiments, we collected 100 trajectories and calculated SI using mapped step length and step time for each trajectory. The step length was mapped using a logistic function with close to linear behaviour within the domain of $[-1, 1]$ to account for possible negative step length (foot-dragging behaviour). The signs of all SI values were corrected to account for different dominant legs across different agents. This ensured the positive mean values for step length and step time SIs. The SI values for all three agents were obtained and statistically compared using the Friedman test followed by Wilcoxon Rank Sum test with Bonferroni correction, which is summarized in Fig. 3.1. Both step length and step time showed group-level significant difference according to Friedman test ($p \ll 0.05$). The CL trial showed significantly lower SI magnitude compared to no CL trials for step length (Wilcoxon $p \ll 0.0083$) indicating more symmetric gait behaviour. Both no CL trials showed dragging foot behaviour with no significantly different step-length SI values (Wilcoxon $p = 0.61$). The SI values for step time are statistically different across all three agents (Wilcoxon $p \ll 0.0083$).

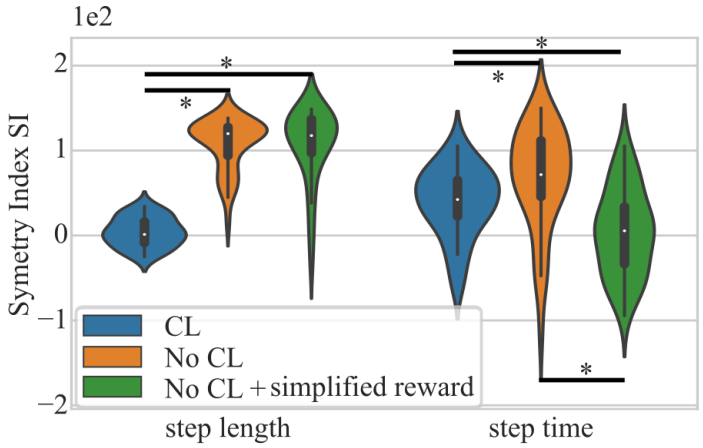

Figure 3.1: Evaluation of agent gait symmetry using SI; * indicates significant difference with $p < 0.0083$.

3.4.2 Learning to Run

To evaluate the agent capability in running tasks, we adopted the last reward function from phase one CL, but replaced R_f in (3.1) with $R'_f = v_{cx}$. We trained the agent for 20 million time steps with a max simulation time of $10s$ without reward function fine-tuning. The running gait generated by the agent is summarized in Fig. 3.2. Our learned solution showed good alignment with existing solutions from NeurIPS Learning to Run challenge and similarities with experimental data [45]. The reduced hip extension and ankle dorsiflexion of our solution relative to existing solutions were partially caused by the joint hyperextension penalty during phase one learning which was not given in NeurIPS Learning to Run challenge [45]. In addition, NeurIPS Learning to Run challenge required ground obstacle avoidance which encouraged hip extension and ankle dorsiflexion [45].

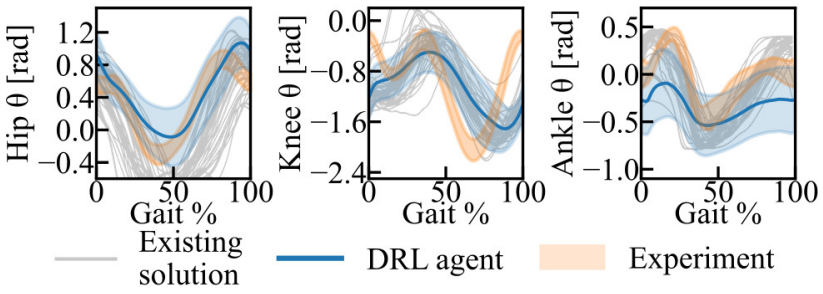

Figure 3.2: Comparison of the running gait of our DRL agent at 3.2m/s ± SD (in blue), the experimental data at 4m/s ± SD from [45] (in orange), and average gait of 47 existing solutions at 2.5-4.6m/s from [45] (in grey).

3.4.3 Phase Two - Gait Fine Tuning

Unlike running where realistic gait can emerge by simply maximizing forward distance per unit time [46], realistic walking at relatively low speed is more difficult to achieve due to many local optima [47]. In phase two learning, we fine tuned the walking-capable agent from phase one learning using Alg. 1 with a max simulation time of $10s$. We chose large weight update steps by setting the weight step vector to $\boldsymbol{c} = [5, 0, 10, 4, 0, 0, 0]$ in Alg. 1 and listed detailed weight values during each course in Table 3.3. This choice is to reduce the number of courses in Alg. 1 mainly due to computational limitations. The episodic reward plot along with the individual forward reward is summarized in Fig. 3.3. We demonstrated the the consistent walking capability of the agent even with gradually increased penalty in the reward function (see the supplementary video). The entire gait tuning process took less than 50 hours on a Ryzen 3600 6-core CPU. This is on par with off-policy methods, and more computationally efficient than other on-policy methods which often requires large CPU clusters with days of training [46, 47]. The final agent was tested 20 times for a total of 400 seconds of simulation time and walked without falling for all tests. The agent behaviour is shown in Fig. 3.4 which is close to human behaviour.

Table 3.3: The reward weight values for phase two courses

	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7
Iter. 1	0	1	10	0	4	1	1
Iter. 2	5	1	10	0	4	1	1
Iter. 3	10	1	10	0	4	1	1
Iter. 4	15	1	10	0	4	1	1
Iter. 5	20	1	10	0	4	1	1
Iter. 6	20	1	10	4	4	1	1
Iter. 7	20	1	10	8	4	1	1

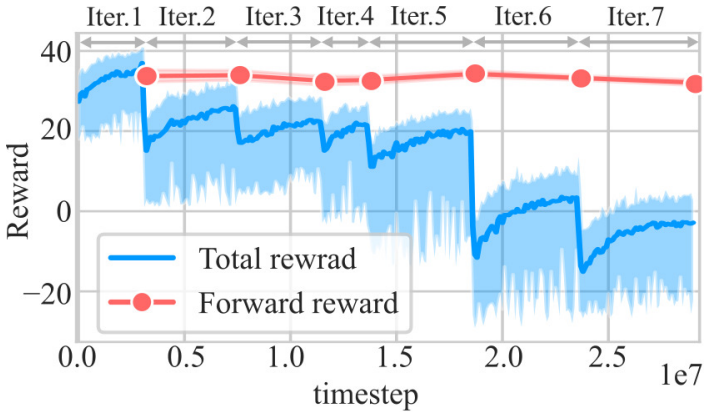

Figure 3.3: Reward plot of phase two learning consisting seven iterations; Iteration 1 penalizes on instability; Iteration 2-5 penalizes on muscle activation; Iteration 6-7 penalizes on jerk.

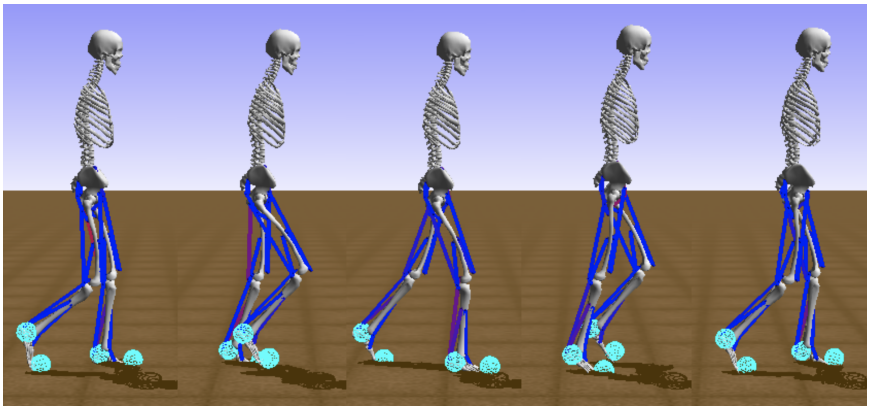

Figure 3.4: Behaviour after the phase two learning at $\sim$ 2 steps/s.

Observing close-to-natural agent walking behaviour, we then extracted kinematic data for 925 steps and analyzed the gait patterns in comparison to human gait trajectories as shown in Fig. 3.5. The extracted gait exhibits variations from stride to stride due to the stochasticity of the policy. However, the naturalness of such variation requires further analysis. The mean of the agent gait showed high correlation, greater than 0.92, with the optimal trajectory from DC at all three joints with a downward shift for the ankle angle plot. The agent gait also showed high correction with the experimental gait at hip and knee with slightly delayed knee extension during the swing phase (see the supplementary video). Despite the similarities, we identified two main differences between the agent and the experimental gait behaviour. First, the agent did not present enough knee extension before foot strike as shown in the middle plot in Fig. 3.5 which also occurred in the running case shown in Fig. 3.2. Second, the agent demonstrated toe-walking behaviour or early ankle plantar flexion prior to push-off as shown in the right plot in 3.5 which is similar to many other DRL-based musculoskeletal control experiments [47]. The agent spatiotemporal gait parameters were in the same range as human data as summarized in table 3.4, the agent showed similar walking speed and stride length to humans with slightly faster step frequency. The swing time and stance time percentage are also close to humans. No direct gait comparison to other DRL approaches for walking with musculoskeletal models [47, 83] is included since these studies either addressed different problems, or did not provide biological analysis of the gait.

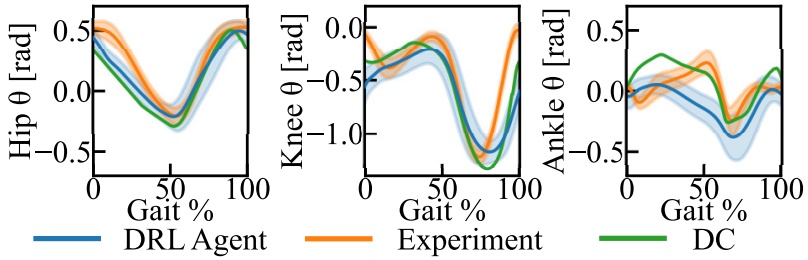

Figure 3.5: Agent gait patterns ± SD after the second phase learning compared to DC result and experimental gait patterns ± SD [67]. The Pearson correlation of hip, knee, and ankle joint angles are 0.99, 0.98, and 0.93 between agent and DC, and 0.95, 0.83, 0.39 between agent and experimental data.

Table 3.4: Comparison of spatiotemporal parameters between agent and human gait from [54]

	DRL agent	Human
Dimensionless stride length	0.60 ± 0.08	0.73 ± 0.09
Dimensionless walking speed	0.28 ± 0.04	0.27 ± 0.04
step time (s)	0.43 ± 0.06	0.56 ± 0.05
swing time (s)	$0.34 \pm 0.08(39\%)$	$0.45 \pm 0.03(40\%)$
stance time (s)	$0.53 \pm 0.08(61\%)$	$0.68 \pm 0.07(60\%)$

Based on the gait results from phase two learning, we believe there are two main factors that may contribute to the missing knee extension before foot strike and the toe walking behaviour of the agent. The first factor is the limitation of the OpenSim model. The constrained sagittal-plane simulation of the musculoskeletal model may limit the model flexibility in generating frontal plane motion which can be important to maintain foot-ground clearance. The toe-walking behaviour can be seen as the alternative strategy of keeping foot-ground clearance given the variation of gait from the stochastic policy. Additional experiments with reduced jerk penalty showed less ankle plantarflexion during walking at the expense of more delayed knee extension and more hip flexion to maintain foot-ground clearance. The predefined ground contact model may also influence the

emerged agent gait due to potentially inaccurate contact properties for the walking scenario. The second factor is the limitation from the DRL agent. After phase two learning, the muscle excitation command from the control policy is highly sparse. When coupled with the first-order activation dynamics [15], the sparse muscle excitation command could not reproduce relatively smooth muscle activation profiles that are usually seen from DC studies, which makes the fine-tuning of detailed agent behaviour difficult. In addition, the composition of the reward function in this study may not fully capture the rationale of walking. For example, the benefit of the energy dissipation during foot strike with the heel-sole-toe walking [91] is not considered in our analysis.

Chapter 4

Cost Function Optimization using IOC

Although CL-based reward function tuning is powerful in recovering the expert policy for close-to-natural walking behaviour in the DRL environment, it requires a long training time which makes subject-specific gait simulation difficult. In this chapter, we explored an alternative formulation of the cost function tuning problem using predictive simulation for individual participants. We first discuss the customization of the subject-specific musculoskeletal models, followed by the formulation of the inner-loop optimization using DC, and last the outer-loop AR-IOC algorithm inspired by SC and DLM [74, 59]. In this study, we first test our AR-IOC algorithm on synthetic walking data generated by a known cost function. Then, we perform IOC analysis on two treadmill walking datasets. The first dataset is publically available [57] which includes ten healthy participants walking at self-selected speeds (i.e., the most comfortable speeds for daily walking) and stride lengths. The second dataset is collected at the Neuromechanics and Assistive Robotics Lab (Arami Lab) at the University of Waterloo. This dataset includes five healthy subjects walking at four prescribed speeds and two different stride lengths. Note that a part of this chapter will appear soon in the IEEE Transactions on Neural Systems and Rehabilitation Engineering (J. Weng, E. Hashemi, and A. Arami. "Adaptive Reference Inverse Optimal Control for Natural Walking with Musculoskeletal Models," under review).

4.1 Subject Specific Musculoskeletal Models

Natural gait patterns vary across individuals due to their diverse body geometries, muscle fibre type proportions, strength, and possible variations in neural control. When performing cost function optimization for individual participants, a personalized musculoskeletal model is required to better represent the physical characteristics. Here, we first construct a baseline musculoskeletal model, then scale it to match the anthropometric parameters of each participant.

The baseline musculoskeletal model is customized based on the *gait10dof18musc* model from OpenSim [15] which is shown in Fig. 4.1. The model allows sagittal-plane (2D) motion with 11 DoF, including 8 DoF at the hip, knee, ankle, and metatarsophalangeal (toe) joints, and 3 DoF for global translation and rotation at the pelvis. The full system configuration is described using a 40-dimensional state vector, including the position and velocity of all 11 DoF and 18 muscle activation signals. The 18-dimensional control signals specify muscle excitation, which is linked to muscle activation using first-order activation dynamics [15]. Lower limb muscles include HAM, BFsh, GM, IL, RF, VA, GAS, SOL, and TA. Unlike other locomotion related IOC studies that uses full-body humanoid [11, 64, 24], this model ignores upper-limb segments such as arms and hands for simplicity. We choose this simplified model as a starting point for its computational-efficiency when solving the gait optimization problems discussed in section 4.2. The utilization of the full-body musculoskeletal model is remained as a future research direction. The hip, knee, and ankle joints are softly constrained using passive joint torque actuators with double exponent formulation [3] to simulate the effect of ligaments. The metatarsophalangeal joints are passively actuated using linear passive spring-damper actuators. The default muscle fibre lengths and pennation angles are obtained from [96]. The default tendon slack lengths are obtained from [66]. Six Hunt Crossley contact spheres at the heel, first metatarsal head (MH), third MH, fifth MH, hallux, and middle toe are added for each foot to cover all high-pressure zones during walking [69].

The subject-specific musculoskeletal models are obtained by scaling the baseline model. The body geometries, contact sphere locations, and optimal muscle fibre lengths are scaled based on the marker tracking data. The model mass is also scaled to the reported subject's body mass while maintaining the same mass distribution as the baseline model. The maximum isometric muscle force is calculated using the method explained in [66]. First, the subject's body mass and height are taken as input to estimate the individual muscle volumes using the experimental results from [34]. The muscle volumes are then used along with the scaled optimal muscle fibre lengths and the muscle-specific tension from [72] to estimate the maximum muscle isometric forces.

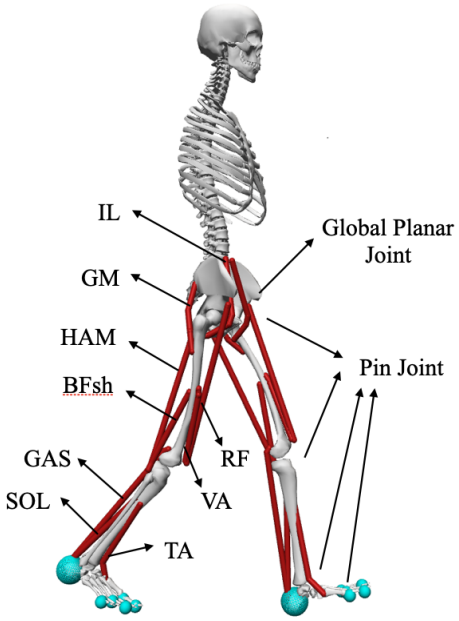

Figure 4.1: The musculoskeletal model has 11 DoFs and 18 muscles.

4.2 Optimization Formulation

4.2.1 Inner-loop Trajectory Optimization

The inner-loop trajectory optimization assumes a given cost function and finds the optimal trajectory that minimizes the given cost using DC. We used OpenSim Moco [16] to formulate the DC problem with musculoskeletal models. The inner-loop cost function for generating the walking motion consists of six terms with four variable weights:

$$
\begin{aligned}
J_{inner} =& \frac{1}{d}\int_0^{t_f} (\boldsymbol{\omega}^T \boldsymbol{C}) dt \\
=& \frac{1}{d}\int_0^{t_f} \Bigg[10 \sum_i \left(\frac{u_i^3}{n}\right) + 10^{-2} \sum_j \hat{\tau}_j^2 \\
&+ \omega_1 \times 10 (\theta_{torso})^2 \\
&+ \omega_2 \times (\hat{a}_{CoM})^2 \\
&+ \omega_3 \times 10^2 \left[\left(\hat{F}_{lf}^y \hat{v}_{lf}^{fx}\right)^2 + \left(\hat{F}_{rf}^y \hat{v}_{rf}^{fx}\right)^2 \right] \\
&+ \omega_4 \times 10 (\hat{p}_{CoMx} - \hat{p}_{midBoS})^2 \Bigg] dt,
\end{aligned} \tag{4.1}
$$

which can be seen as the integral of a dot product of the weight vector $\boldsymbol{\omega}$ and the cost vector $\boldsymbol{C}$. As experimental data from multiple participants is used in our study, non-dimensionless quantities including torque, acceleration, force, position, and velocity exhibit different ranges across the participants due to factors such as segment length and body mass (can be seen in Table 4.2). These quantities are scaled to the equivalent dimensionless forms [39] using the following equations:

$$
\hat{\tau} = \frac{\tau}{m g l_{leg}}, \quad \hat{F} = \frac{F}{mg}, \quad \hat{v} = \frac{v}{\sqrt{l_{leg} g}}, \quad \hat{p} = \frac{p}{l_{leg}}, \tag{4.2}
$$

where m is the subject mass, l_{leg} is the leg length, g is $9.81 m/s^2$. Such scaling improves the cost function generalization across different body sizes. The first term penalizes the cubed muscle excitation u where i is the muscle index, and n is the total number of muscles. This term has been widely used in previous studies to capture the CNS effort to minimize energy expenditure and muscle fatigue [23, 24, 90, 2, 67]. The second term penalizes the use of scaled passive ligament torque $\hat{\tau}_j$ similar to [90, 23, 24] to keep joint angles within the limits of the physiological range of motion, where j is the index of the passive joint actuators at hip, knee, and ankle joints. The third term penalizes the deviation of torso angle θ_{torso} from the upright position similar to [67] as an approximation of upper-limb energy expenditure. The fourth term penalizes the scaled acceleration of the CoM inspired by [24] to encourage motion smoothness. The fifth term penalizes the foot sliding similar to [64] to ensure proper foot clearance. This is evaluated based on the scaled vertical ground

reaction force (GRF) $\hat{F}^y$ and the scaled horizontal foot velocity $\hat{v}^{fx}$, where lf, rf represent left and right feet. The last term encourages the stability by penalizing deviation of the scaled forward CoM position $\hat{p}_{CoMx}$ from the center of the base of support (BoS) $\hat{p}_{midBoS}$ similar to [67]. Coefficients of the cost terms scale the cost values to the same order of magnitude given the desired motion trajectories from the experiment. After integration, the final cost is divided by the total forward distance d measured at the CoM. In addition, we also experimented metabolic cost based on [8] as an alternative to the muscle excitation cost. The weight of the metabolic cost is set to 0.35 which results in a similar cost value as the muscle activation term in (4.1) while maintaining other weights the same (i.e., all weights (4.1) are set to 1). The resultant optimal gait by using the metabolic cost is almost identical to the one by using the muscle excitation cost with mean joint angle deviation of 0.6, 1.4, and 1.3 degree and Pearson correlation of 0.999, 0.999, and 0.998 for hip, knee, and ankle joint angles, respectively. However, the metabolic calculation introduced 20% more computational time when solving the direct collocation problem. As a result, we only use muscle excitation penalty for higher computational efficiency.

The first two terms in the cost function are designed with fixed constant weights due to two reasons. First, uniformly scaling the multipliers of all cost terms does not change the optimal solution. To prevent uniform scaling, the muscle activation multiplier is set to a constant so that the weight updates ensure relative weight changes between cost terms and, consequently, changes in the solution. Second, the passive joint torque term is fixed to improve the stability of the optimization. During natural walking, the lower-limb joints rarely reach extreme angles resulting in small cost values (i.e., close to zero). This can cause the weight update to the passive joint torque term to be unreliable due to poor numerical precision. In some cases, the weight may converge to zero, which may cause the inner-loop optimization to generate abnormal walking with knee hyperextension and subsequently lead to a large rebound of the cost weight. Therefore, the weight of passive joint torque term is fixed to prevent its convergence to zero and ensure the stability of the optimization program.

To reduce the complexity of the DC problem, the optimized walking trajectory is limited to half a gait cycle with gait symmetry constraints. These symmetry constraints ensure mirrored state values (i.e., position and velocity of all DoF except the global forward distance, and all muscle activation signals) about the sagittal plane at the trajectory start and end. After solving the DC problem, the full gait cycle is reconstructed by concatenating the trajectories from the left and the right leg. Several equality and inequality constraints are imposed on the state and control variables. Muscle excitation and activation are constrained to $[0, 1]$. Joint angles and global body positions are constrained to feasible ranges (i.e., $[-15°, 60°]$, $[-70°, 5°]$, $[-15°, 25°]$, for hip, knee, and ankle angles,

respectively; [0m, 1m], [0.75m, 1.25m], [$-20°$, $20°$] for global forward distance, height, and orientation at pelvis, respectively). The average walking speed and the final trajectory time are constrained to the experimental values to maintain the same walking task. Last, the trajectories are discretized to only 15 nodes which are shown to be sufficient by [67] and our investigations.

4.2.2 Outer-loop Cost Function Tuning

The outer-loop optimization aims to optimize $\boldsymbol{\omega}$ in the cost function J_{inner} such that the inner-loop solution matches the reference trajectories from the experiments. With the inspiration from SC [74] and DLM [59], we formulate the AR-IOC for the outer-loop optimization shown in Alg. 2 and Fig. 4.2. AR-IOC consists of three main steps. First, the experimental gait data is collected and pre-processed to ensure dynamic feasibility. Next, two DC problems are solved in parallel with one generating predictive gait trajectories given the current cost weights, and the other one generating adaptive reference trajectories by adding experimental data tracking to the current cost function. Last, the converged trajectories from these two DC problems are compared to update the cost weights. Steps 2 and 3 are repeated until the convergence criteria are satisfied.

We first represent the walking task-related parameters (e.g., walking speed and stride time) as $\boldsymbol{x}$, and the walking trajectories (including both states and controls) as $\boldsymbol{y}$. If the cost weight $\boldsymbol{\omega}$ is optimal (i.e., $\boldsymbol{\omega} = \boldsymbol{\omega}^*$), then the cost of $\boldsymbol{y}_{ref}$ as the dot product of $\boldsymbol{\omega}$ and cost vector $\boldsymbol{C}$ is smaller than all other choices of $\boldsymbol{y} \in \mathcal{Y}$, where $\mathcal{Y}$ is the superset of all possible trajectories. such a relationship can be represented as:

$$\boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}_{ref}(\boldsymbol{x})) \leq \boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x})) + \zeta, \tag{4.3}$$

where $\zeta \geq 0$ (i.e., a slack variable) deals with possible violation of the inequality when $\boldsymbol{\omega}$ is sub-optimal. Intuitively, we want ζ to be small to ensure the optimality of $\boldsymbol{\omega}$. In addition, the following inequality holds for any arbitrary $\boldsymbol{\omega}$:

$$\boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}_{ref}(\boldsymbol{x})) \geq \min_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x})). \tag{4.4}$$

The inequality in (4.4) reduces to equality when $\boldsymbol{\omega}$ is optimal (i.e., $\boldsymbol{\omega} = \boldsymbol{\omega}^*$). As (4.3) holds for all $\boldsymbol{y}$, we can substitute $\boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x}))$ in (4.3) by its minimum over $\boldsymbol{y}$. We then obtain:

$$\zeta \geq \boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}_{ref}(\boldsymbol{x})) - \min_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x})) \geq 0. \tag{4.5}$$

Since we want small ζ, we have to minimize its lower bound. Therefore, the problem becomes:

$$\min_{\boldsymbol{\omega}} J_{outer} = \min_{\boldsymbol{\omega}} [(\boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}_{ref}(\boldsymbol{x})) - \min_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x}))]. \tag{4.6}$$

The gradient of the objective function with respect to $\boldsymbol{\omega}$ can then be calculated as follows:

$$\nabla J_{outer} = \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}_{ref}) - \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}') \tag{4.7}$$

$$\boldsymbol{y}' = \underset{\boldsymbol{y} \in \mathcal{Y}}{\text{argmin}} \boldsymbol{\omega}^T \boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}(\boldsymbol{x})), \tag{4.8}$$

where $\boldsymbol{y}_{ref}$ is the reference trajectory from experiments, $\boldsymbol{y}'$ is the optimal inner-loop DC solution given $\boldsymbol{\omega}$. As $\boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}_{ref})$ relies on clean control signals (muscle excitation) which are nearly impossible to collect during experiments, we estimate the control signals by solving a tracking DC problem using the cost function $\hat{J}_t$:

$$\hat{J}_t = J_{inner} + \frac{1}{d} \int_0^{t_f} c_t dt \tag{4.9}$$

$$c_t = \omega_k ||\boldsymbol{s}_k - \boldsymbol{s}_k^{ref}||_2^2 + \omega_f ||(\boldsymbol{s}_f - \boldsymbol{s}_f^{ref})||_2^2, \tag{4.10}$$

where $\boldsymbol{s}_k$, $\boldsymbol{s}_f$ are the kinematic and GRF trajectories to be optimized in this tracking DC problem, and $\boldsymbol{s}_k^{ref}$, $\boldsymbol{s}_f^{ref}$ are the reference trajectories from experiment. Weights ω_k and ω_f control the trade-off between the kinematic and the kinetic tracking which are set to 0.01 and 0.005 for all experiments to allow deviations from the tracking data. Such deviation allows the adaptive reference trajectory to be closer to the trajectories obtained using the current cost weights, which avoids large gradient updates and smooths out the gradient history. The difference between ω_k and ω_f ensures an approximately similar magnitude of the kinematic tracking cost and the ground reaction tracking cost for subjects. As J_{inner}

is a function of $\boldsymbol{\omega}$, $\hat{J}_t$ varies during the weight optimization. This requires the tracking DC problem to be re-solved after each weight update. Intuitively, we allow the estimated reference trajectory $\boldsymbol{y}_{ref}$ to adapt to the optimized weight vector while moving towards the reference motion. This modification nicely results in the DLM for structured prediction which was originally proposed in [59]. Consequently, we replace $\boldsymbol{C}(\boldsymbol{x}, \boldsymbol{y}_{ref})$ in (4.7) with $\boldsymbol{C}(\boldsymbol{x}, \hat{\boldsymbol{y}}_{ref})$ where $\hat{\boldsymbol{y}}_{ref}$ is the adaptive reference generated by the tracking DC problem using (4.9). Then gradient descent is used for the weight update. Comparing to using a static reference $\boldsymbol{y}_{ref}$ (further away from $\boldsymbol{y}'$), the adaptive reference $\hat{\boldsymbol{y}}_{ref}$ (closer to $\boldsymbol{y}'$) also reduces the inaccuracy in calculating the cost value differences in (4.7) due to numerical errors from the direct collocation problem (e.g., trajectory reconstruction errors), and improve the stability of the cost update. The resulting algorithm for learning optimal weights (that indicates the optimal cost function) for natural walking, called AR-IOC, is summarized in Alg. 2, and Fig. 4.2 shows the connection among different components of this algorithm.

Algorithm 2: AR-IOC

Initialize: desired trajectories $\boldsymbol{s}_k^{ref}$, $\boldsymbol{s}_f^{ref}$ from solving DC problems using J_t;
initial weight vector $\boldsymbol{\omega}$; learning rate η

while *max iter. not reached* **do**

$\hat{\boldsymbol{s}}_{ref}, \hat{\boldsymbol{u}}_{ref} = \underset{\boldsymbol{s},\boldsymbol{u}}{\operatorname{argmin}} \hat{J}_t(\boldsymbol{\omega})$;

compute cost vector $\hat{\boldsymbol{C}}_{ref} = \boldsymbol{C}(\hat{\boldsymbol{s}}_{ref}, \hat{\boldsymbol{u}}_{ref})$;

$\boldsymbol{s}', \boldsymbol{u}' = \underset{\boldsymbol{s},\boldsymbol{u}}{\operatorname{argmin}} J_{inner}(\boldsymbol{\omega})$;

compute cost vector $\boldsymbol{C}' = \boldsymbol{C}(\boldsymbol{s}', \boldsymbol{u}')$;

$\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} + \eta(\boldsymbol{C}' - \hat{\boldsymbol{C}}_{ref})$;

end

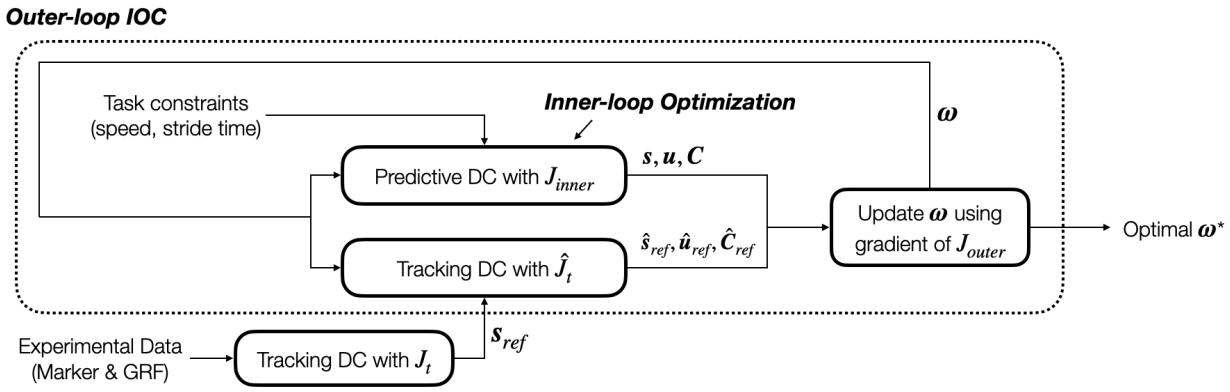

Figure 4.2: Block diagram of the AR-IOC algorithm. The tracking DC problem outside of the outer-loop IOC module generates feasible trajectories $\boldsymbol{s}_{ref}$. The inner-loop predictive DC problem generates state trajectories $\boldsymbol{s}$, control trajectories $\boldsymbol{u}$, and the cost vector $\boldsymbol{C}$ subject to stride time and average speed constraints, as well as state and control bounds. The tracking DC problem inside the outer-loop IOC module provides adaptive reference trajectories for the weight update. All DC problems are solved with the scaled musculoskeletal models using methods described in 4.1.

In actual experiments, the experimental data $\boldsymbol{s}_k^{ref}$, $\boldsymbol{s}_f^{ref}$ in (4.10) is pre-processed by solving a tracking DC problem with the cost function J_t:

$$J_t = \frac{1}{d}\int_0^{t_f} \left(c_t + ||\boldsymbol{u}||_2^2\right) dt. \tag{4.11}$$

This is to convert treadmill walking to level ground walking in simulation and ensure dynamic feasibility of the reference trajectory. The simple gradient descent update rule is replaced with the momentum update to accelerate convergence. The learning rate η is tuned using grid search during the experiment with synthetic data (explained in section 4.1) to achieve stable and efficient convergence. The tuned η is then applied to all other experiments. A dynamically adjusted weight lower bound (set to 1/10 of the initial weights, and reduced by half if the bound is activated for more than 5 consecutive iterations) and gradient clipping are also used to avoid the weight speedy drop to zero due to the momentum update which can cause significantly different inner-loop DC solutions and outer-loop weight oscillations.

4.3 Results and Discussion

4.3.1 IOC on Synthetic Data

To test the convergence of the proposed algorithm, we used a synthetic reference trajectory by solving a DC problem with known weights in (4.1). The DC problem used the scaled musculoskeletal model for the first participant in the dataset [57]. The walking task is configured to be at 1.2m/s with a stride frequency of 1Hz. The synthetic known weights in (4.1) are all configured to 0.5 for demonstration. To test the robustness of the AR-IOC algorithm against the weight initialization, we used three different sets of initial weights to represent different initialization conditions (i.e., smaller / larger than the known weights, and mix of both) as shown in table 4.1. All three trials were run in parallel and converged after 100 weight updates which took about 3 hours of total computation using AMD Ryzen 7 3700X CPU at 3.6GHz. The final converged weights and the weight errors are shown in Table 4.1 and Fig. 4.3, respectively. The final converged weights are similar to the desired values for all three trials, and the weight error progress showed a nice reduction over the inner-loop evaluations. Trial 1 and 2 required more inner-loop updates as the initial weights are further away from the reference weights. Some oscillations exist in the weight error plot which are primarily caused by the weight overshoot from the momentum update. The final weight errors stabilized slightly above 0 after iteration 80 for all three trials. This is because the adaptive reference trajectories $\hat{\boldsymbol{s}}_{ref}, \hat{\boldsymbol{u}}_{ref}$ in Fig. 4.2 allows small deviation from the reference trajectories in exchange for a slightly easier outer-loop optimization problem. This deviation is small and can be neglected as shown in Fig. 4.4.

Table 4.1: Weight comparison after solving the synthetic IOC problem

	Initial Weights $[\omega_1, \omega_2, \omega_3, \omega_4]$	IOC solutions $[\omega_1, \omega_2, \omega_3, \omega_4]$
Trial 1	[2.5, 2.5, 2.5, 2.5]	[0.57, 0.49, 0.54, 0.46]
Trial 2	[0.1, 0.1, 0.1, 0.1]	[0.56, 0.48, 0.55, 0.46]
Trial 3	[0.1, 0.1, 2.5, 2.5]	[0.57, 0.49, 0.55, 0.46]
Desired weights		[0.50, 0.50, 0.50, 0.50]

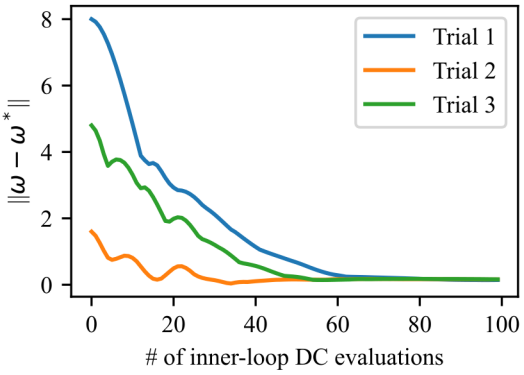

Figure 4.3: Weight vector error reduction plot using synthetic reference trajectories. Initial and final weights for all three trials are listed in Talbe 4.1.

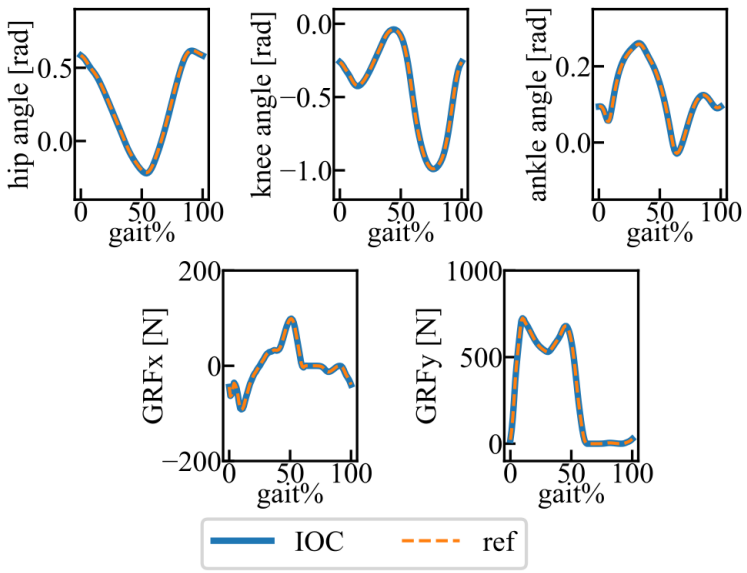

Figure 4.4: Comparison of gait kinematics and GRFs between the IOC solution and the synthetic reference gait.

4.3.2 IOC on Walking data with Self-selected Speed

In this section, treadmill walking data of ten healthy participants [57] were used. The initial weight vector $[\omega_1, \omega_2, \omega_3, \omega_4]$ in (4.1)is set to [1, 1, 1, 1] for all subjects. The IOC problems were solved with 193 weight updates on average (i.e., about 10 hours of computation per problem using AMD Ryzen 7 3700X CPU at 3.6GHz). Comparing to other derivative-free methods for cost function determination with similar setup [67], the AR-IOC significantly reduces the number of inner-loop evaluations. The subject information, walking configurations, and the optimized weights are summarized in Table 4.2. The converged CoM acceleration weights ω_2 (0.605 ± 0.325, mean $\pm$ standard deviation (SD) computed across subjects), foot sliding weights ω_3 (1.175 ± 0.460), and stability weights ω_4 (0.337 ± 0.175) are considerably larger than zero for all participants. This behaviour can indicate the significance of these cost terms in capturing natural walking behaviour. In contrast, the torso orientation weights ω_1 for five participants (participant 2, 4, 5, 9, and 10) converge to small values (either 0.001, 0.01, or 0.027). This may be due to poor estimation of the desired torso orientation in the reference trajectories. As the motion tracking data only includes lower limbs, the estimation of the torso orientation solely depends on reconstructed pelvis orientation, which can have excessive forward tilt due to marker placement errors. Such inaccuracy leads to a large torso orientation cost for the reference trajectories, which essentially pushes ω_1 to zero. In addition, potential redundancy in the cost function may also cause some cost weights to converge to zero.

Given the optimized weights in Table 4.2, the optimal joint kinematic and GRF trajectories are computed and compared to the reference trajectories. Fig. 4.5 shows the gait comparison of a typical participant (participant 1 in [57]). Overall, the joint angles and GRFs from IOC match closely with the tracking solution. The ankle joint angles from IOC have a smoother profile than the reference during the stance phase.

Table 4.2: Subject mass, height, leg length, speed, stride time and optimal weights by solving IOC problems.

#	M kg	h m	l_{leg} m	v m/s	t_s s	ω_1	ω_2	ω_3	ω_4
1	64.9	1.71	0.87	1.31	1.08	0.112	0.241	1.634	0.589
2	57.7	1.58	0.82	1.26	0.99	0.010	0.329	1.638	0.606
3	70.7	1.77	0.94	1.05	1.30	0.521	0.142	1.466	0.088
4	70.2	1.77	0.95	1.10	1.18	0.001	0.514	0.757	0.202
5	79.2	1.79	0.97	1.05	1.17	0.027	0.787	1.180	0.330
6	73.5	1.78	0.98	1.16	1.22	0.714	0.584	0.717	0.445
7	73.0	1.61	0.86	0.95	1.06	0.788	0.845	1.116	0.226
8	73.1	1.76	0.93	1.05	1.26	0.988	1.077	0.958	0.324
9	94.3	1.85	1.02	1.00	1.25	0.001	1.045	1.833	0.160
10	59.5	1.54	0.79	1.10	1.08	0.010	0.488	0.452	0.404

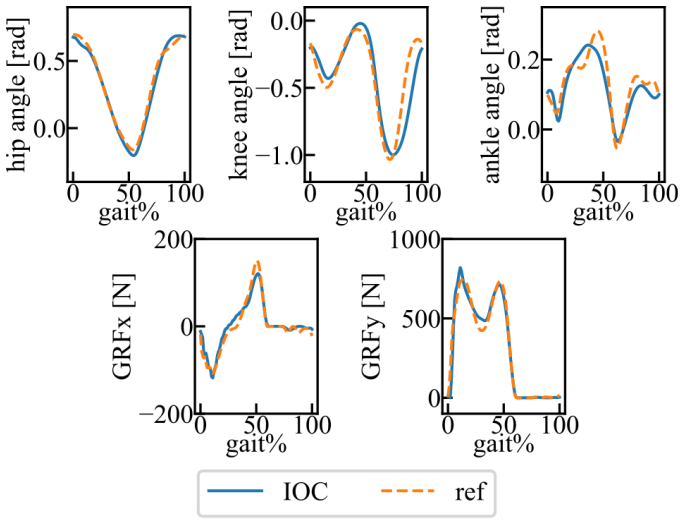

Figure 4.5: Comparison of joint angles and GRFs between the IOC solution and the experimental reference gait from the first participant in [57]; GRFx and GRFy are along the forward and vertical directions, respectively.

Fig. 4.6 shows the muscle activation plots obtained from the optimal cost function (through IOC) of the same participant. The EMG signals of four lower-limb muscles, which were measured and available in [57], are overlaid in Fig. 4.6. The EMG signals are scaled to best match the magnitude of the muscle activation from the optimal cost function for visualization. The EMG signals of HAM, GAS, and TA showed good alignment to the obtained muscle activation trajectories. The EMG signal of RF shows good alignment to the combined muscle activation from two knee extensors (i.e., RF and VA). There may be two explanations for the aforementioned observations. First, surface EMG sensors only measure lumped muscle action potential at the location of measurement which can cause inaccuracies in individual muscle measurement when sensor placement is sub-optimal. Second, the individual muscle weights in the muscle activation penalty term may not be optimal, which may favour activating VA over RF. The obtained GAS activation is slightly delayed compared to the EMG signal similar to [67]. This may be caused by the problem reduction from 3D to the sagittal plane where delayed toe push-off is required to facilitate the delayed knee extension for proper foot clearance.

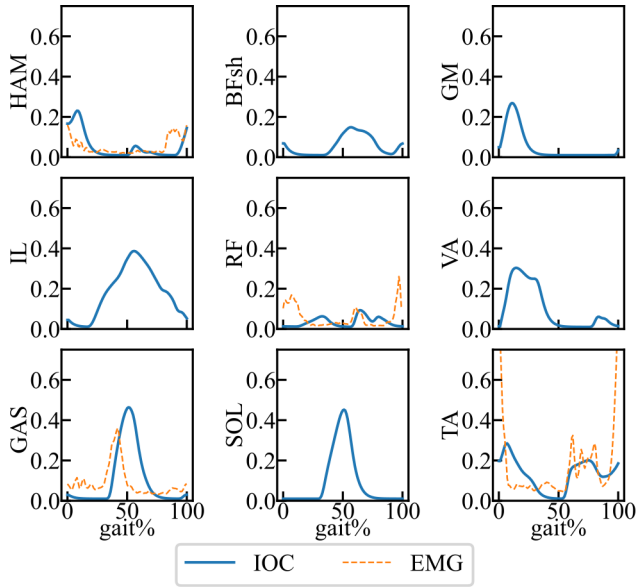

Figure 4.6: Comparison of muscle activations from IOC and average EMG signals of four measured muscles of the first participant in [57]. EMG signals for the other five muscles were not measured in [57].

To analyze the gait pattern quality, gait errors are computed with respect to the subject-specific reference trajectories using root mean square errors (RMSE). After IOC weight

optimization, hip, knee and ankle RMSEs of $0.041 \pm 0.007, 0.122 \pm 0.017, 0.054 \pm 0.009$ in radians, and GRFx and GRFy RMSEs of 0.020 ± 0.004, 0.064 ± 0.013 in terms of body weight ratio (i.e., GRF error [N] / body weight [N]) were obtained, respectively. The hip and ankle joint angles generally align well with the reference after weight optimization. The knee angles have relatively larger errors due to the delayed knee extension (visible in Fig. 4.5). The ground reaction forces also align well with the reference. RMSEs of the trajectories obtained from the initial weights are also computed to confirm the gait improvement from the weight optimization, which is summarized in Fig. 4.7. Such comparison indicates that the improvement mainly happens in kinematics features, especially hip and knee angles. GRF profiles start with small errors and show mild improvement from the weight modification. The small initial errors for the GRF are potentially due to the constraints on the task parameters (i.e., walking speed and stride length) during the IOC experiments (walking speed and stride length can impact GRF features like peak height and stance duration). In addition, the optimized gait patterns have Pearson correlations of 0.992 ± 0.004, 0.917 ± 0.024, 0.843 ± 0.077 for hip, knee, ankle angles, and 0.957 ± 0.024, 0.990 ± 0.003 for GRFx and GRFy, respectively.

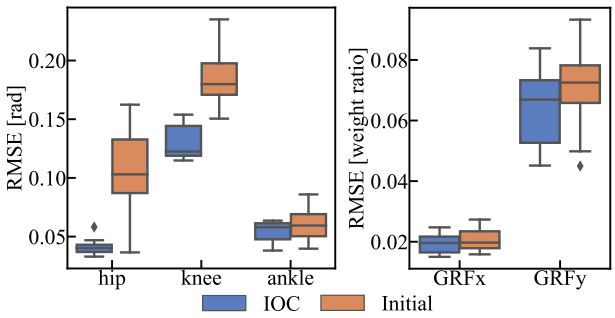

Figure 4.7: RMSE of the joint angles and GRFs before and after solving the IOC problems; GRFx and GRFy are along the forward and vertical directions, respectively.

Last, a correlation analysis is performed between the optimized weights and parameters, including body mass, walking speed, and stride time (raw data available in Table 4.2). As the CNS weighting of different cost terms may vary for different participants (e.g. mass, height, muscle properties) and walking tasks (e.g. speed, stride time), this correlation analysis could bring insights into how the cost weights vary as a function of those parameters. This is valuable particularly when we need to infer an optimal cost function for a walking task when we do not have access to its specific reference data. As the correlation is calculated across participants that have various body geometries, the walking speed and

the stride time are scaled to their dimensionless forms using the following equations from [39]:

$$\hat{v} = \frac{v}{\sqrt{l_{leg} g}}, \quad \hat{t}_s = \frac{t_s}{\sqrt{l_{leg}/g}} \tag{4.12}$$

where l_{leg} is the leg length, g is $9.81 m/s^2$. The Pearson correlation coefficients are summarized in Table 4.3. In terms of participant parameters, the body mass is negatively correlated to the scaled walking speed meaning slower walking for participants with larger body mass. The stride time is also negatively correlated to the scaled walking speed. In terms of weights, the torso orientation weights ω_1 and the foot sliding weights ω_3 show low correlations to the resultant scaled walking speed and stride time. The CoM acceleration weight ω_2 has a negative correlation with the scaled walking speed suggesting less demand on the smooth motion at higher speeds. The stability weight has a positive correlation with the scaled walking speed, indicating more penalty to stability with faster walking. However, such an observation is not sufficient to draw a concrete conclusion as the correlation is computed based on only ten participants walking with limited gait speed and stride length variations. Extending this study to include variable gait data of each participant, which is covered in section 4.3.4, would better illustrate the existence of such speed-cost relationships.

Table 4.3: Pearson correlation coefficients between scaled walking parameters and weights.

	M	$\hat{v}$	$\hat{t}_s$	ω_1	ω_2	ω_3	ω_4
M	1	-0.78	0.34	0.07	0.68	0.31	-0.60
$\hat{v}$	-0.78	1	-0.52	-0.35	-0.67	0.10	0.88
$\hat{t}_s$	0.34	-0.52	1	0.43	0.13	-0.17	-0.63

One main challenge in using the proposed IOC framework to solve optimal locomotion cost weights is the stability of the inner-loop DC problems. DC transcripts the trajectory optimization problem (i.e., in the form of differential equations) into constrained parameter optimization problems with algebraic equations by discretizing the trajectories and reconstructing them using function approximations. The discretized optimization program is then solved by a non-linear solver such as IPOPT [94]. Such an optimization technique does not guarantee convergence to the global optima and is often sensitive to the initial

trajectory guesses, especially for high-dimensional complex systems such as musculoskeletal models. We found that DC problems sometimes converge poorly if the initial trajectory guesses are infeasible (e.g. the musculoskeletal model sinks into the solid ground) or far off from the walking motion (e.g. jumping motion is given with two legs moving together which also satisfies symmetry constraints). We also observed that the optimized trajectories alternate between two configurations during IOC (e.g. with and without extensive knee extension during the stance phase). This indicates that the slightly changing weights cause the DC problem to converge to different local optima, which can cause oscillations in the outer-loop weight updates. This issue was mitigated by imposing proper coordinate constraints as explained in section 4.2.1, fixing the weight to the passive joint torque term in J_{inner}, and using initial trajectories with feasible walking motion. However, it is impossible to consider all edge cases that can lead to inner-loop DC instability. In addition, improving the expressive capacity of the defined cost function may improve the inner-loop DC stability. This improvement essentially worsens the other local optima and expand the convex region around the desired motion.

4.3.3 IOC-GA Performance Comparison

To compare the AR-IOC algorithm with derivative-free optimization methods in optimizing cost weights for natural walking, we solve the IOC problems for all ten participants using the GA implementation in MATLAB R2021a and compare the performance in Table 4.4. The bounds for weight sampling are set to $[0, 4]$. A population with a size of 40 is generated with random initial weights uniformly sampled within the weight bounds. The fitness function for the GA explicitly minimizes the sum of tracking errors for hip angle, knee angle, ankle angle, GRFx and GRFy. The converged weights from GA are generally larger than the ones from AR-IOC except for ω_1 (AR-IOC exhibits significantly larger ω_1 for subjects 6, 7, and 8). When excluding these three subjects, the Pearson correlation between the obtained weight vectors is 0.930 ± 0.058 (0.731 ± 0.329 when including all subjects). The high Pearson correlation coefficients for seven subjects indicate that the ratio between the four cost weights contributes more to the final converged gait than the weight magnitudes. The small differences in joint angle errors and Pearson correlations between the AR-IOC and GA for subjects 6, 7, 8 suggest that our cost function formulation may have redundancy which allows similar converged gait with different cost weights.

Table 4.4: Comparison between AR-IOC and GA

	AR-IOC (mean±SD)	GA (mean±SD)
ω_1	0.317 ± 0.392	0.076 ± 0.068
ω_2	0.605 ± 0.325	1.200 ± 0.918
ω_3	1.175 ± 0.460	2.142 ± 0.893
ω_4	0.337 ± 0.175	0.707 ± 0.290
e_{hip}[rad]	0.041 ± 0.007	0.042 ± 0.020
e_{knee}[rad]	0.122 ± 0.017	0.114 ± 0.010
e_{ankle}[rad]	0.054 ± 0.009	0.056 ± 0.010
e_{GRFx} [weight ratio]	0.020 ± 0.004	0.018 ± 0.010
e_{GRFy} [weight ratio]	0.064 ± 0.013	0.064 ± 0.010
r_{hip}	0.992 ± 0.004	0.993 ± 0.002
r_{knee}	0.917 ± 0.024	0.938 ± 0.011
r_{ankle}	0.843 ± 0.077	0.815 ± 0.070
r_{GRFx}	0.957 ± 0.024	0.965 ± 0.021
r_{GRFy}	0.990 ± 0.003	0.991 ± 0.003
# of DC eval.	385 ± 157	3124 ± 941
computing time [hr]	10 ± 4	69 ± 21

Despite the differences in weights, both methods produce gait patterns similar to the reference trajectories with no statistical differences for most errors and correlation measures between the AR-IOC and GA ($p > 0.05$ using the Wilcoxon Signed Rank test) except r_{knee} (GA outperforms AR-IOC with $p = 0.006$) and r_{ankle} (AR-IOC outperforms GA with $p = 0.012$). Although e_{GRFx} and r_{GRFx} showed statistical differences ($p = 0.001$ for both cases), such differences can be neglected as the mean difference is small (i.e., less than 1%). The subtle differences in converged gait between GA and the AR-IOC can be caused by their different ways of gait evaluation as GA considers joint angle and GRF errors directly, whereas the AR-IOC considers the cost values which is the abstraction of the original gait. The AR-IOC only requires about 12% of the internal DC evaluations and 14% of the computing time compared to the GA. Besides the computational improvement, the

proposed algorithm, unlike GA, has no requirement on the initial bounds for weight vector sampling.

4.3.4 IOC on Walking data with Prescribed Speed

The experimental data used in section 4.3.2 lacks variation in walking speed and stride length which makes it not an ideal choice for analyzing the correlation between the walking tasks and the cost weights. To address this issue, We first collected a dataset of natural walking at different speeds and stride lengths. During the experiment, body kinematics, GRF, and EMG data are collected synchronously using the Vicon Vero motion capture system (total of eight cameras sampled at 100Hz), Bertec split belt instrumented treadmill (sampled at 1000Hz), and Delsys EMG sensor system (sampled at 2000Hz), respectively. Fig. 4.8 shows an example of the marker and EMG sensor placement on the human body. The marker placement is modified based on the Plugin-gait lower limb convention (Vicon, UK) with sixteen lower-limb markers. Four pelvis markers are placed over the anterior and posterior superior iliac spine. Four markers are placed over the thigh and shank. Two knee markers are placed on the lateral epicondyle. Two ankle markers are placed on the lateral malleolus. Four foot markers are placed on the calcaneous and the second MH. Two additional shoulder markers on top of the Acromio-clavicular joint are included for torso orientation tracking. This is to address the issues from the previous experiment in section 4.3.2 that using only lower-limb markers causes inaccuracy in torso orientation estimation. Fourteen EMG sensors are placed on major lower-limb muscles for sagittal plane motion (i.e., GM, RF, Vastus Lateralis (VAl), TA, SOL, and Gastrocnemius Medialis (GASm)) to record muscle activities. The exact sensor locations and orientations follow the conventions explained in the Surface Electromyography for the Non-Invasive Assessment of Muscles (SENIAM) project [37].

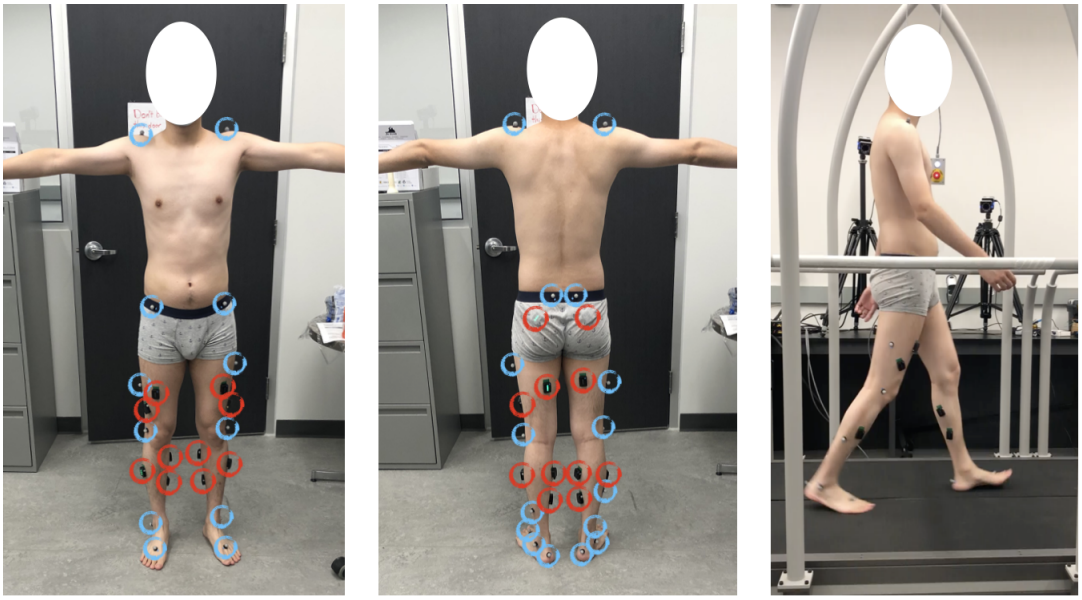

Figure 4.8: Examples illustrating the sensor and marker placement for natural walking data collection; the left and the middle images show EMG sensor (circled in red) and marker (circled in blue) placement on the human body; the right image shows a participant walking on the instrumented treadmill.

The experiment comprises of two phases: the calibration phase and the data collection phase. During the calibration phase, the participant is instructed to stand on the treadmill at an upright pose. The marker and GRF data is collected to provide a reference of the participant's body geometries and weight. A squat-to-toe-standing motion is performed to set the reference for maximum muscle contraction. During the data collection phase, each participant is instructed to walk on the treadmill at eight different configurations that cover four different walking speeds and two different stride lengths (summarized in Table 4.5). The walking speed is represented in the scaled form ($v = \hat{v} \times \sqrt{l_{leg} g}$) to account for different body geometries [39], where $v, \hat{v}$ are the actual and scaled speed, respectively, l_{leg} is the leg length, and g is $9.81 m/s^2$. The trials with slower speeds are performed first to avoid sweat accumulation (often under the EMG sensor during high-speed walking) which can influence the accuracy of the EMG measurement. The normal stride length is self-selected by the participants at their most comfortable gait, whereas the longer stride length requires the participants to actively enlarge the steps. The amount of the step increases relative to the normal steps is not controlled due to its practical challenges during the experiments. The first trial has four minutes of acclimatization period (i.e., the period

where the subject is walking, but the collected data is ignored) for proper adaptation to the treadmill walking according to [60]. The later trials reduce the acclimatization is reduced to one minute to reduce the overall experimental time to avoid muscle fatigue. For each trial, two minutes of walking data are used for processing. Between each trial, half a minute break is given for the participants to rest as well as for the experimenter to update the equipment configurations. The participant's gender, height, leg length, mass, and four actual walking speeds are summarized in Table 4.6.

Table 4.5: Experimental setup during the data collection phase

Trial	scaled speed	stride length	acclimatization time (min)	walking time (min)	rest time (min)
1	0.2	normal	4	2	0.5
2	0.2	longer	1	2	0.5
3	0.3	normal	1	2	0.5
4	0.3	longer	1	2	0.5
5	0.4	normal	1	2	0.5
6	0.4	longer	1	2	0.5
7	0.5	normal	1	2	0.5
8	0.5	longer	1	2	0.5

Table 4.6: Subject information and walking speeds for natural gait data collection

subject	gender	Height (m)	Mass (kg)	Leg length (m)	speed (m/s)
1	male	1.80	65.73	0.97	[0.62, 0.93, 1.23, 1.53]
2	female	1.69	65.51	0.89	[0.59, 0.88, 1.18, 1.47]
3	male	1.79	106.95	0.93	[0.60, 0.91, 1.21, 1.51]
4	male	1.83	87.80	0.94	[0.61, 0.92, 1.23, 1.54]
5	female	1.61	58.83	0.84	[0.57, 0.86, 1.15, 1.44]

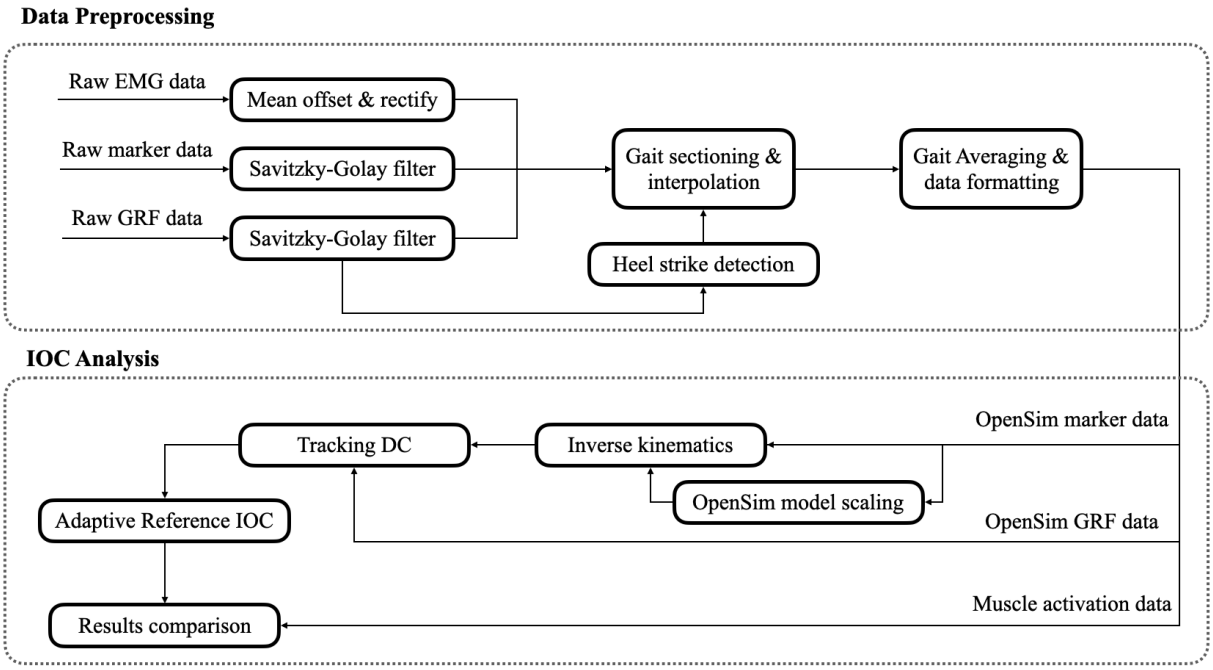

Figure 4.9: The pipeline of natural walking data analysis from data preprocessing to IOC. The raw EMG, marker, and GRF data were sampled at 2000Hz, 100Hz, and 1000Hz respectively. The interpolated gait data is discretized to 100 points for one gait cycle.

The collected marker and GRF data require pre-processing before being used in the IOC analysis. The pipeline for data analysis is summarized and shown in the top portion of the Fig. 4.9. First, Savitzky-Golay filters are used on the marker and GRF data to reduce noise. Then, individual gait data is split into gait cycles by identifying heel strikes using the smoothed vertical GRF data with the threshold of 50 N. Then, the individual gait trajectories are interpolated to a fixed length. Finally, the interpolated data are averaged and formatted to be used in the OpenSim environment.

To solve the IOC problem, the subject-specific musculoskeletal models are constructed. The baseline musculoskeletal model is mostly identical to the one used in section 4.1 but with one additional lumbar joint which allows rotation between the torso and the pelvis. An ideal torque actuator with 200Nm of maximum torque is added at the lumbar joint (similar to the OpenSim sample model formulation) to simulate the actuation from the upper body. Same body scaling procedures are performed as explained in section 4.1 to ensure accurate estimation of the participant's body geometries and muscle strengths. To further improve the generalization of the cost function (i.e., the torso orientation cost

in (4.1) is sensitive to the marker errors from the experimental data), the torso orientation penalty in (4.1) is replaced with the torso effort penalty as shown in (4.13). The torso torque is scaled to its dimensionless form using (4.2).

$$
\begin{aligned}
J_{inner_{mod}} =& \frac{1}{d}\int_0^{t_f} (\boldsymbol{\omega}^T \boldsymbol{C}) dt \\
=& \frac{1}{d}\int_0^{t_f} \Bigg[10 \sum_i \left(\frac{u_i^3}{n}\right) + 10^{-2} \sum_j \hat{\tau}_j^2 \\
& + \omega_{torso_{ef}} \times 10^{10} (\hat{\tau}_{torso})^2 \\
& + \omega_{CoM} \times (\hat{a}_{CoM})^2 \\
& + \omega_{slide} \times 10^2 \left[\left(\hat{F}_{lf}^y \hat{v}_{lf}^{fx}\right)^2 + \left(\hat{F}_{rf}^y \hat{v}_{rf}^{fx}\right)^2 \right] \\
& + \omega_{stability} \times 10 (\hat{p}_{CoMx} - \hat{p}_{midBoS})^2 \Bigg] dt,
\end{aligned}
\tag{4.13}
$$

After the modification to the cost function, a total of 40 IOC problems (i.e., four speeds and two stride length conditions for five subjects) are solved using the AR-IOC. The values of the final cost weights under different walking speed and stride length conditions are summarized in Fig. 4.10. Several observations can be made from the normal walking cost weights. First, both the CoM acceleration weights and the stability weights show upward trends with respect to the walking speed. On the other hand, the foot sliding weights reduce when the walking speed increases. The torso effort weights show no significant change with respect to the walking speed. For walking with longer strides, the cost weight trends show similarities to the ones from normal walking, but with a larger spread in weight values which may be potentially due to the uncontrolled amount of stride length increase during the experiments. The Pearson correlation between the final weight values for all participants (i.e., ω_{CoM}, ω_{slide}, $\omega_{torso_{ef}}$, and $\omega_{stability}$) and the walking task parameters (i.e., scaled walking speed and stride time) are summarized in Table 4.7 for normal stride lengths and Table 4.8 for longer stride lengths.

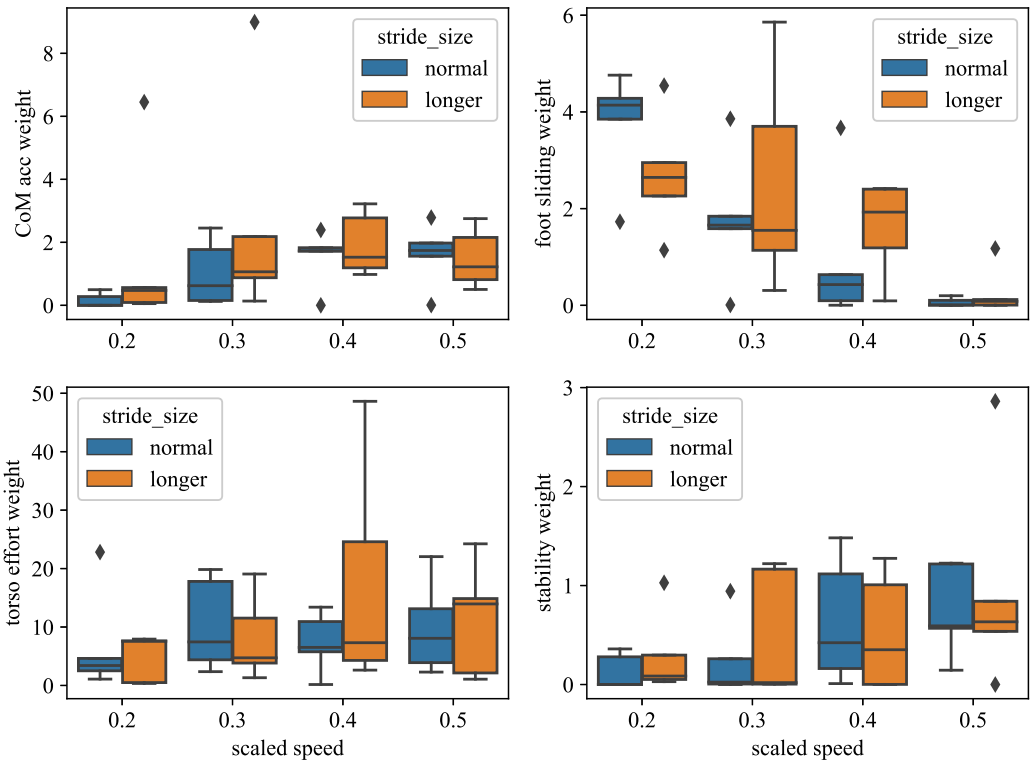

Figure 4.10: Comparison between cost weights and scaled walking speed for both normal and longer stride lengths.

Table 4.7: Pearson correlation coefficients between scaled walking parameters and cost weights for five subjects walking with normal stride lengths.

	$\hat{v}$	$\hat{t}_s$	ω_{CoM}	ω_{slide}	$\omega_{torso_{ef}}$	$\omega_{stability}$
$\hat{v}$	1	-0.90	0.57	-0.75	-0.03	0.53
$\hat{t}_s$	-0.90	1	-0.44	0.75	0.046	-0.41

Table 4.8: Pearson correlation coefficients between scaled walking parameters and cost weights for five subjects walking with longer stride lengths.

	$\hat{v}$	$\hat{t}_s$	ω_{CoM}	ω_{slide}	$\omega_{torso_{ef}}$	$\omega_{stability}$
$\hat{v}$	1	-0.94	-0.04	-0.58	0.28	0.33
$\hat{t}_s$	-0.94	1	0.01	0.59	-0.35	-0.29

For both normal and longer stride lengths, the torso effort weight $\omega_{torso_{ef}}$ shows a relatively small correlation to the walking speed and stride time. As the muscle excitation cost term is fixed in (4.13), it indicates that there is no significant shift in effort minimization between the lower and the upper bodies. The foot sliding weight ω_{slide} shows a large negative correlation with respect to the scaled velocity for both normal and longer stride lengths. This behaviour aligns with the experimental observation that participants' feet slided against the treadmill during swing more often at higher walking speeds. As the walking speed increases, larger steps are required which effectively reduces the height of the CoM during walking. To maintain ground clearance during foot swing, more knee flexion is required which is against the muscle excitation cost term. As a result, participants reduce the foot sliding weight and tolerate more foot sliding during the swing phase. For normal walking, both the CoM acceleration weight ω_{CoM} and the stability weight $\omega_{stability}$ have positive correlations with respect to the scaled walking speed. As the muscle excitation cost term is fixed in 4.13, increases in other weight (i.e., ω_{CoM} and $\omega_{stability}$) indicate an equivalent reduction in the weight of muscle excitation. This aligns with the intuition as more energy is required for higher-speed walking, and consequently a reduction in weights for the energy-related terms (i.e., the muscle excitation term and the torso effort term). Different from normal walking, the CoM acceleration weight ω_{CoM} for longer stride lengths shows no correlation to the walking speed. This is evident from the top-left plot in Fig. 4.10 as the weights reduce at the scaled speed of 0.5. When participants walk with increased stride lengths as instructed, more push-off force is required (consequently increase the CoM acceleration cost) especially at higher walking speed to maintain the walking motion. To tolerate this, the weight of the CoM is reduced at high walking speed. CoM acceleration weights at low speeds (i.e., 0.2 and 0.3) also include some large values. We observed that during the experiment, participants' longer-stride walking styles vary significantly at low speeds with some walking very carefully and some being more relaxed. Such differences in walking styles may contribute to the large difference of CoM acceleration cost at low walking speed.

Chapter 5

From Simulation to Exoskeleton Control

In this chapter, we explore the possibilities of using the AR-IOC developed in Chapter 4 to assist exoskeleton controller design. We first construct a realistic exoskeleton model and attach it to the musculoskeletal model (section 5.1). Then, we use DC along with the optimal human cost function and the exoskeleton cost to obtain physics-informed optimal reference trajectories for the human-exoskeleton system (section 5.2). Last, we use a spatial-temporal tracking controller with the obtained optimal reference trajectories on one participant to evaluate the exoskeleton performance (section 5.3 and 5.4).

5.1 Human-exoskeleton Modeling

The human-exoskeleton model includes a musculoskeletal model and an exoskeleton model which is shown in Fig. 5.1. The musculoskeletal model is customized based on a baseline OpenSim model described in section 4.3.4 which allows sagittal-plane motion. The model mass, body geometries, and muscle properties are scaled (methods also described in section 4.1) to match a healthy male subject with a height of 1.80m, mass of 65.2kg, and age of 25.

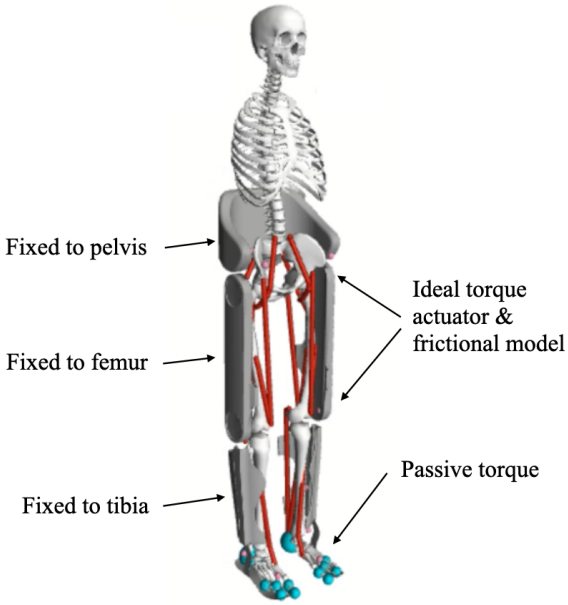

Figure 5.1: The 12-DoF 18-muscle msculoskeletal model is combined with an exoskeleton system that has 4 powered motors.

The exoskeleton is modelled based on the Indego Explorer exoskeleton (Parker Hannifin, USA). It contains five segments with powered motors at hip and knee joints. These segments are connected to the pelvis, thigh, and shank of the musculoskeletal model using fixed joints. Pin joint constraints are not included between the exoskeleton segments to avoid over constraints of the human-exoskeleton system. The joint friction is modelled as a combination of viscous damping and a constant kinetic friction term. The exoskeleton segments are assumed to be solid bodies where their inertial properties are identified together with the frictional properties of the joints by exciting the exoskeleton using chirp torque. As the effect of inertia is attributed to both the segment and the motor during motion, the identification results in larger segment inertia to include the motor effect. This simplification allows easier implementation of the exoskeleton systems in OpenSim. The motor actuation is modelled as ideal torque actuators with 200Nm of maximum torque between exoskeleton segments. As the exoskeleton does not have powered ankle joints, we add two passive joint actuators to simulate the constraints imposed by the exoskeleton's foot support. We also experimented with a more realistic version of the exoskeleton model where the interaction between the exoskeleton and the musculoskeletal model is formulated as bushing forces at all contact locations. This better represents the compliance due to soft body tissues. Although we were able to generate animation by running model forward dy-

namics, we encountered challenges during the optimization stages (e.g., optimization time increased from 5 minutes to 3 hours) due to significantly more complex dynamic system description. In addition, obtaining realistic impedance for the bushing forces is challenging. Thus, we decide to use the former formulation instead.

5.2 Gait Optimization

In this work, we formulated the gait optimization problem as a DC problem which is solved using OpenSim Moco [16]. The objective function for the direct collocation problem is modified based on (4.13):

$$\begin{aligned} J = & \frac{1}{d} \int_0^{t_f} \Bigg[10 \sum_i \left(\frac{u_i^3}{n} \right) + 10^{-2} \sum_j \tau_j^2 \\ & + \omega_{torso_{ef}} \times \tau_{torso}^2 \\ & + \omega_{CoM} \times a_{CoM}^2 \\ & + \omega_{slide} \left[\left(\hat{F}_{lf}^y \hat{v}_{lf}^{fx} \right)^2 + \left(\hat{F}_{rf}^y \hat{v}_{rf}^{fx} \right)^2 \right] \\ & + \omega_{stability} \times \left(p_{CoMx} - p_{midBoS} \right)^2 \\ & + \omega_e \times \sum_\alpha \left(\frac{\tau_\alpha}{200} \right)^2 \Bigg] dt. \end{aligned} \tag{5.1}$$

The first six terms of the objective function are adopted from (4.13). The first term penalizes the cubed muscle excitation. The second term penalizes the sum of all passive ligament torques. The third term penalizes the torso actuation. The fourth term penalizes the acceleration at the CoM. The fifth term penalizes the foot sliding behaviour. The sixth term penalizes deviation of the CoM position away from the middle point of the BoS. The last term is added for the exoskeleton which penalizes the sum of squared motor torques scaled by 200 (scaling the maximum torque to 1). This captures the energy consumption of the exoskeleton and relates to the interaction force between the human and the exoskeleton. The scaling is applied to the torque values so that the exoskeleton control signal has a relatively small magnitude, which we found to be beneficial for the numerical optimization

of the DC problem. Finally, the integral of the cost function is divided by the total forward distance d measured at the CoM.

The weight values $[\omega_{torso_{ef}}, \omega_{CoM}, \omega_{slide}, \omega_{stability}]$ are tuned to accurately match the simulated gait to the reference gait collected from the experiment. We first obtain the natural walking data without exoskeleton from the participant and optimized these weights using the AR-IOC described in Alg. 2. We assume that the CNS cost function structure for walking does not have significant changes after wearing the exoskeleton. In reality, the CNS cost function adapts in interacting with the exoskeleton and may vary based on how long the subject interacts with the exoskeleton, and how the exoskeleton controller is implemented. if the exoskeleton controller is adaptive, co-adaptation between the user and the exoskeleton can also occur which is challenging to model in simulation. To simplify the problem, we assume that the human has a static cost function in our study. Besides the objective function, additional constraints (i.e., symmetry and coordinate ranges) are applied to the optimization program similar to section 4.3.2. The walking speed is constrained to the desired speed while the trajectory final time allows modification to achieve the lowest cost of transport. Last, the trajectories are discretized to 50 nodes during optimization.

The optimal joint angle and feedforward torque trajectories are obtained by solving DC problems for the human-exoskeleton system. Considering that humans generally walk slowly when wearing the exoskeleton, we constrained the DC problem to 0.2 scaled speed (i.e., 0.59m/s for the participant). ω_e in (5.1) is set to be 0.1 so that the assistive torques from the exoskeleton have roughly the same magnitude as the natural human torque during walking (obtained from DC as well). To compare the change in the optimal trajectories for the human-exoskeleton system, we obtained the DC solutions for three cases: natural walking without the exoskeleton, walking with the passive exoskeleton, and walking with the powered exoskeleton. The obtained joint angle and the GRF trajectories are summarized in Fig. 5.2.

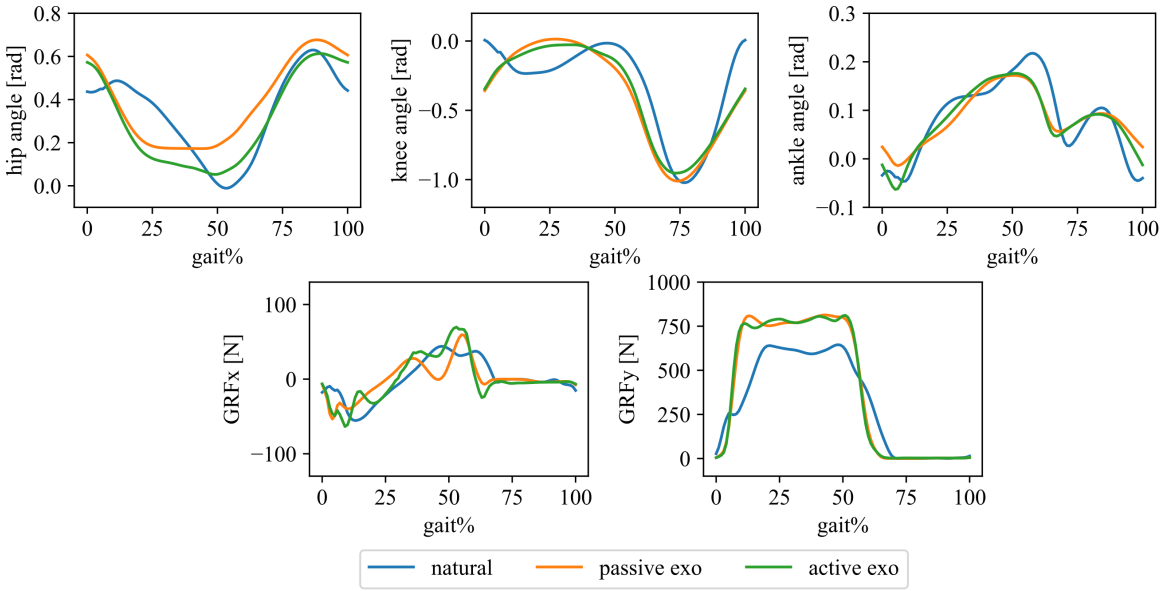

Figure 5.2: Comparison of the joint angle (hip, knee, and ankle) and GRFs (not normalized, horizontal as x, and vertical as y) trajectories between natural walking without exoskeleton (blue), walking with passive or unpowered exoskeleton (orange), and walking with active or powered exoskeleton (green).

The obtained joint angle and GRF trajectories for the human-exoskeleton system show significant differences from the natural walking trajectories. First, the knees do not flex during the stance phase (i.e., 0% to 60% of the gait cycle). This is caused by the user's compensatory strategy in response to the additional weight from the exoskeleton (can be seen from the vertical GRF plot in Fig. 5.2) as flexing knees during the stance face significantly increases the muscle excitation cost. Second, the ankle angle showed a reduction in range of motion due to the passive ankle constraints from the exoskeleton formulation. Consequently, the hip angles exhibit early hip extension to account for the ankle's reduction in mobility. On the other hand, the joint angle and GRF trajectories between the passive and the active exoskeleton conditions do not differ significantly except the hip angle having a larger range of motion and the horizontal GRF having more oscillation in the active case. However, there is a significant reduction in large muscle activation from the simulation as shown in Fig. 5.3. In addition, the change in the muscle activation patterns between the natural and passive/active case is likely due to the change in gait patterns as shown in Fig. 5.2.

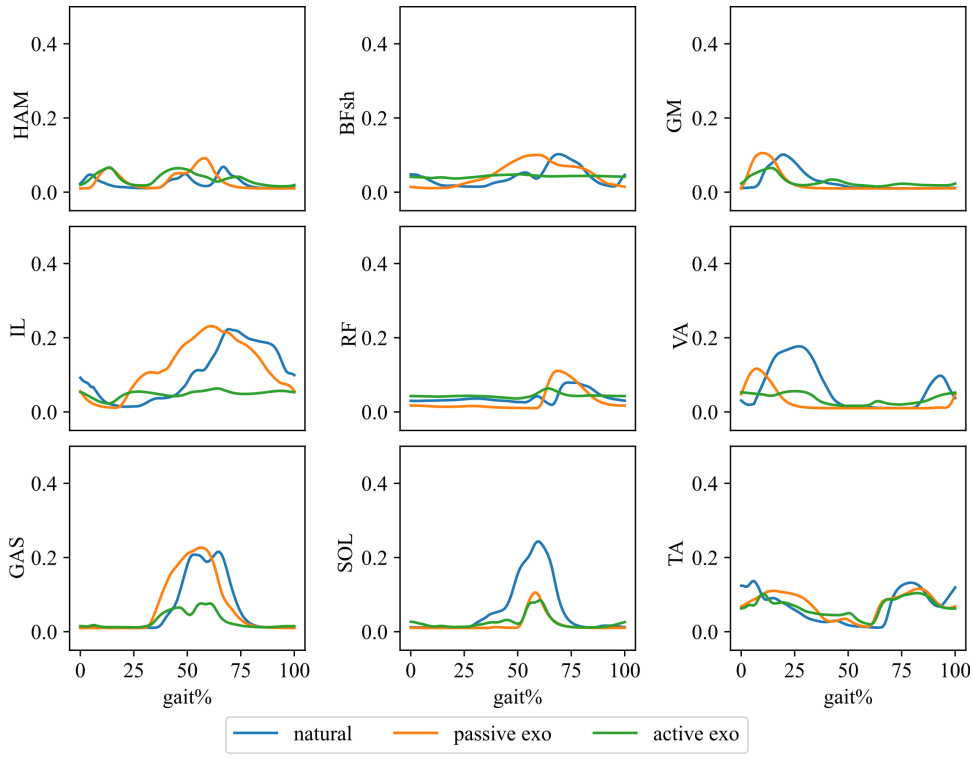

Figure 5.3: Comparison of simulated muscle activation between natural walking without exoskeleton (blue), walking with passive or unpowered exoskeleton (orange), and walking with active or powered exoskeleton (green).

Fig. 5.4 also compares the hip and knee joint torque trajectories between natural walking without the exoskeleton, and the optimized feedforward assistive torque from the exoskeleton. Due to the different dynamics of the human-exoskeleton system from the humans, the obtained assistive joint torque differs from the human torque during walking. This indicates the disadvantage of using human torque as the feedforward torque during exoskeleton assistance.

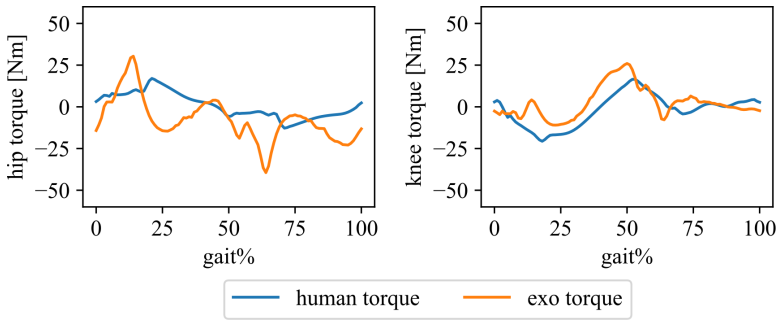

Figure 5.4: Comparison between the simulated human joint torque during natural walking without the exoskeleton (blue), and the simulated exoskeleton torque (exluding the human torque) during assistive walking with the exoskeleton (orange).

5.3 Experimental Setup

The goal of the exoskeleton control experiment is to evaluate the performance of exoskeleton assistance using different reference trajectories with a simple spatial-temporal tracking controller. The simple spatial-temporal tracking controller consists of two components: feedback control and feedforward control. The feedback component consists of four individual proportional-derivative (PD) controllers at hip and knee joints that minimize the tracking errors between the reference joint angles and the actual angles measured from the exoskeleton. The proportional gain k_p and the derivative gain k_d are tuned manually for the user comfort ($k_p = 3$ and $k_d = 0.25$ at hip, and $k_p = 0.9$ and $k_d = 0.08$ at knee). The feedforward component applies the predefined open-loop torque profile directly. Besides k_p and k_d for the feedback control, two scaling factors k_{fb}, k_{ff} are applied to the feedback and feedforward components respectively. These two scaling factors control the overall assistance levels which are also tuned based on the user comfort (0.2 and 0.3 for k_{fb} and k_{ff} respectively for all cases). Due to potential different reference angles between the exoskeleton and the simulation, and the compliance between the exoskeleton and the human body, the amplitude and the offset of the reference trajectories are also tuned by trial and error and the user feedback during the experiments to ensure comfort and proper level of assistance.

The experimental setup is summarized in Table 5.1 including one trial (Trial 1) with exoskeleton powered off and four trials (Trial 2-5) with exoskeleton powered on. Among trial 2-5, two sets of reference trajectories and two controllers are tested. First, The reference trajectories include the optimized gait from solving the DC problem using the exoskeleton-

human system (i.e., green trajectories in Fig. 5.2 and orange trajectories in Fig. 5.4), and the natural walking gait from solving the DC problem using only the musculoskeletal model (i.e., the blue trajectories in Fig. 5.2 and Fig. 5.4). Two controllers include only feedback control and a combination of feedback and feedforward control. For each trial, the participant is instructed to walk for 4 minutes but only the data from the last 2-min period is used for analysis (the first 2-min walking is for proper acclimatization to the walking condition). The trial information is not disclosed to the participant during the experiment.

Table 5.1: Different trials for the exoskeleton control experiment for comparing the performance of using different reference trajectories and control schemes.

Trials	Exoskeleton power	controller	Reference	Time [min]
Trial 1	OFF	N/A	N/A	4
Trial 2	ON	Feedback	Human + Exoskeleton	4
Trial 3	ON	Feedback + feedforward	Human + Exoskeleton	4
Trial 4	ON	Feedback	Human	4
Trial 5	ON	Feedback + feedforward	Human	4

The performance of the exoskeleton assistance is evaluated based on the reduction of muscle activities which requires measurement of muscle EMG signals. Fig. 5.5 shows the overall experimental setup with the EMG sensor placement. The amount of muscle activity is calculated by integrating the rectified EMG signals. The muscle activity ratio (shown in (5.2)) is then computed which compares the amount of muscle activity from Trial 2-5 to the baseline from Trial 1.

$$\text{muscle activity ratio} = \frac{\int |EMG_{trial}| dt}{\int |EMG_{baseline}| dt} \tag{5.2}$$

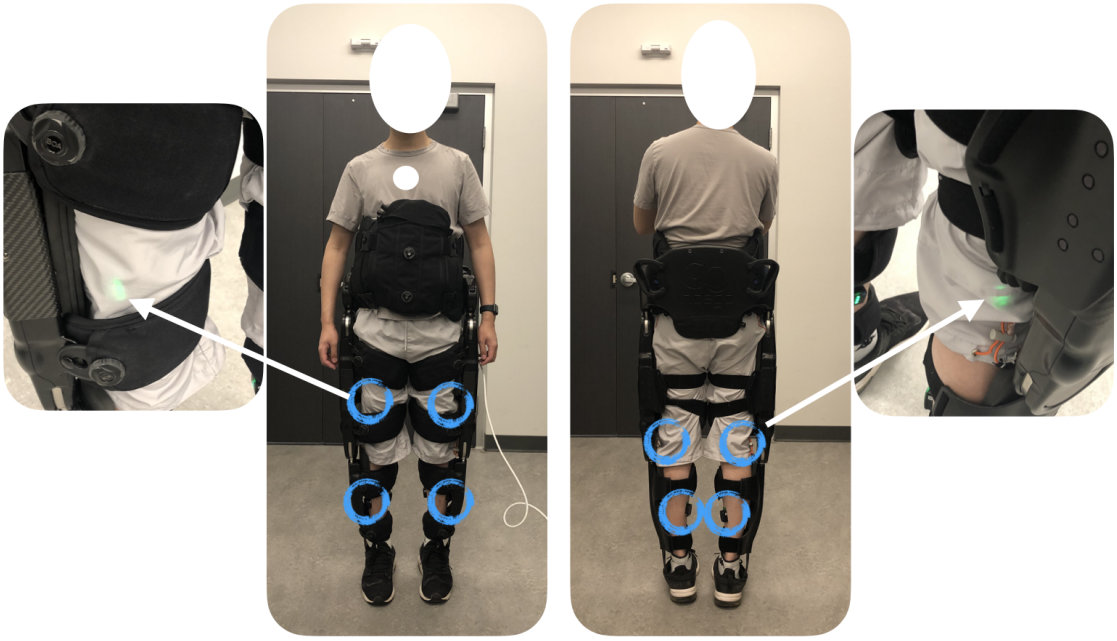

Figure 5.5: Experimental setup of the exoskeleton control experiments. The subject is instructed to wear a powered exoskeleton with actuated hip and knee joints. Eight EMG sensors are placed at RF, BF, TA, and GASl (circled in blue) so that the sensors do not interfere with the exoskeleton. During the experiment, the EMG, GRF, and exoskeleton data (kinematic and torque signals) are recorded.

5.4 Results and Discussion

The experiment was performed with the same participant that we used to model the musculoskeletal model. To visualize the change in muscle activities, the average EMG signals for all five trials over one gait cycles are provided in Fig. 5.6. The individual gait cycles were sectioned based on the GRF data recorded from the treadmill with a 50-N threshold on the filtered vertical GRF for heel strike detection. The RF activities in Fig. 5.6 showed good peak alignment with the two close-by knee extensors VA and RF in Fig. 5.3. The Biceps Femoris (BF) activities in Fig. 5.6 showed good peak alignment with two close-by knee flexors HAM and BFsh in Fig. 5.3. TA and Gastrocnemius Lateralis (GASl) activities also showed close match to the simulated muscle activation profiles at TA and GAS in Fig. 5.3. Comparing to the passive exoskeleton walking (Trial 1), muscle

activities for RF, TA, and GASl showed significant reduction for all assistive trials (Trial 2-5), whereas BF showed no reduction in activities when exoskeleton assistance is applied. Table 5.2 quantifies the change in muscle activities with respect to the reference trial (Trial 1) using (5.2). Comparing Trial 3 to Trial 2, the optimized feedforward torque profile from the exoskeleton-human system reduced the thigh muscle activities (i.e., RF and BF) while maintaining the shank muscle activities at the similar levels. This indicates the potential of using feedforward control to improve the exoskeleton assistance. Comparing Trial 5 to Trial 4, the optimized feedforward torque profile from only the muscukosleletal model, however, increased the muscle activities at all muscles except BF.

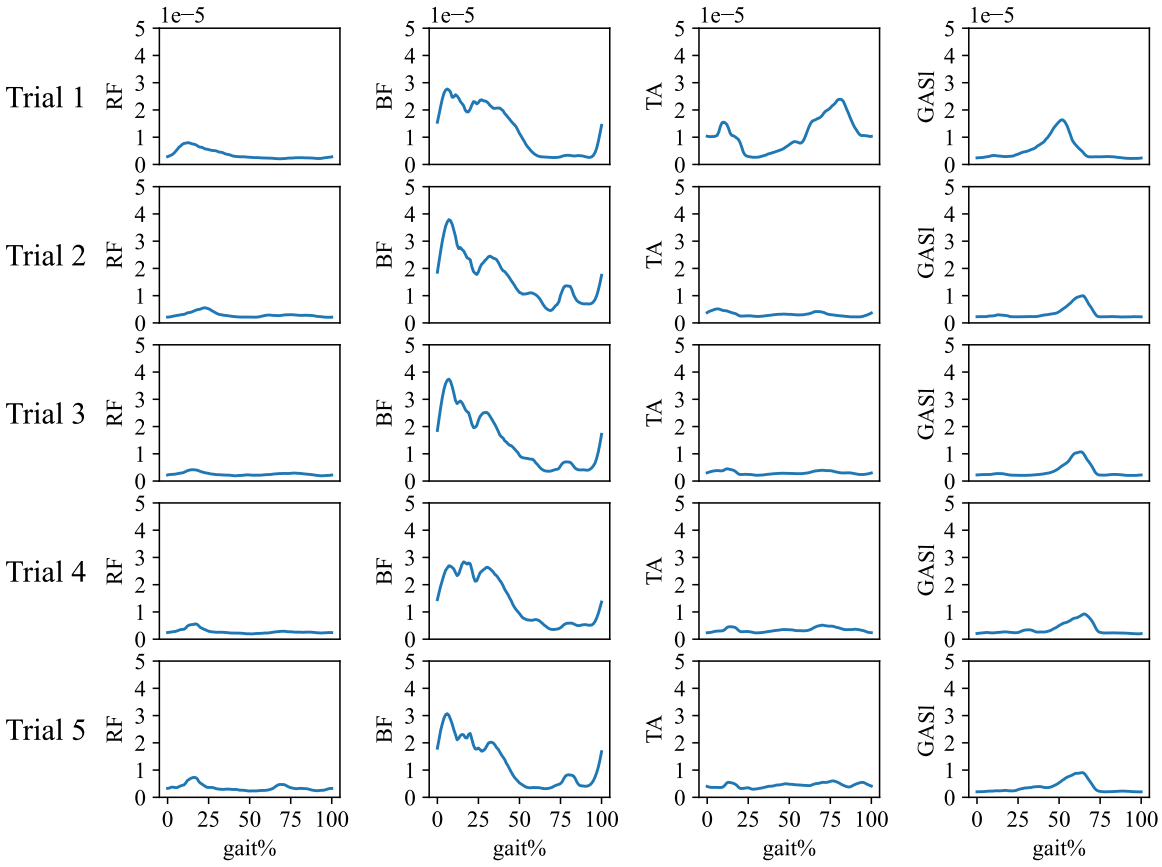

Figure 5.6: Average EMG signals at RF, BF, TA, and GASl over one gait cycles for 5 different trials.

Table 5.2: muscle activity ratio of experimental trials with powered exoskeleton.

	Trial 2	Trial 3	Trial 4	Trial 5
RF	0.81	0.70	0.77	0.97
BF	1.29	1.15	1.11	0.96
TA	0.28	0.27	0.31	0.40
GASl	0.67	0.69	0.69	0.71

The differences in change of muscle activities when introducing the feedforward torque assistance between two sets of the reference trajectories can be potentially caused by the difference in overall assistive torque profiles. Fig. 5.7 and Fig. 5.8 visualize the total torque as well as the feedback / feedforward torque breakdown for Trial 2 to 5. The feedforward torque profiles modified the feedback torque profiles for both reference trajectory sets. The feedback torque modifications at the knee are similar between the two feedforward torque profiles which increase the knee's range of motion during stance. For hip joints, the feedforward torque in Trial 3 added more modifications to the feedback torque around the stance phase which may contribute to the reduction in thigh muscle activities. When comparing the feedback and feedforward components for Trial 3 and 5, the feedback components counteract the feedforward component during the swing phase (i.e., feedforward and feedback torques showed opposite signs). This is undesirable as such offset indicates the conflict between the two control components. This conflict may be caused by the naive implementation of the time-dependent feedforward control. As feedforward control requires an accurate match between the human gait phase and the target gait phase from the reference trajectories, any small amount of time shift will cause incorrect assistance to the walking motion and even cause disturbances. We noticed this during the experiment when we experimented with high feedforward control with low feedback, and the participant failed to maintain the walking motion.

For future experiments, gait phase-based or time-independent feedforward controller [65] will be more desirable for evaluating the feedforward assistance as they provide better assistance timing. The proposed modelling approach for the exoskeleton-human system can also enable future research about personalized reference trajectories for impaired subjects (through musculoskeletal modelling), special rehabilitation setup (e.g., walking with external support such as weight compensation), and specific rehabilitation needs (e.g., optimizing gait and assistance to reducing activation of targeted muscles).

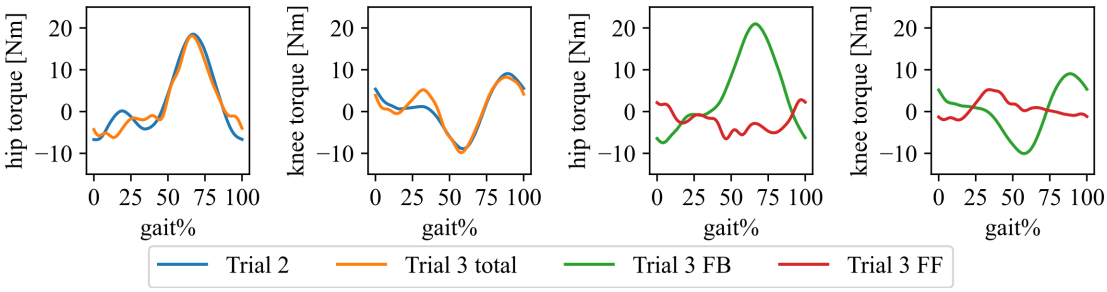

Figure 5.7: Average assistive torque applied by the exoskeleton over one gait cycle for Trial 2 and 3. The reference trajectories are optimized for the human-exoskeleton system; The first two plots show the total torque; the last two plots show the breakdown of the feedback (FB) and the feedforward (FF) torque in Trial 3.

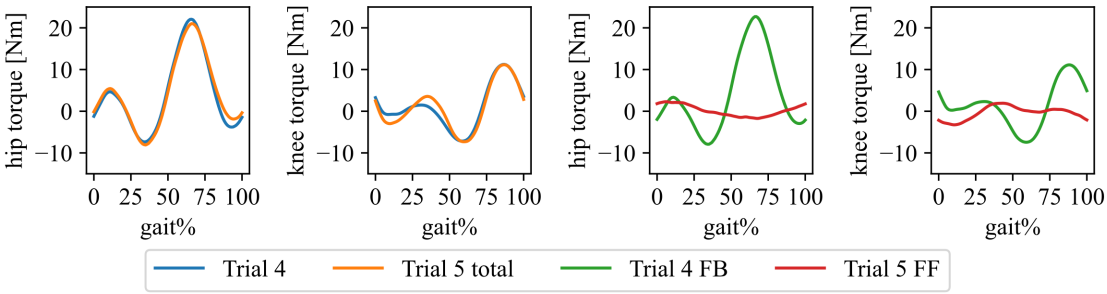

Figure 5.8: Average assistive torque applied by the exoskeleton over one gait cycle for Trial 4 and 5. The reference trajectories are optimized for the human musculoskeletal model only; The first two plots show the total torque; the last two plots show the breakdown of the feedback (FB) and the feedforward (FF) torque in Trial 4.

Chapter 6

Conclusion

The main goal of this thesis is to recover the optimal cost function for natural walking, which would improve our understanding of the criteria that the CNS uses to optimize the gait patterns, and assist the design of controllers for assistive devices. The primary objectives established for this main goal are to 1) construct a predictive simulation environment for natural walking using musculoskeletal models, 2) formulate efficient optimization problems for solving optimal gait trajectories given cost functions, 3) develop efficient inverse methods for cost function tuning to recreate natural walking in simulation, 4) and validate the obtained optimal cost functions in assisting lower-limb exoskeleton controller design.

Prior to these objectives, a thorough understanding of the existing gait simulation and analysis work using humanoids and musculoskeletal models was established which covered two main optimization-based methods: OC and RL. These two methods suit the purpose of developing predictive simulation environments for natural walking. However, existing studies that used both these methods for musculoskeletal simulation lacked systematic approaches for cost function tuning due to high complexity of the optimization problem. DRL-based locomotion studies experienced additional challenges of generating realistic walking motion due to a larger optimization scope than OC.

As the first attempt to address the cost function tuning problem for natural walking, a novel DRL learning method, which generates a control policy with close-to-natural walking behaviour, was developed. The proposed neuromechanically-inspired cost function contributes to the effective learning of the realistic gait by the DRL agent. The nature-inspired CL scheme leads to efficient convergence to natural and bilateral symmetric gait by adaptively tuning the cost function weights while maintaining the agent's walking capability. With the proposed DRL approach, we identified two differences between the agent

and experimental gait patterns, including lack of knee extension before foot strike, and toe walking. Potential causes of such differences are also discussed.

Although DRL approaches generated closed-loop feedback policies for generating walking motion, its requirement for long computational time hinders its extension to subject-specific cost function tuning. As the second attempt to address the cost function tuning problem, an efficient IOC algorithm named AR-IOC is proposed. We showcased the efficiency of the proposed algorithm in tuning cost functions and matching gait trajectories using both synthetic data and experimental data. The optimized cost weights and the corresponding optimal gait trajectories from our algorithm were also compared with the solutions from GA. With the AR-IOC, the correlation between the walking tasks (i.e., walking speed and stride time) and the cost function weights were studied. We also discussed the limitation of the proposed algorithm in terms of stability of the inner-loop DC problems and limitation of the experimental data.

With the efficient AR-IOC algorithm, we explored the potential for using predictive simulation to generate physics-informed reference trajectories for lower-limb exoskeleton tracking controllers. This is to address the issue of existing tracking controllers that often neglect the personalized human cost functions and the altered dynamics due to the addition of the exoskeleton systems. By combining the optimal human cost function obtained using AR-IOC and the exoskeleton cost, we obtained the optimal gait trajectories different from the natural walking trajectories. These trajectories were then tested in real exoskeleton systems using a spatial-temporal controller and their performances in reducing muscle activities are compared to using nominal natural walking trajectories. The limitation of the controller design is also discussed.

With the predictive simulation environments and the efficient cost function tuning methods, this thesis serves as a catalyst for enabling personalized rehabilitation design based on a more accurate musculoskeletal modeling and simulation pipeline. The presented frameworks, that cover from data collection and post-processing, to simulation and experiments, serve as a guidance and reference to future development in this field, such as extending the musculoskeletal simulation to impaired subjects, different locomotion tasks, different rehabilitation systems and control schemes.

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[2] Marko Ackermann and Antonie J Van den Bogert. Optimality principles for model-based prediction of human gait. *Journal of biomechanics*, 43(6):1055–1060, 2010.

[3] Frank C Anderson and Marcus G Pandy. Dynamic optimization of human walking. *J. Biomech. Eng.*, 123(5):381–390, 2001.

[4] RJ Baker, FY Leboeuf, Julie Reay, Morgan Sangeux, et al. The conventional gait model-success and limitations. *Handbook of human motion*, pages 489–508, 2018.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[6] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *25th annual conference on neural information processing systems (NIPS 2011)*, volume 24. Neural Information Processing Systems Foundation, 2011.

[7] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[8] Lindsay J Bhargava, Marcus G Pandy, and Frank C Anderson. A phenomenological model for estimating metabolic energy consumption in muscle contraction. *Journal of biomechanics*, 37(1):81–88, 2004.

[9] Gabriele Bovi, Marco Rabuffetti, Paolo Mazzoleni, and Maurizio Ferrarin. A multiple-task gait analysis approach: kinematic, kinetic and emg reference data for healthy young and adult subjects. *Gait & posture*, 33(1):6–13, 2011.

[10] Stéphane Caron and Abderrahmane Kheddar. Dynamic walking over rough terrains by nonlinear predictive control of the floating-base inverted pendulum. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5017–5024. IEEE, 2017.

[11] Debora Clever, R Malin Schemschat, Martin L Felis, and Katja Mombaur. Inverse optimal control based identification of optimality criteria in whole-body human walking on level ground. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1192–1199. IEEE, 2016.

[12] Michael Damsgaard, John Rasmussen, Søren Tørholm Christensen, Egidijus Surma, and Mark De Zee. Analysis of musculoskeletal systems in the anybody modeling system. *Simulation Modelling Practice and Theory*, 14(8):1100–1111, 2006.

[13] Roy B Davis III, Sylvia Ounpuu, Dennis Tyburski, and James R Gage. A gait analysis data collection and reduction technique. *Human movement science*, 10(5):575–587, 1991.

[14] Leanne De Vree and Raffaella Carloni. Deep reinforcement learning for physics-based musculoskeletal simulations of healthy subjects and transfemoral prostheses' users during normal walking. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2021.

[15] 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. *IEEE transactions on biomedical engineering*, 54(11):1940–1950, 2007.

[16] Christopher L Dembia, Nicholas A Bianco, Antoine Falisse, Jennifer L Hicks, and Scott L Delp. Opensim moco: musculoskeletal optimal control. *PLOS Computational Biology*, 16(12):e1008493, 2020.

[17] Christopher L Dembia, Amy Silder, Thomas K Uchida, Jennifer L Hicks, and Scott L Delp. Simulating ideal assistive devices to reduce the metabolic cost of walking with heavy loads. *PloS one*, 12(7):e0180320, 2017.

[18] Joachim Denk and Gunther Schmidt. Synthesis of a walking primitive database for a humanoid robot using optimal control techniques. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, pages 319–326, 2001.

[19] Andreas Doerr, Nathan D Ratliff, Jeannette Bohg, Marc Toussaint, and Stefan Schaal. Direct loss minimization inverse optimal control. In *Robotics: Science and Systems*, 2015.

[20] Peter Englert, Ngo Anh Vien, and Marc Toussaint. Inverse kkt: Learning cost functions of manipulation tasks from demonstrations. *The International Journal of Robotics Research*, 36(13-14):1474–1488, 2017.

[21] Bellman Richard Ernest. Dynamic programming, 2003.

[22] Mahdokht Ezati, Borna Ghannadi, and John McPhee. A review of simulation methods for human movement dynamics with emphasis on gait. *Multibody System Dynamics*, 47(3):265–292, 2019.

[23] Antoine Falisse, Lorenzo Pitto, Hans Kainz, Hoa Hoang, Mariska Wesseling, Sam Van Rossom, Eirini Papageorgiou, Lynn Bar-On, Ann Hallemans, Kaat Desloovere, et al. Physics-based simulations to predict the differential effects of motor control and musculoskeletal deficits on gait dysfunction in cerebral palsy: a retrospective case study. *Frontiers in human neuroscience*, 14:40, 2020.

[24] Antoine Falisse, Gil Serrancolí, Christopher L Dembia, Joris Gillis, Ilse Jonkers, and Friedl De Groote. Rapid predictive simulations with complex musculoskeletal models suggest that diverse healthy and pathological human gaits can emerge from similar control strategies. *Journal of The Royal Society Interface*, 16(157):20190402, 2019.

[25] Martin L Felis and Katja Mombaur. Using optimal control methods to generate human walking motions. In *International Conference on Motion in Games*, pages 197–207. Springer, 2012.

[26] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated machine learning*, pages 3–33. Springer, Cham, 2019.

[27] Nicholas P Fey, Glenn K Klute, and Richard R Neptune. Optimization of prosthetic foot stiffness to reduce metabolic cost and intact knee loading during below-knee amputee walking: a theoretical study. *Journal of biomechanical engineering*, 134(11), 2012.

[28] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58. PMLR, 2016.

[29] Patrick W Franks, Nicholas A Bianco, Gwendolyn M Bryan, Jennifer L Hicks, Scott L Delp, and Steven H Collins. Testing simulated assistance strategies on a hip-knee-ankle exoskeleton: a case study. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 700–707. IEEE, 2020.

[30] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[31] Hartmut Geyer and Hugh Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering*, 18(3):263–273, 2010.

[32] Martin Grimmer, Mahdy Eslamy, and André Seyfarth. Energetic and peak power advantages of series elastic actuators in an actuated prosthetic leg for walking and running. In *Actuators*, volume 3, pages 1–19. Multidisciplinary Digital Publishing Institute, 2014.

[33] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[34] Geoffrey G Handsfield, Craig H Meyer, Joseph M Hart, Mark F Abel, and Silvia S Blemker. Relationships of 35 lower limb muscles to height and body mass quantified using mri. *Journal of biomechanics*, 47(3):631–638, 2014.

[35] Ayonga Hereid, Eric A Cousineau, Christian M Hubicki, and Aaron D Ames. 3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1447–1454. IEEE, 2016.

[36] Ayonga Hereid, Christian M Hubicki, Eric A Cousineau, and Aaron D Ames. Dynamic humanoid locomotion: A scalable formulation for hzd gait optimization. *IEEE Transactions on Robotics*, 34(2):370–387, 2018.

[37] Hermie J Hermens, Bart Freriks, Roberto Merletti, Dick Stegeman, Joleen Blok, Günter Rau, Cathy Disselhorst-Klug, and Göran Hägg. European recommendations for surface electromyography. *Roessingh research and development*, 8(2):13–54, 1999.

[38] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.

[39] At L Hof. Scaling gait data to body size. *Gait & posture*, 3(4):222–223, 1996.

[40] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, pages 507–523. Springer, 2011.

[41] MD Jacquelin Perry. Gait analysis: normal and pathological function. *New Jersey: SLACK*, 2010.

[42] Saso Jezernik, Gery Colombo, and Manfred Morari. Automatic gait-pattern adaptation algorithms for rehabilitation with a 4-dof robotic orthosis. *IEEE Transactions on Robotics and Automation*, 20(3):574–582, 2004.

[43] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 1620–1626. IEEE, 2003.

[44] Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017.

[45] Łukasz Kidziński, Sharada P Mohanty, Carmichael F Ong, Jennifer L Hicks, Sean F Carroll, Sergey Levine, Marcel Salathé, and Scott L Delp. Learning to run challenge: Synthesizing physiologically accurate motion using deep reinforcement learning. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 101–120. Springer, 2018.

[46] Łukasz Kidziński, Sharada Prasanna Mohanty, Carmichael F Ong, Zhewei Huang, Shuchang Zhou, Anton Pechenko, Adam Stelmaszczyk, Piotr Jarosik, et al. Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 121–153. Springer, 2018.

[47] Łukasz Kidzinski, Carmichael Ong, Sharada Prasanna Mohanty, Jennifer Hicks, Sean Carroll, Bo Zhou, Hongsheng Zeng, Fan Wang, Rongzhong Lian, Hao Tian, et al. Artificial intelligence for prosthetics: Challenge solutions. *The NeurIPS'18 Competition: From Machine Learning to Intelligent Conversations*, page 69, 2019.

[48] Henning Koch and Katja Mombaur. Exoopt-a framework for patient centered design optimization of lower limb exoskeletons. In *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 113–118. IEEE, 2015.

[49] Anne D Koelewijn and Jessica D Selinger. Predictive simulations of gait with exoskeletons that alter energetics. *bioRxiv*, 2021.

[50] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[51] Fabien Leboeuf, R Baker, Arnaud Barré, Julie Reay, Richard Jones, and Morgan Sangeux. The conventional gait model, an open-source implementation that reproduces the past but prepares for the future. *Gait & posture*, 69:235–241, 2019.

[52] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47), 2020.

[53] Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. Scalable muscle-actuated human simulation and control. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

[54] Song Joo Lee and Joseph Hidler. Biomechanics of overground vs. treadmill walking in healthy individuals. *Journal of applied physiology*, 104(3):747–755, 2008.

[55] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[56] Du-Xin Liu, Xinyu Wu, Wenbin Du, Can Wang, Chunjie Chen, and Tiantian Xu. Deep spatial-temporal model for rehabilitation gait: Optimal trajectory generation for knee joint of lower-limb exoskeleton. *Assembly Automation*, 2017.

[57] R Macaluso, K Embry, D Villarreal, and R Gregg. Human leg kinematics, kinetics, and emg during phase-shifting perturbations at varying inclines, 2020.

[58] Giorgos Marinou, Matthew Millard, Nejc Šarabon, and Katja Mombaur. Comparing the risk of low-back injury using model-based optimization: Improved technique versus exoskeleton assistance. *Wearable Technologies*, 2, 2021.

[59] David A McAllester, Tamir Hazan, and Joseph Keshet. Direct loss minimization for structured prediction. In *NIPS*, volume 1, page 3. Citeseer, 2010.

[60] Christian Meyer, Tim Killeen, Christopher S Easthope, Armin Curt, Marc Bolliger, Michael Linnebank, Björn Zörner, and Linard Filli. Familiarization with treadmill walking: How much is enough? *Scientific reports*, 9(1):1–10, 2019.

[61] Katja Mombaur, Jean-Paul Laumond, and Eiichi Yoshida. An optimal control model unifying holonomic and nonholonomic walking. In *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, pages 646–653. IEEE, 2008.

[62] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion—an inverse optimal control approach. *Autonomous robots*, 28(3):369–383, 2010.

[63] Katja Mombaur, Heike Vallery, Yue Hu, Jonas Buchli, Pranav Bhounsule, Thiago Boaventura, Patrick M Wensing, Shai Revzen, Aaron D Ames, Ioannis Poulakakis, et al. Control of motion and compliance. *Bioinspired Legged Locomotion*, pages 135–346, 2017.

[64] Igor Mordatch, Jack M Wang, Emanuel Todorov, and Vladlen Koltun. Animating human lower limbs using contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 32(6):1–8, 2013.

[65] Rezvan Nasiri, Mohammad Shushtari, Hossein Rouhani, and Arash Arami. Virtual energy regulator: A time-independent solution for control of lower limb exoskeletons. *IEEE Robotics and Automation Letters*, 6(4):7699–7705, 2021.

[66] Vinh Nguyen. Predictive simulation of human movement and applications to assistive device design and control. 2019.

[67] Vinh Q Nguyen, Russell T Johnson, Frank C Sup, and Brian R Umberger. Bilevel optimization for cost function determination in dynamic simulation of human gait. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(7):1426–1435, 2019.

[68] Vinh Q Nguyen, Brian R Umberger, and Frank C Sup. Predictive simulation of human walking augmented by a powered ankle exoskeleton. In *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, pages 53–58. IEEE, 2019.

[69] Todd C Pataky and John Y Goulermas. Pedobarographic statistical parametric mapping (pspm): a pixel-level approach to foot pressure image analysis. *Journal of biomechanics*, 41(10):2136–2143, 2008.

[70] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[71] Anne-Sophie Puydupin-Jamin, Miles Johnson, and Timothy Bretl. A convex approach to inverse optimal control and its application to modeling human locomotion. In *2012 IEEE International Conference on Robotics and Automation*, pages 531–536. IEEE, 2012.

[72] Apoorva Rajagopal, Christopher L Dembia, Matthew S DeMers, Denny D Delp, Jennifer L Hicks, and Scott L Delp. Full-body musculoskeletal model for muscle-driven simulation of human gait. *IEEE transactions on biomedical engineering*, 63(10):2068–2079, 2016.

[73] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736, 2006.

[74] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. (approximate) subgradient methods for structured prediction. In *Artificial Intelligence and Statistics*, pages 380–387. PMLR, 2007.

[75] Heydar Sadeghi, Paul Allard, François Prince, and Hubert Labelle. Symmetry and limb dominance in able-bodied gait: a review. *Gait & posture*, 12(1):34–45, 2000.

[76] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[77] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[78] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[79] Ajay Seth, Jennifer L Hicks, Thomas K Uchida, Ayman Habib, Christopher L Dembia, James J Dunne, Carmichael F Ong, Matthew S DeMers, Apoorva Rajagopal, Matthew Millard, et al. Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology*, 14(7):e1006223, 2018.

[80] Mohammad Sharif Shourijeh. Optimal control and multibody dynamic modelling of human musculoskeletal systems. 2013.

[81] Mohammad Sharif Shourijeh and John McPhee. Forward dynamic optimization of human gait simulations: a global parameterization approach. *Journal of Computational and Nonlinear Dynamics*, 9(3), 2014.

[82] Mohammad Shushtari, Rezvan Nasiri, and Arash Arami. Online reference trajectory adaptation: A personalized control strategy for lower limb exoskeletons. *IEEE Robotics and Automation Letters*, 7(1):128–134, 2022.

[83] 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. *Journal of neuroengineering and rehabilitation*, 18(1):1–17, 2021.

[84] Manish Sreenivasa, Matthew Millard, Martin Felis, Katja Mombaur, and Sebastian I Wolf. Optimal control based stiffness identification of an ankle-foot orthosis using a predictive walking model. *Frontiers in computational neuroscience*, 11:23, 2017.

[85] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[86] Darryl G Thelen and Frank C Anderson. Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *Journal of biomechanics*, 39(6):1107–1115, 2006.

[87] Darryl G Thelen, Frank C Anderson, and Scott L Delp. Generating dynamic simulations of movement using computed muscle control. *Journal of biomechanics*, 36(3):321–328, 2003.

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

[89] Thomas K Uchida, Ajay Seth, Soha Pouya, Christopher L Dembia, Jennifer L Hicks, and Scott L Delp. Simulating ideal assistive devices to reduce the metabolic cost of running. *PloS one*, 11(9):e0163417, 2016.

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

[91] James R Usherwood, AJ Channon, JP Myatt, JW Rankin, and TY Hubel. The human foot and heel–sole–toe walking strategy: a mechanism enabling an inverted pendular gait with low isometric muscle force? *Journal of The Royal Society Interface*, 9(75):2396–2402, 2012.

[92] Heike Vallery, Edwin HF Van Asseldonk, Martin Buss, and Herman Van Der Kooij. Reference trajectory generation for rehabilitation robots: complementary limb motion estimation. *IEEE transactions on neural systems and rehabilitation engineering*, 17(1):23–30, 2008.

[93] Antonie J Van den Bogert, Maarten Hupperets, Heiko Schlarb, and Berthold Krabbe. Predictive musculoskeletal simulation using optimal control: effects of added limb mass on energy cost and kinematics of walking and running. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, 226(2):123–133, 2012.

[94] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[95] Letian Wang, Shiqian Wang, Edwin HF van Asseldonk, and Herman van der Kooij. Actively controlled lateral gait assistance in a lower limb exoskeleton. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 965–970. IEEE, 2013.

[96] Samuel R Ward, Carolyn M Eng, Laura H Smallwood, and Richard L Lieber. Are current measurements of lower extremity muscle architecture accurate? *Clinical orthopaedics and related research*, 467(4):1074–1082, 2009.

[97] Jiacheng Weng, Ehsan Hashemi, and Arash Arami. Natural walking with musculoskeletal models using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 6(2):4156–4162, 2021.

[98] Jiacheng Weng, Ehsan Hashemi, and Arash Arami. Adaptive reference inverse optimal control for natural walking with musculoskeletal models. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2022. under review.

[99] James T Wilson, Frank Hutter, and Marc Peter Deisenroth. Maximizing acquisition functions for bayesian optimization. *arXiv preprint arXiv:1805.10196*, 2018.

[100] David A Winter. Kinematic and kinetic patterns in human gait: variability and compensating effects. *Human movement science*, 3(1-2):51–76, 1984.

[101] Xinyu Wu, Du-Xin Liu, Ming Liu, Chunjie Chen, and Huiwen Guo. Individualized gait pattern generation for sharing lower limb exoskeleton robot. *IEEE Transactions on Automation Science and Engineering*, 15(4):1459–1470, 2018.

[102] Zhaoming Xie, Hung Yu Ling, Nam Hee Kim, and Michiel van de Panne. Allsteps: Curriculum-driven learning of stepping stone skills. *arXiv preprint arXiv:2005.04323*, 2020.

[103] Tingfang Yan, Marco Cempini, Calogero Maria Oddo, and Nicola Vitiello. Review of assistive strategies in powered lower-limb orthoses and exoskeletons. *Robotics and Autonomous Systems*, 64:120–136, 2015.

[104] Bo Zhou, Hongsheng Zeng, Fan Wang, Rongzhong Lian, and Hao Tian. Efficient and robust learning on elaborated gaits with curriculum learning. In *The NeurIPS'18 Competition*, pages 265–276. Springer, 2020.

[105] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.