공학박사학위논문

Control of Physically Simulated Humanoids for Generating Realistic Human Locomotion

사람의 자연스러운 보행 동작 생성을 위한 물리 시뮬레이션 기반 휴머노이드 제어 방법

2014년 8월

서울대학교 대학원
전기.컴퓨터공학부
이 윤 상

Control of Physically Simulated Humanoids for Generating Realistic Human Locomotion

사람의 자연스러운 보행 동작 생성을 위한 물리 시뮬레이션 기반 휴머노이드 제어 방법

지도교수 이 제 희

이 논문을 공학 박사 학위논문으로 제출함

2014년 8월

서울대학교 대학원

전기.컴퓨터공학부

이 윤 상

이윤상의 공학박사 학위논문을 인준함

2014년 8월

위 원 장	김 명 수
부위원장	이 제 희
위 원	박 종 우
위 원	박 문 석
위 원	권 태 수

Abstract

Controlling artificial humanoids to generate realistic human locomotion has been considered as an important problem in computer graphics and robotics. However, it has been known to be very difficult because of the underactuated characteristics of the locomotion dynamics and the complex human body structure to be imitated and simulated. In this thesis, we presents controllers for physically simulated humanoids that exhibit a rich set of human-like and resilient simulated locomotion. Our approach exploits observable and measurable data of a human to effectively overcome difficulties of the problem. More specifically, our approach utilizes observed human motion data collected by motion capture systems and reconstructs measured physical and physiological properties of a human body.

We propose a data-driven algorithm to control torque-actuated biped models to walk in a wide range of locomotion skills. Our algorithm uses human motion capture data and realizes an human-like locomotion control facilitated by inherent robustness of the locomotion motion. Concretely, it takes reference motion and generates a set of joint torques to generate human-like walking simulation. The idea is continuously modulating the reference motion such that even a simple tracking controller can reproduce the reference motion. A number of existing data-driven techniques such as motion blending, motion warping, and motion graph can facilitate the biped control

with this framework.

We present a locomotion control system that controls detailed models of a human body with the musculotendon actuating process to create more human-like simulated locomotion. The simulated humanoids are based on measured properties of a human body and contain maximum 120 muscles. Our algorithm computes the optimal coordination of muscle activations and actively modulates the reference motion to fathifully reproduce the reference motion or adapt the motion to meet new conditions. Our scalable algorithm can control various types of musculoskeletal humanoids while seeking harmonious coordination of many muscles and maintaining balance.

We demonstrate the strength of our approach with examples that allow simulated humanoids to walk and run in various styles, adapt to change of models (e.g., muscle weakness, tightness, joint dislocation), environments (e.g., external pushes), goals (e.g., pain reduction and efficiency maximization), and perform more challenging locomotion tasks such as turn, spin, and walking while steering its direction interactively.

Keywords: Computer Animation, Physically Based Simulation, Biped Locomotion Control, Humanoid Locomotion Control, Musculoskeletal Humanoid Control

Student Number: 2007-23045

Contents

List of Figures

Chapter 1

Introduction

Locomotion controllers for physically simulated humanoids has been regarded as an significant topic in computer graphics and robotics. They can considerably increase realism of computer-generated movements of characters in films or games and provide solid bases to develop versatile humanoid robots. However, locomotion control has been known to be a notorious problem because of the complexity of the its dynamics and simulated models. This thesis addresses the problem and aims to build controllers for physically simulated humanoids that generate realistic human locomotion. The controllers need to satisfy several important requirements such as human-likeness, richness, and resilience. Before taking up the main subject, we will briefly discuss the meaning of each keyword in the title.

Locomotion is a self-propulsion action of animals to move. In nature, it is a fundamental skill for almost animals, because they should move to find food, find a mate,

escape from dangerous situations and fulfill many other needs. A human also needs to move to other places to satisfy social needs. Likewise, locomotion is an essential skill for artificial human-like creatures such as humanoid robots or virtual characters in games. They basically have to move to somewhere to perform given tasks or avoid enemies in front of them.

Today, **physics simulation** has been increasingly used to reproduce human movements by humanoids to create more realistic motions and interactions using a mathematical model based on the laws of physics. In this approach, a human body is generally modeled as a set of rigid bodies, joints connecting them, and actuators moving them (e.g., joint motors, artificial muscles). Actuation input signals like joint torques drive the humanoid to change its pose and move.

Physics simulation of human motion requires a **control** because the human motion is actively actuated, dissimilar to passive motion of free-fall objects or water surface. Generally, control means guiding or regulating a dynamical system or a device to lead to desired behaviors. For example, a human can perform actions without losing its balance through interactions with environments by subtle manipulation of muscle forces. Similarly, a humanoid also requires subtle manipulation of actuation input signals to perform its tasks. Then the question is; how do we determine the actuation input signals at every moment? The answer implies the control. We have to specify time-varying actuation input signals for our humanoids to perform tasks without losing balance, meaning that we have to "control" the humanoids.

In remainder of this chapter, we will discuss motivations of our research from three

academic domains; computer graphics, robotics and biomechanics. Then we will explain the problem addressed in this thesis and our approach in detail. Lastly, outline of this thesis will be described.

1.1 Motivation

1.1.1 Computer Graphics Perspective

The advancement of three-dimensional computer graphics technologies realized highly realistic virtual world and characters before our eyes, which had existed in our imagination in the past. Now, it is easy to find outcomes of computer graphics technologies around us at every moment (e.g., visual effects in commercial advertisements, large-scale rendered scenes in films). With all these changes, computer graphics technologies have led to the greatest innovation to the entertainment industry such as the movie industry or the game industry. For example, three of the top ten worldwide highest-gross films of 2013 were fully 3D rendered animated feature films [2], which have never existed 20 years ago. The game industry actually had not even existed before development of computer graphics technologies, which is now estimated to be worth as much as $93 billions in 2013 [23].

As main characters of a film or a game play an important role to convey its story and empathy to their audiences, virtual characters are required to be more realistic to make people get immersed in the story. Although both high quality appearance and movements are necessary for realism, a sense of realism of virtual characters largely

comes from how much their motion is realistic because of our innate susceptibility to human motion. Moreover, virtual characters should be able to exhibit a diverse set of movements and variety of different styles for each movement, like a human. For example, locomotion styles of humans depend on what emotion they feel, what situation they are in or what cultural background they are living in.

The advances of motion capture systems allowed us to obtain high quality three-dimensional motion data of human actors without much difficulty. The captured motion data contains subtle details such as mood and emotion of a human actor. However, the context of the motion data are intrinsically limited to the environments and subjects of the motion capture sessions. To address this problem, animation researchers have developed a number of data-driven animation techniques such as motion retargeting, motion blending and motion graphs to overcome the limitation and improve reusability of captured motion data. Because of their high quality results with relevant technical improvements, motion capture systems have been extensively used in production of feature films and commercial games (e.g., *Polar Express* [2004], *Avatar* [2009], *The Hobbit* [2012], *The Last of Us* [2013]).

Physical realism is another crucial element for the realistic feeling of virtual characters, because every movement in the real world is governed by the physics laws. For example, suppose a virtual character that is demonstrating some parkour skills with motion capture data. No matter how realistic the captured motion is, the character will seem to be unrealistic if its feet or hands penetrate into the ground or walls.

The methods to generate physically realistic motions are roughly categorized into two

types; trajectory optimization and physics simulation with a controller. The trajectory optimization optimizes the motion to be physically correct with proper objectives and modeling of contacts and collisions. It generates realistic motions, but it is difficult to be interactive because a motion trajectory for a long duration should be optimized. In contrast, the physics simulation automatically generates physically correct motions. It can enhances user experience with interactivity because it produces simulated movements on-the-fly. A well-designed controller are required for the physically simulated characters to perform the tasks while maintaining balance. However, developing such a good controller is cumbersome in this method.

1.1.2 Robotics Perspective

The Industrial Revolution promoted a development of a modern concept of autonomous artificial creatures with many breakthroughs notably in mechanics. Since the term *robot* was first introduced in a 1921 play *R.U.R.* [7] by Karel Čapek, people have tried to create humanoid robots that resemble the human body. The first modern use of robots has been industrial robots that can perform repetitive tasks including assembly, painting, packaging in manufacturing plants. The most representative form of them is a robotic arm, inspired by human arms. Full-body humanoid robots have been developed mainly for research purpose. They have been developed to understand and reproduce basic human skills (e.g., walking, running, holding an object). Most developed humanoid robots such as *ASIMO* [2000] by Honda, *HUBO* [2005] by KAIST, *DEXTER* [2007] by Anybots, *HRP-4C* [2009] by AIST, *PETMAN* [2011]

by Boston Dynamics are commonly equipped with basic locomotion skills with some other features. Although full-body humanoid robots currently have limited practical use, they can be much more useful in the near future.

There are several practical purposes of humanoid robots. People expect them to accomplish severe tasks instead of human workers by interacting with human tools. The DARPA Robotics Challenge which aims to "develop ground robots capable of executing complex tasks in dangerous, degraded, human-engineered environments" [15] is a good future example of this purpose. Humanoid robots could be useful for military purposes due to their locomotion mechanism and interactivity with human tools. They also can provide services for people due to their familiar appearance. For example, a female robot *Ursula* by Florida Robotics walks, sings, dances for visitors at Universal Studios.

Humanoid robots also have scientific purposes. One of the most important scientific purposes would be understanding human body and human movements. Imitating human body and human movements with humanoid robots helps researchers to better understand of them. Understanding of the human facilitates to build humanoid robots simultaneously.

Humanoid robots requires the controller to conduct various tasks, because they have to handle given tasks while responding to unexpected noise in sensors and actuators, and slight changes in environments. Physics simulation is essential for developing such a controller, because it enables developers to design the structure of the robot and verify their control algorithm with much lower cost.

1.1.3 Biomechanics Perspective

The first modern approach on the study of human movements was begun with the development of photography. Muybridge, a pioneering photographer in studies of motion, published his well-known work on animal locomotion in 1887 [50] which used stop-motion photographs. Large advances of observation technologies have been promoted analysis and understanding on human movements. Today, researchers can obtain high quality human motion data and ground reaction force data via motion capture systems and force plates without much difficulty. Biomechanics researchers also have investigated the underlying mechanism of locomotion and the structure of a human body to understand human locomotion more intensively. With vast improvements on movement measurement and understanding of the human body, biomechanists have proposed a number of biomechanical models to describe human motion in more principled way, such as the spring-loaded inverted pendulum (SLIP) model [6].

The process of human muscle contraction have been investigated by researchers in physiology and biophysics. Among several types of muscle models, the most popular one in biomechanics is *Hill-type muscle model* proposed by Hill [27, 28]. It reasonably imitates input and output mappings of real human muscles without considering their internal details. With *Hill-type muscle model*, many musculoskeletal model of humans have been developed by biomechanics researchers. The computer simulation techniques allows biomechanics researchers to effectively analyze human movements and test their hypotheses with these models. This approach also allows researchers to

evaluate metabolic quantities such as metabolic energy consumption.

Controller-based approach with musculoskeletal models can offer us a wider range of research opportunities because it allows us to predict movements continuously adapting to a new condition. It can also help us to more understand about how the locomotion had been driven. This has significant practical meaning for medical applications such as understanding and treating muscle disorders. For example, individuals with cerebral palsy suffers from unnatural, energy-consuming gaits (e.g., crouch gait, stiff-knee gait, in-toeing gait) due to abnormal muscle tone, contracture and bony deformities caused by the brain damage during their infancy. Although the damaged brain cannot be restored, gait-improving surgery, that is, single-event multilevel surgery, is performed presently. Single-event multilevel surgery includes tendon lengthening and transfer and correction of any bony deformities(e.g., Tendo Achilles lengthening (TAL), distal hamstring lengthening (DHL), rectus femoris transfer (RFT), femoral derotation osteotomy (FDO)). The controller for musculoskeletal models can be used as a framework when planning the surgery and simulating the outcome.

1.2 Aim of the Thesis

The aim of this thesis is to build controllers for physically simulated humanoids that generate realistic human locomotion. We propose locomotion control systems which include a data-driven control algorithms and designing humanoid musculoskeletal models and their musculotendon actuators. To achieve our goal, we need to overcome challenges as follows.

Locomotion Control. A human body is an underactuated system, meaning that we cannot control the unactuated root directly in the space we are in. Instead, we control the position and orientation of our root in the global coordinate system indirectly via the gravity and the ground reaction force. These "unactuated" degrees of freedom make the system underactuated. Along with a limited range of the ground reaction force (e.g., contact force cannot pull the feet), this indirect control requires harmonious coordination of internal muscle forces. Locomotion control for artificial humanoids is not an easy problem because we still have limited knowledge about the indirect control when walking or running although we have already get used to them. High-dimensional, non-linear, and discretely changing characteristics of a human body dynamics make the problem more difficult.

Human Body Complexity. A human musculoskeletal system is very complex. It consists of hundreds of bones and muscles, and is actuated by about 700 skeletal muscles. We need to design humanoids models to be simulated properly without loss of main characteristics of the human body. Another challenging part is an undetermined nature of muscle actuation problem. Because the number of muscles is greater than the degrees of freedom of a human body, various sets of muscle actuation might cause the same set of joint torques, and results in the same motion. Each muscle has a number of parameters related to force generation such as the maximum isometric force and time-varying moment arm, so finding an efficient muscle actuation is not a simple problem.

1.3 Requirements

The requirements of our locomotion control systems and their challenges are described as follows.

- Human-likeness : Reproducing human-like locomotion behaviors and metabolic quantities such as metabolic energy consumption.
- Richness : Generating a versatile array of locomotion behaviors.
- Resilience : Responding to change of model parameters or environments.

Human-Likeness, Richness. It has been believed to be more challenging to control humanoids to walk and run in a highly natural and human-like manner, because we have not yet perfectly understood human locomotion mechanism and completely replicated the body structure of a human. Generating various styles of human locomotion also has been believed to be a difficult task because it had been largely time-consuming and laborious tasks with traditional tuning approaches for robot controllers.

Resilience. A locomotion controller always have to respond to small variations of internal information (e.g., actuation noise) or external information (e.g., sensory noise, tiny fragments on the ground). To respond wider range of variations, we need a more robust locomotion controller. It is challenging because it aggravates challenges of locomotion control.

1.4 Approach

To deal with the forementioned challenges, we utilize motion capture data of human locomotion. We propose a control algorithm that exploits inherent robustness of the captured locomotion motion. Our simulated locomotion is natural and human-like as it is based on real-human motion data. Once a robust control algorithm is built up, this approach allows us to easily reproduce a wide array of locomotion behaviors if their reference motion data is available. Feedback rules allow the simulated humanoids to respond to various types of changes robustly.

It is difficult to represent human body with real number of bones and muscles because of its complexity. A practical approach would be to represent the human body with simpler representations. We use two levels of human body representations. First, we use rigid bodies and joints connecting them as torque actuators. In this model, joints are able to actuated without torque limits whenever the controller indicates. Second, we use musculoskeletal representations which are composed by bones, joints and actuated by musculotendon actuators with activation signals. Each muscle force is generated to be in a similar range with that of the human body. The muscle redundancy issue is handled with optimization techniques to find minimum effort muscle activation signals.

In this thesis, we propose locomotion control systems that exploit observable and measurable data of a human. Our approach utilize observed human motion data collected by motion capture systems, and reconstruct measured physical and physiological properties of a human body in the simulated environment. It allows simulated

humanoids to walk and run in various styles while maintaining balance and interacting with the environment. Our controllers are resilient to the change of humanoids, environmental properties, or objective functions. They generate reasonable adapted locomotion motions with these changes.

We concretize our approach with two controllers; a walking controller for humanoids actuated by joint torques, and a walking and running controller for humanoids actuated by musculotendon actuators. For the first controller, we concentrate on a control algorithm. For the second one, we concentrate on modeling musculoskeletal humanoids and its force-generating and force-transferring processes. More specific descriptions of our approach are given as follows.

Data-Driven Biped Control. In this topic, we focus on the control algorithm to drive humanoids actuated by joint torques. We exploits human motion data to facilitate human-like locomotion control with inherent robustness of human locomotion itself. Our dynamic controller takes reference motion capture data and generates a set of joint torques for each time step that reproduce human-like walking simulation similar to the reference motion. The idea is continuously modulating the reference motion for even a simple tracking controller to reproduce the reference motion. A number of existing data-driven techniques such as motion blending, motion warping, and motion graph can facilitate a biped controller with this framework. Examples include walking, turning, spinning of simulated humanoids, and adapted walking simulation for variation of simulated humanoids and the environment.

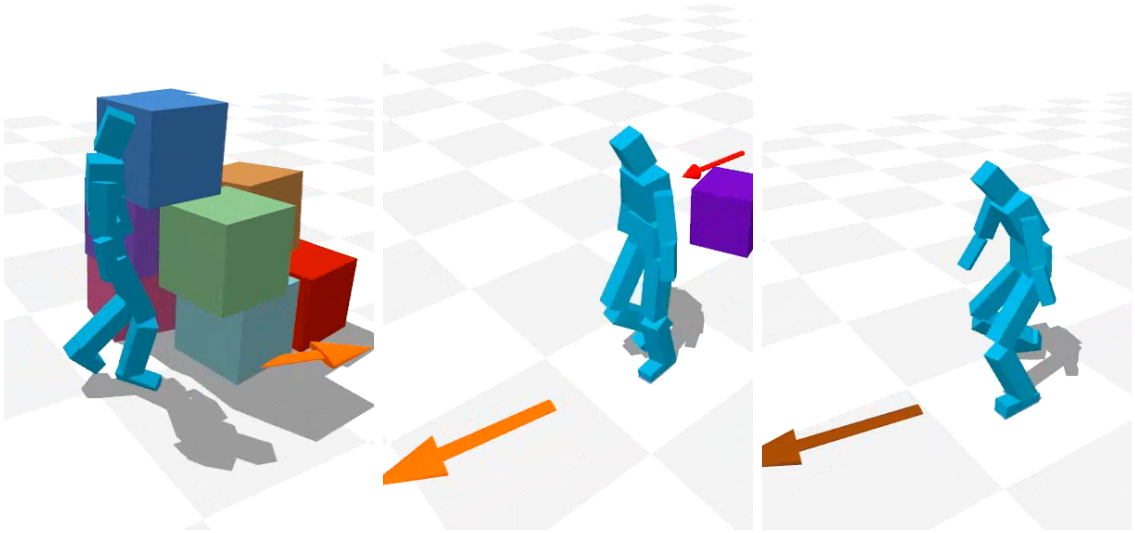


Figure 1.1: *Data-driven biped control. The humanoid can be controlled interactively while changing directions, walking styles and interacting with obstacles.*

Locomotion Control for Many-Muscle Humanoids. In this topic, we build a locomotion control system including detailed model of the human body and musculotendon actuating process. We reconstruct physical and physiological properties of a human body in a physically-simulated virtual environment. This enables realistic locomotion simulation with actuation ranges analogous to that of real humans. Our controller can control humanoid models with more than a hundred of muscles. It has two major technical components, muscle optimization and trajectory optimization. The muscle optimization computes the optimal coordination of muscle activations and the trajectory optimization actively modulates the reference motion to fathifully reproduce the reference or adapt the motion to meet new conditions. Any specific algorithmic changes are not required for different types of humanoid models because our algorithm is general for the structure of the humanoid.

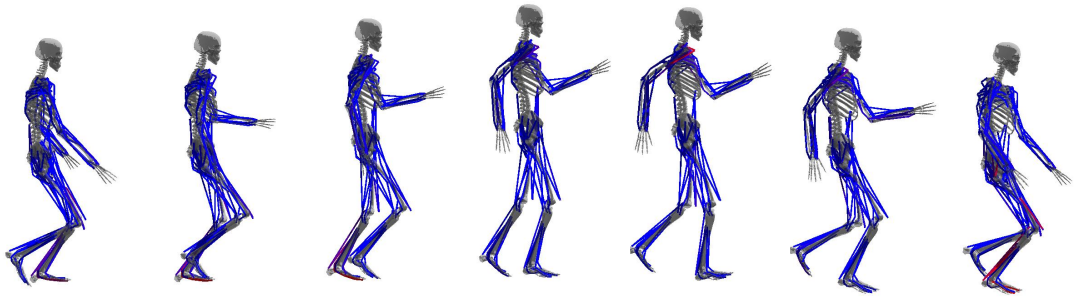


Figure 1.2: *Locomotion control for many-muscle humanoids. This controller can generate various types of locomotion simulations even for the detailed musculoskeletal model with 120 muscles.*

1.5 Thesis Outline

The remainder of the thesis is organized as follows.

Chapter 2 gives a review of previous research in the biped control, and design and simulation of musculoskeletal humanoid models.

Chapter 3 presents the data-driven biped control, a locomotion control system for torque-actuated humanoid models.

Chapter 4 presents the locomotion control for many-muscle humanoids, a locomotion control system for detailed musculoskeletal humanoid models with musculotendon actuators.

Chapter 5 concludes this thesis, and discuss possible future work.

Appendix A describes the mathematical definitions used in Chapter 3.

Appendix B describes the humanoid models used in this thesis.

Appendix C describes the dynamics equations of musculotendon actuators used in Chapter 4.

Chapter 2

Previous Work

In this chapter, we summarize the review of previous work on biped control and musculoskeletal simulation which are relevant to this thesis. In the first section, we review previous biped controllers in computer graphics community. A wide range of work from early biped controllers to recent explosive progress of controllers is discussed in the section. In the second section, we review previous work about simulation of musculoskeletal humanoids. We discuss three categories; simulations of specific body parts, simulation of full-body models, and controllers for musculoskeletal humanoids.

2.1 Biped Control

Biped controllers have extensively been explored in computer graphics and robotics. Hodgins and her colleagues presented manually designed biped controllers for highly-dynamic athletic motor skills, such as running, jumping, and bicycling [30]. Their controllers were equipped with finite state machines for phase transition control and a feedback balancing mechanism based on step placements. van de Panne and his colleagues have extensively studied the design of biped and quadruped controllers. Most notably among them, SIMBICON is a robust, three-dimensional walking controller employing a series of key-poses to shape a reference trajectory and a step-based feedback loop to follow the reference trajectory [80]. Robust walking controllers can also be acquired by approximating bipeds with simplified inverted pendulums, which allow a guaranteed balancing strategy in a closed-form solution [32, 69]. Those walking controllers tend to raise their swing feet higher and keep them in the air longer than natural human walking. Their robustness partly comes from the tendency of extended swing phases, which allows more flexibility in step placements for balancing.

Since the early results based on finite state machines and foot placement algorithms [55, 30], researchers have continuously improved the robustness of the controllers. Some of them experimented the basic principles [80, 72]. Others employed simplified models that abstract the human body to mitigate the complexity of full-body model [36, 46, 78, 9, 69]. Motion capture data have also been used as an important component in achieving human-like quality of result simulated motion. Many controllers have used motion capture data to improve the naturalness of simulated gaits [62, 12, 49,

42, 36].

Once a collection of robust controllers are acquired, high-level control over biped behaviors is desired. Faloutsos and van de Panne [21] discussed the precondition and postcondition of individual controllers to make transition between controllers. Integrated controllers equipped with various motor skills have been employed to clear stepping stones [11] and steer through obstacles [10]. da Silva et al. [14] studied a combination of controllers to create inbetween controllers and coordinate the operation of multiple controllers.

In robotics, biped humanoid robots are often driven by oscillatory movement pattern generators described by differential equations. Several researchers have explored a strategy to synchronize reference movement patterns and an actual humanoid via phase resetting [51, 52]. Phase resetting is conceptually similar to our synchronization method, though our controller can deal with realistic motion capture references that necessitate coordinated movements of many actuated joints (upto 42 degrees of freedom in our experiments).

2.1.1 Controllers with Optimization

Optimization has served as a key methodology in biped controller design. Hodgins and Pollard [29] adapted existing controllers to new characters of different scales by searching control parameters via simulated annealing. This type of optimization is a very challenging problem because each controller has a lot of parameters to tune and the objectives are highly non-linear. The continuation method employed by Yin et

al. [79] addressed a difficult controller-adaptation problem by solving a progressive sequence of problems that trace a path from a solved problem to the target unsolved problem. Control policy searching techniques have also been used to learn walking controllers of physical robots [48, 65]. Sok et. al. proposed optimization method for motion capture data to be adapted to simulate planar biped [62].

Optimization has also been a key algorithm for improving the quality and robustness of biped controllers under various conditions. Wang et al. [72] optimized SIMBICON controllers to allow more human-like gaits using biomechanically-motivated objective functions. Given a baseline controller, stochastic optimization techniques have been successfully applied to find controller parameters that make robust and natural results [72]. Controller parameters have also been optimized to adapt to unexpected disturbances or changes of terrains [73, 77]. Liu et al. exploited the technique to learn an affine feedback policy to perform parkour-style terrain runs [44].

Optimization for instantaneous control signal has also been used for designing biped controllers [1, 12, 16, 77]. The method allows controllers to achieve various task goals with properly designed objectives. Such optimizations are usually formulated as quadratic programming problems because they can be solved efficiently online.

It should be noted that our research goal is different than trajectory optimization [43, 22, 58, 71], which attempts to find a specific trajectory of motion minimizing energy consumption subject to user constraints.

2.1.2 Controllers with Motion Capture Data

Several biped controllers have been supplemented with the realism of motion capture data. To circumvent the difficulty of balance control, some controllers allow only the upper-body to be driven by motion capture data while the lower-body is either fixed or controlled by a conventional balance controller [82, 53]. Data-driven control of two-dimensional biped locomotion was first addressed by Sok et al. [62]. They pointed out that motion capture data are physically inaccurate and rectified motion capture data to make them physically plausible using spacetime optimization. They also demonstrated that even very simple regression and tracking methods can generate stable biped walking, running and jumping when they are combined with physically plausible reference data. da Silva et al. [13] developed three-dimensional walking controllers that exhibit improved robustness and stability. Their controller employed short-horizon tracking and quadratic programming to maintain biped balance. This idea has further been improved with LQR (Linear Quadratic Regulator) balance control, which precomputes optimal balancing strategies using a simplified 3-link character model [12]. Muico et al. [49] employed an even more sophisticated model, NQR (Nonlinear Quadratic Regulator), to track the full DOFs of a human body model. Their controllers coped with non-penetration constraints by incorporating an NQR formulation into a linear complementary problem.

2.2 Simulation of Musculoskeletal Humanoids

2.2.1 Simulation of Specific Body Parts

Many researchers in computer graphics have designed musculoskeletal models and simulate them in physically based simulation. There have been muscle models designed for specific body parts, for example, torso [20], neck-head-face [41], hand [64], face [60] and upperbody [66, 40]. These models are actuated by muscle forces to generate anatomically and physically realistic simulated motions. A simplified form of Hill-type model has been used to determine actuation forces based only on the force-length relations of muscle fibers ignoring the time-derivatives of muscle lengths [60, 64]. Lee et al. used another form of simplified Hill-type model using a linearized force-length-velocity curve [41, 40]. Their coactivation control method exploits the linearity to resolve the redundancy of muscle activations approximately. A more realistic nonlinear force-length-velocity curve is used in torso simulation [20]. Komura et. al. used more realistic full body model using data of [19] with variable tendon length and simulated kicking, standing-up, walking motion [34]. Sapio et al. discussed operational-space feedback control for musculoskeletal simulation using a nonlinear force-length-velocity curve [17]. These studies assumed musculotendon actuators with constant-length tendons or no tendon at all to reduce the complexity of the muscle dynamics model. Several studies used volumetric representations of human muscles to improve appearance or modeling accuracy [60, 66, 40].

Optimization is broadly used to compute activation [60, 70, 20, 40]. Several researcher

used inverse dynamics to compute joint torque and optimize muscle force to generate torque [34, 40, 64]. Optimization also used to generate training data set as input of artificial neural network [41].

2.2.2 Simulation of Full-Body Models

Biomechanics researchers have studied the simulation of various human activities, such as vertical jumps, walks, and pedaling. Anderson and Pandy [4, 5] computed excitation trajectories for jumping and walking by formulating the problem as an optimal control problem, often called dynamic optimization in the biomechanics community. The excitations for a half cycle of locomotion were simultaneously optimized to minimize the metabolic energy consumption per unit moving distance. Thelen and Anderson [68] computed excitation trajectories to simulate walking by performing optimizations only at the current time instance. In biomechanics, optimization approaches are often classified into two categories: *dynamic* optimization and *static* optimization, which roughly corresponds to *space-time* optimization and *online* optimization commonly used in computer graphics. Our muscle optimization is similar to *static* optimization in the sense that control signals are optimized at every time step. A fundamental difference is that these biomechanics approaches aim at finding a single *optimal trajectory* performing a specific task, and lacks the ability to respond to unexpected disturbances. Our goal is to find an *optimal strategy* having the ability to interactively adapt.

2.2.3 Controllers for Musculoskeletal Humanoids

Recently, graphics researchers have explored the simulation of muscle-actuated bipedal locomotion with Hill-type muscle models. Wang et al. [74] presented a locomotion controller for a simplified musculoskeletal model, which has eight muscles on each leg. The muscle forces were applied only at the degrees of freedoms on the sagittal plane. Geijtenbeek et al. [24] presented another muscle-based controller for various bipedal creatures, in which both muscle routing and control parameters are optimized simultaneously. Mordatch et al. [47] proposed a trajectory optimization method to generate walking, running and kicking motions actuated by muscles. These studies commonly used simplified musculoskeletal models and metabolic energy expenditure to measure efforts.

Chapter 3

Data-Driven Biped Control

We present a dynamic controller to physically simulate under-actuated three-dimensional full-body biped locomotion. Our data-driven controller takes motion capture reference data to reproduce realistic human locomotion through realtime physically based simulation. The key idea is modulating the reference trajectory continuously and seamlessly such that even a simple dynamic tracking controller can follow the reference trajectory while maintaining its balance. In our framework, biped control can be facilitated by a large array of existing data-driven animation techniques because our controller can take a stream of reference data generated on-the-fly at runtime. We demonstrate the effectiveness of our approach through examples that allow bipeds to turn, spin, and walk while steering its direction interactively.

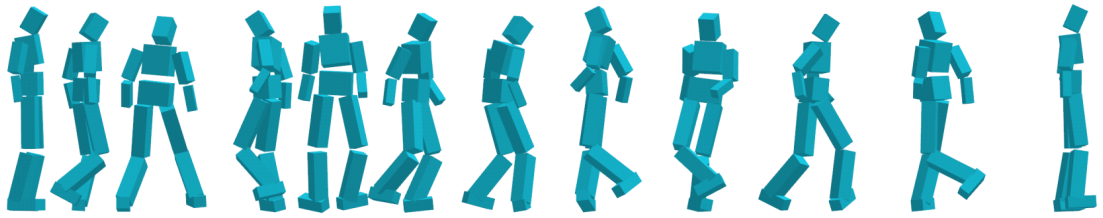


Figure 3.1: *Our data-driven controller allows the physically-simulated biped character to reproduce challenging motor skills captured in motion data.*

3.1 Overview

Physically simulating under-actuated biped locomotion has been a notorious challenge in computer graphics for two decades. Most of early biped controllers were either manually designed and hand-tuned or relying on optimization with energy-minimizing objectives. Though some of those controllers are very robust, they tend to result in stereotyped gaits often looking robotic and lifeless. SIMBICON is an exemplar of manually-crafted biped controllers, which is simple, easy-to-implement, and remarkably robust [80]. Its robustness allowed it to be employed in further challenges such as controller adaptation [79], composition [10] and stepping planning [11]. However, it is also true that SIMBICON generates stereotyped, marching-like gaits.

Recently, motion capture data were employed to achieve natural and realistic locomotion from physically based controllers. Reference-tracking controllers pose yet other challenges such as collecting physically-feasible training data and developing robust feedback control algorithms for the tracking of reference trajectories and the maintenance of balance.

Our goal is to build full-body, three-dimensional locomotion controllers those are as simple and robust as SIMBICON, and still can faithfully reproduce natural and realistic locomotion guided by reference motion capture data. Our data-driven controllers can generate a variety of locomotor behaviors, such as turning and spinning. The key challenge of our approach is modulating a continuous stream of reference data in a seamless way while synchronizing with forward dynamic simulation. Reference-tracking controllers often fail when the swing foot lands on the ground earlier/later than the reference data indicates. Because the ground reaction force is the only source of control to balance under-actuated bipeds, unexpected changes in ground contacts could easily drive the controllers to unrecoverable failure states. In this chapter, we show that carefully synchronizing the reference trajectory and the simulated biped at contact changes in a feedback loop allows us to achieve both the robustness of feedback controllers and the quality of motion capture data simultaneously.

The biggest advantage of our approach is that physically based control can be facilitated by a large array of existing data-driven animation techniques. Our biped controllers are equipped with a data-driven animation engine at the front end. The data-driven engine generates a sequence of movement patterns by editing [39, 33], blending [56], retargeting [25], and composing [38, 35] motion fragments in the database. In this framework, the role of dynamic controllers can be greatly simplified, that is, tracking reference trajectories. We will demonstrate the effectiveness of our approach through examples that allow bipeds to turn, spin, and walk while steering its direction interactively.

Comparing to previous data-driven controllers, our controller requires neither any

precomputed control model (such as regression models, LQR, or NQR) nor non-linear optimization (such as quadratic programming). The model-free approach allows our controllers to take any reference trajectory generated on-the-fly at runtime. It also allows us to decouple the data-driven animation engine and physically based controllers. Therefore, any existing data-driven techniques can be used to actuate and drive physically simulated bipeds without any restriction or precomputation. Our controller does not require the derivative evaluation of equations of motion or a non-linear optimization solver. This makes our controller easy-to-implement and computationally efficient.

3.2 System Overview

Our interactive biped control system consists of three main components (see Figure 3.2): Animation engine, data-driven control, and dynamic tracking control. The animation engine provides the user with high-level control over the behavior of the simulated biped through interactive user interfaces and generates a stream of movement patterns by searching through the motion database. The stream of patterns are fed into the reference motion queue and then consumed by tracking control that drives the biped through forward dynamic simulation. The key challenge is with data-driven control, which continuously modulates the reference trajectory such that even a simple tracking controller can reproduce the reference motion. The role of data-driven control is twofold: Maintenance of balance and synchronization between reference data and the actual simulation.

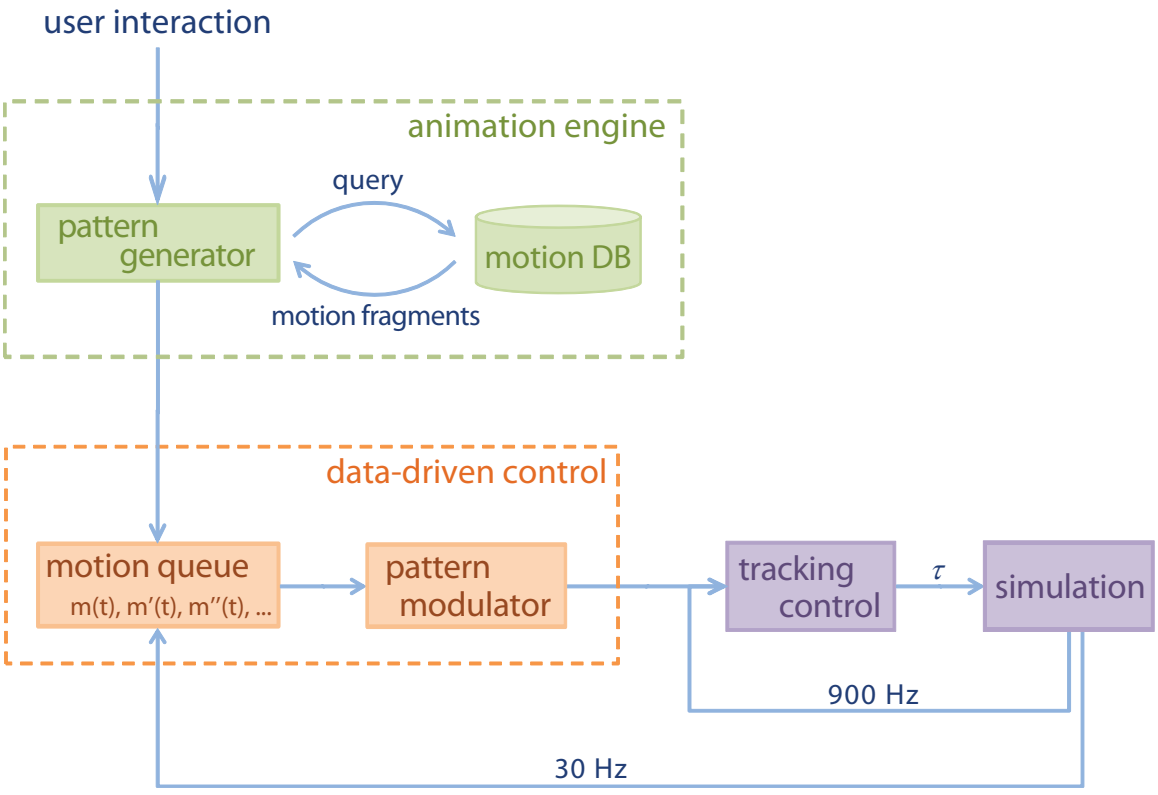


Figure 3.2: *System Overview*

Biped Model. Our biped model has 13 rigid body parts and 12 actuated ball-and-socket joints connecting the body parts and is also a tree-structured model (Figure 3.3, see Appendix B.1 for more details). The total degrees of freedom of the model is 42 including the six degrees of freedom at the unactuated root (pelvis).

The current configuration of the biped model are represented by $\mathbf{P} = (\mathbf{v}^0, \mathbf{q}^1, \mathbf{q}^2, \cdots, \mathbf{q}^{12})$, where $\mathbf{v}^0 \in \mathbb{R}^3$ and $\mathbf{q}^1 \in \mathbb{S}^3$ are the position and orientation respectively, of the root, $\mathbb{S}^3$ is the unit quaternion space, and $\mathbf{q}^k \in \mathbb{S}^3$ for $k = 2, \cdots, 12$ is the relative orientation of joint k with respective to its parent link. The reference frame of local coordinate system of each body part is located at the position of the joint connecting the body part and its parent (Figure 3.3).

The equation of motion of biped model is written as:

$$\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \boldsymbol{\tau} + \mathbf{J}_c^T \mathbf{f}_c, \tag{3.1}$$

where $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\ddot{\boldsymbol{\theta}}$ are the generalized position, velocity and acceleration of all DOFs, $\mathbf{M}$ is the inertia matrix, $\mathbf{c}$ represents the Coriolis, centrifugal and gravitational force. The Jacobian matrix $\mathbf{J}_c$ maps the generalized velocity $\dot{\boldsymbol{\theta}}$ to global velocities at contact points. The $\mathbf{f}_c$ is the contact force and computed by a penalty-based method. The $\boldsymbol{\tau}$ is the joint torques which are the control signal of our system. The forward dynamics simulation engine takes $\boldsymbol{\tau}$ and updates states of the biped. We used Virtual Physics as the forward dynamics simulation engine [31].

Motion Fragments. We annotated motion capture data with ground contact information in a similar way as done by Lee et al. [38] and then segmented motion data

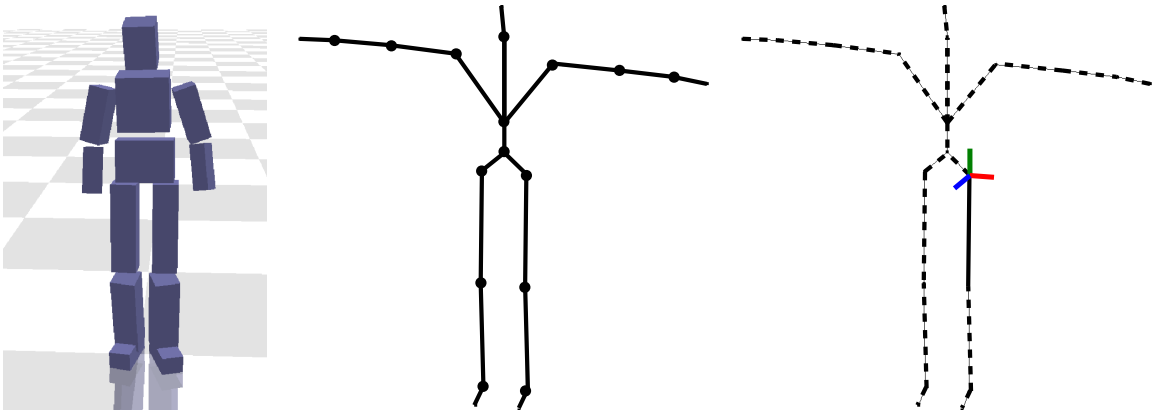


Figure 3.3: *The biped dynamic model has 13 body parts connected by 12 ball-and-socket joints (left and middle). The reference frame of local coordinate system of each body part is located at the position of the joint connecting the body part and its parent. For example, the reference frame of the left femur is located at the left hip joint (right).*

into fragments where ground contact changes. Each fragment contains a half-cycle of locomotion starting from left foot landing to right foot landing or vice versa. Extended double stance phases (e.g., stop to stand still) and flight phases (e.g., broad jump) are also segmented into fragments where double stance/flight begins or terminates. Motion fragments thus obtained are maintained in a directed graph to allow transitioning between them [38].

Motion Representation. The reference motion is represented in the same way with representation of the current configuration of the biped model.

$\mathbf{M}(t) = (\mathbf{v}_r^0(t), \mathbf{q}_r^1(t), \mathbf{q}_r^2(t), \cdots, \mathbf{q}_r^{12}(t))$ is a fragment of motion, where $0 \leq t \leq T$ is the index of motion frames, where $\mathbf{v}_r^0(t) \in \mathbb{R}^3$ and $\mathbf{q}_r^1(t) \in \mathbb{S}^3$ are the position and

orientation respectively, of the root, and $\mathbf{q}_r^k(t) \in \mathbb{S}^3$ for $k = 2, \cdots, 12$ is the relative orientation of joint k with respective to its parent link.

Tracking Control. Tracking control attempts to follow a reference motion trajectory. Our system used a controller similar to Macchietto et al. [45]. The desired acceleration is computed separately for each joint i.

$$\ddot{\mathbf{q}}_{des}^i = k_t^i(d_q(\mathbf{q}_r^i, \mathbf{q}^i)) + k_v^i(\dot{\mathbf{q}}_r^i - \dot{\mathbf{q}}^i) + \ddot{\mathbf{q}}_r^i \tag{3.2}$$

where $\ddot{\mathbf{q}}_{des}^i, \ddot{\mathbf{q}}_r^i \in \mathbb{R}^3$ are the desired and reference angular accelerations of ith joint, $\dot{\mathbf{q}}_r^i, \dot{\mathbf{q}}^i \in \mathbb{R}^3$ are the angular velocities of ith joint of the reference motion and the simulated biped, $d_q(\mathbf{q}_1, \mathbf{q}_2) = log(\mathbf{q}_2^{-1} \cdot \mathbf{q}_1)$ computes the difference between $\mathbf{q}_1$ and $\mathbf{q}_2$ where $\cdot$ represents quaternion multiplication, k_t^i and k_v^i are gain parameters. $\dot{\mathbf{q}}_r^i$ and $\ddot{\mathbf{q}}_r^i$ are estimated from the reference motion data by finite differences. Joint torques are computed from the desired joint accelerations using inverse dynamics and then fed into a forward dynamics simulator to actuate the biped. This simple tracking control is easy-to-implement and stable with small integration time steps. Note that tracking control operates at the rate of 900 Hz for stability, while data-driven control operates at 30 Hz to match the requirement of visual fidelity (see Figure 3.2). We used Virtual Physics to solve inverse dynamics [31]. Because our bipeds are underactuated, we are unable to solve for joint torques those produce desired accelerations at full degrees of freedom via inverse dynamics. The inverse dynamics of an underactuated system takes the desired accelerations at actuated joints and external forces (including ground reaction force) as input, and produces output torques at actuated joints. The accelerations at unactuated degrees of freedom (in our case, linear and

angular accelerations of the root) are passively determined as a result of applying the output torques to actuated joints. Therefore, the unactuated root cannot be directly manipulated through explicit forces/torques, but can only be maneuvered indirectly via harmonious coordination of actuated joints.

3.3 Data-Driven Control

Controlling a dynamic biped model to imitate biological locomotion captured from a live actor is difficult because of many reasons. At first, the dynamic model has fewer degrees of freedom than the actual human skeleton and idealized ball-and-socket joints are different than human joints. The physical properties, such as mass and inertia, are roughly estimated based on statistical data. Motion capture data include measurement errors in estimating skeletal movements from markers placed on deforming skin. On the other hand, forward dynamics simulation is sensitive to input conditions and external perturbation. Tracking control of under-actuated bipeds is particularly susceptible to even small deviation in ground contact from the reference trajectory.

Our data-driven controller modulates the reference motion capture data actively and continuously at runtime to compensate for the discrepancy between the desired reference motion and the actual simulation of a biped. Specifically, data-driven control modulates lower limbs to actively maintain balance. It also adjusts the timing of motion to synchronize the reference data to the simulation.

3.3.1 Balancing

Human balance behavior heavily relies on the hip joints and the stance ankle. As pointed out by Wang et al. [72], the knees are often near-passive throughout the cycle of natural human walking. We apply SIMBICON-style feedback control laws to the hips and stance ankle.

Consider the reference motion fragment at top of the fragment queue at runtime. Motion frames in the queue should be continuously modulated before consumed by tracking control to compensate for possible loss of balance. Let $\mathbf{M}(t_c)$ be the currently referencing motion frame by the tracking controller. The current pose $\mathbf{P}$ is supposed to match $\mathbf{M}(t_c)$ in the reference motion, but may deviate in general. For this reason, simply feeding its subsequent frame $\mathbf{M}(t_c+1)$ to tracking control would not guarantee stable and precise simulation. Instead, we compute an error-compensating, balance-recovering target pose $\hat{\mathbf{P}}$ at every time instance by applying feedback control laws. It guides tracking control to better follow the reference motion. A continuous stream of target poses computed based on motion capture reference data allows our tracking control to be much less stiff than SIMBICON, which uses PD servos with sparse key poses [80].

The target pose $\hat{\mathbf{P}}$ is constructed in three steps starting from the corresponding reference frame $\mathbf{M}(t_c + 1)$. We first decide its stance hip angle with respect to its pelvis orientation and then elaborate on the swing hip and the stance ankle to yield balance feedback. The swing leg is further adjusted for better tracking.

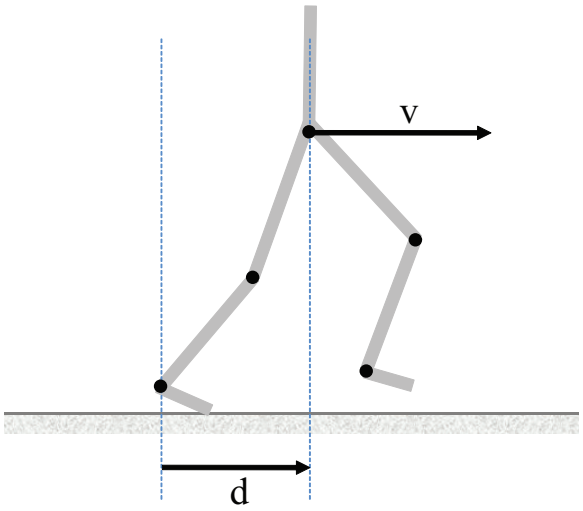


Figure 3.4: *Feedback parameters d and v in the sagittal plane. The center of the pelvis is used as a proxy of the center of mass.*

Feedback on Stance Hip. For an under-actuated system, we do not have a direct control over its root (pelvis), but it can be controlled indirectly by modulating the stance leg. Let $\mathbf{q}_{\text{pelvis}} \in \mathbb{S}^3$ and $\mathbf{q}_{\text{hip}} \in \mathbb{S}^3$ be the orientation of the pelvis and the stance hip with respect to a global, reference coordinate system. We take pelvis orientation $\mathbf{q}_{\text{pelvis}}$ from reference frame $\mathbf{M}(t_c + 1)$ and hip orientation $\mathbf{q}_{\text{hip}}$ from the current configuration $\mathbf{P}$ to compute the desired hip joint angle $\mathbf{q}_d = \mathbf{q}_{\text{pelvis}}^{-1}\mathbf{q}_{\text{hip}}$. The feedback rule to achieve the desired stance hip angle is:

$$\mathbf{q}_{\text{stance\_hip}} = \mathbf{q}_{\text{sth}}\left(\mathbf{q}_{\text{sth}}^{-1}\mathbf{q}_d\right)^{c_0 \cdot s_{\text{stance}}(t)}, \tag{3.3}$$

where $\mathbf{q}_{\text{stance\_hip}}$ is the stance hip angle of the target pose, $\mathbf{q}_{\text{sth}}$ is the angle of the stance hip at $\mathbf{M}(t_c + 1)$, and c_0 is a feedback gain. Transition function $s_{\text{stance}}(t)$ defined over the stance interval allows this feedback control to engage gradually not to make abrupt thrust at the beginning of the stance phase.

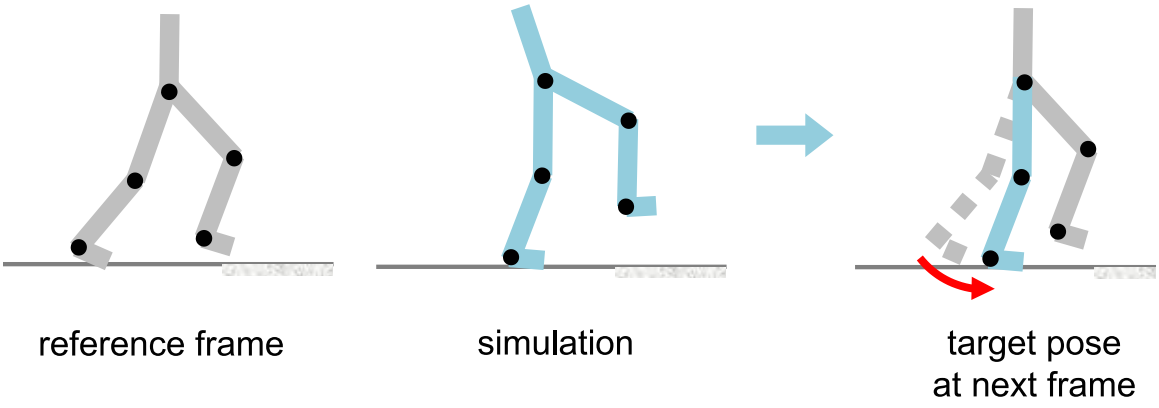


Figure 3.5: *Feedback on Stance Hip. The stance hip affects the balance of the upperbody directly. The target hip angle is first determined to guide the biped to recover its upright pose.*

Feedback on Swing Hip and Stance Ankle. Our feedback rule on the swing hip is similar to the one of SIMBICON, which monitors the location and velocity of the center of mass (COM) to modulate the swing hip angle. We instead use the relative location and velocity comparing to the reference data. We will explain the feedback rule in two-dimensional sagittal plane for simplicity and clarity. The same procedure should be applied in the coronal plane as well for lateral balancing. Let v and d be the horizontal velocity and location, respectively, of the COM with respect to the stance foot position (see Figure 3.4). Let v_d and d_d are their desired values estimated from the reference data. Then, the feedback rule is:

$$\theta_{\text{swing\_hip}} = \theta_{\text{swh}} + \big(c_1(v_d - v) + c_2(d_d - d)\big)s_{\text{swing}}(t), \tag{3.4}$$

where $\theta_{\text{swing\_hip}}$ is the swing hip angle of the target pose, θ_{swh} is the angle of the swing hip at $\mathbf{M}(t_c+1)$, c_1 and c_2 are feedback gains, and transition function $s_{\text{swing}}(t)$ is defined over the swing phase. Similarly, the feedback rule on the stance ankle is defined:

$$\theta_{\text{stance\_ankle}} = \theta_{\text{sta}} + \big(c_3(v_d - v) + c_4(d_d - d)\big)s_{\text{stance}}(t), \tag{3.5}$$

where $\theta_{\text{stance\_ankle}}$ is the stance ankle angle of the target pose, θ_{sta} is the angle of the stance ankle at $\mathbf{M}(t_c+1)$, and c_3 and c_4 are feedback gains. Intuitively speaking, if the current speed is faster than the reference suggests ($v_d < v$) or the biped leans forwards ($d_d < d$), the biped slows down by extending the stance ankle and landing the swing foot forward farther than the reference trajectory indicates. Conversely, if the current speed is slower than the reference ($v_d > v$) or the biped leans backwards ($d_d > d$), the biped accelerates by bending the stance ankle and landing the swing foot closer.

The sagittal and coronal planes change rapidly for turning and spinning motions and sometimes this can be a source of instability. We used a vertical plane containing a moving direction vector and its perpendicular vertical plane, instead of sagittal and coronal planes, to deal with rapid rotational movements.

Feedback for Swing Foot Height. The simulated biped easily loses its balance when its swing foot mistakenly touches the ground. Our controllers modulates the height of the swing foot from the ground surface with a feedback rule:

$$h_{\text{swing\_height}} = h_{\text{swh}} + \big(c_5(h_d - h) + c_6(\dot{h}_d - \dot{h})\big)s_{\text{swing}}(t), \tag{3.6}$$

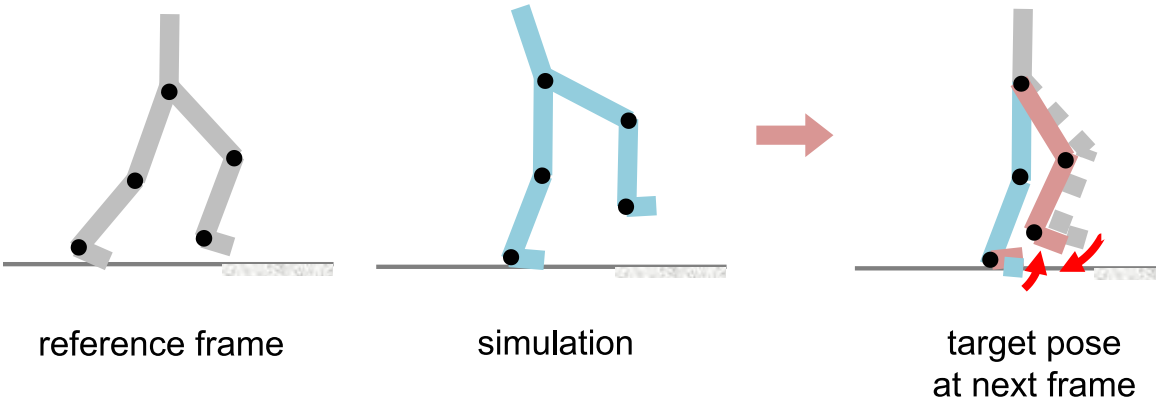


Figure 3.6: *Feedback on Swing Hip and Stance Ankle. After adjusting the stance hip angle, we adjust the swing hip and the stance ankle to make a narrower step. It helps the biped to lean forward.*

where $h_{\text{swing\_height}}$ is the target height of the swing foot, h and $\dot{h}$ are the current height and its time derivative, h_d and $\dot{h}_d$ are their desired values estimated from the reference data, and c_5 and c_6 are feedback gains. Given the target height, we used an inverse kinematics solver developed by Lee and Shin [39] to adjust the target pose.

3.3.2 Synchronization

Let $\mathbf{M}(t)$ be a motion fragment at top of the queue and $\mathbf{M}'(t)$ be its subsequent motion fragment awaiting in the queue. Since we segmented motion data at contact changes, there must be a contact change between the two motion fragments. Typically, for locomotion, a swing foot of $\mathbf{M}$ lands on the ground at the beginning of $\mathbf{M}'$.

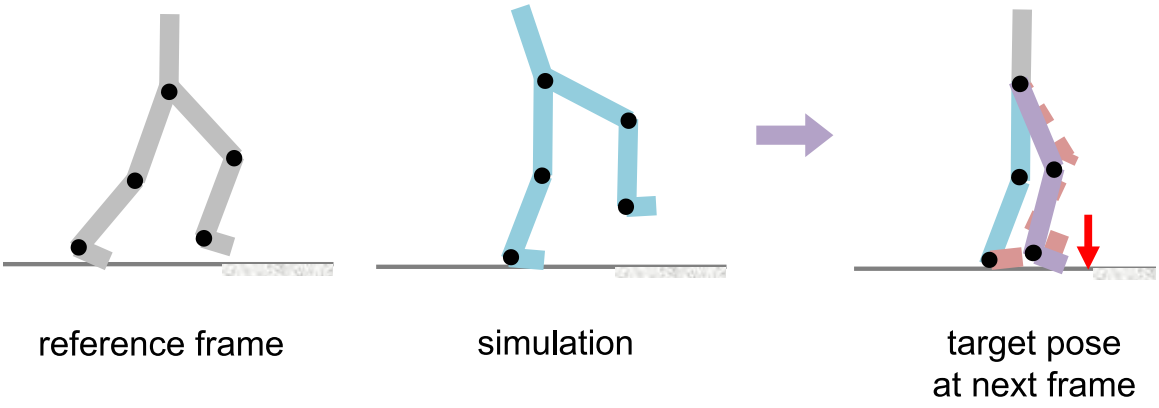


Figure 3.7: *Feedback for Swing Foot Height. Additionally, the biped may adjust the height of the swing foot to match the swing duration of the reference data. This also maintains appropriate ground clearance to avoid inadvertent tumbles.*

In the tracking simulation loop, the swing foot may touch the ground earlier or later than the reference motion indicates even with feedback control. Assume that the tracking controller is currently referencing $\mathbf{M}(t_c)$ when the swing foot is landing. The target pose $\hat{\mathbf{P}}$ computed based on feedback rules may deviate from the reference frame $\mathbf{M}(t_c)$ in general.

Here, we use mathematical notations introduced in [37] to represent motion displacements.

Definition 1. *The displacement between two articulated figure poses can be denoted by an array of linear and angular displacement vectors* $\mathbf{D} = (\mathbf{u}^0, \mathbf{u}^1, \cdots, \mathbf{u}^n) \in \mathbb{R}^{3(n+1)}$.

Primitive operations between poses and displacements are defined:

$$\begin{aligned}
\mathbf{P}_1 \otimes \mathbf{P}_2 &= (\mathbf{q}_1^1 \mathbf{v}_2^0 \mathbf{q}_1^{1-1} + \mathbf{v}_1^0, \mathbf{q}_1^1 \mathbf{q}_2^1, \cdots, \mathbf{q}_1^n \mathbf{q}_2^n) \\
\mathbf{P}_1 \oslash \mathbf{P}_2 &= (\mathbf{q}_2^{1-1}(\mathbf{v}_1^0 - \mathbf{v}_2^0)\mathbf{q}_2^{1-1}, \mathbf{q}_2^{1-1}\mathbf{q}_1^1, \cdots, \mathbf{q}_2^{n-1}\mathbf{q}_1^n) \\
\mathbf{D}_1 \pm \mathbf{D}_2 &= (\mathbf{u}_1^0 \pm \mathbf{u}_2^0, \cdots, \mathbf{u}_1^n \pm \mathbf{u}_2^n) \\
\alpha \cdot \mathbf{D} &= (\alpha \mathbf{u}^0, \cdots, \alpha \mathbf{u}^n) \\
\widetilde{\exp}(\mathbf{D}) &= \big(\mathbf{u}^0, \exp(\mathbf{u}^1), \cdots, \exp(\mathbf{u}^n)\big) \\
\widetilde{\log}(\mathbf{P}) &= \big(\mathbf{v}^0, \log(\mathbf{q}^1), \cdots, \log(\mathbf{q}^n)\big) \\
\mathbf{P}^\alpha &= \widetilde{\exp}\big(\alpha \widetilde{\log}(\mathbf{P})\big),
\end{aligned}$$

where $\alpha \in \mathbb{R}$ *is a scalar value,* $\mathbf{P}_i = (\mathbf{v}_i^0, \mathbf{q}_i^1, \cdots, \mathbf{q}_i^n)$ *and* $\mathbf{D}_i = (\mathbf{u}_i^0, \mathbf{u}_i^1, \cdots, \mathbf{u}_i^n)$. *Intuitively speaking, the "difference" between two poses yields displacement* $\mathbf{P}_1 \oslash \mathbf{P}_2$. *The power* $(\mathbf{P}_1 \oslash \mathbf{P}_2)^\alpha$ *scales the displacement linearly by a factor of scalar value* α. *"Adding" the scaled displacement to pose* $\mathbf{P}$ *yields another pose* $\mathbf{P}' = \mathbf{P} \otimes (\mathbf{P}_1 \oslash \mathbf{P}_2)^\alpha$.

If the actual contact was earlier, the remaining frames of $\mathbf{M}$ is dequeued and the next fragment $\mathbf{M}'$ shifts to the top of the queue. At that moment, $\mathbf{M}'$ should be warped to make a smooth transition.

$$\mathbf{M}(t) \leftarrow \mathbf{M}'(t) \otimes \big(\hat{\mathbf{P}} \oslash \mathbf{M}'(0)\big)^{r(t)} \quad \text{for} \quad \forall t, \tag{3.7}$$

where $\hat{\mathbf{P}} \oslash \mathbf{M}'(0)$ is the displacement between the two poses. $r(t)$ is a smooth transition function, which is one at the beginning of $\mathbf{M}'$, zero at the end, and its derivatives are zero at both ends (see Appendix A.1). Intuitively speaking, the displacement at the beginning of $\mathbf{M}'$ propagates gradually to its subsequent frames to make seamless transition. The transitioning period should be as long as possible to achieve smoothest

visual transition. One exception is the stance foot, which supports the entire body mass. Even a small deviation at the stance foot may influence the fullbody balance significantly. Transitioning of the stance ankle is handled differently than other joints. At first, a quicker transition of the stance foot usually better stabilizes the next stride. In our experiments, transition function $r(t)$ was set to vary from one to zero over the duration of $\mathbf{M}'$ (usually, a half cycle of locomotion) excepting for the stance ankle, which completes its transition in 1/5 of the half-cycle duration. Secondly, the angle of the stance foot with respect to the ground surface is more important than tracking the joint angles. Therefore, the target angle of the stance ankle at the end of the transitioning is set such that the angle between the stance foot and the ground surface matches the reference data.

If the actual landing was later than the reference indicates, there are no reference data to follow until the swing foot touches the ground and the next reference motion engages. We expands the current reference motion by integrating joint angles with constant velocities at the end of $\mathbf{M}$ excepting for the stance leg. Similarly expanding the stance-leg motion tends to make it push off the ground with too much thrust. Therefore, we leave the hip, knee and ankle of the stance leg fixed while the reference motion is expanded.

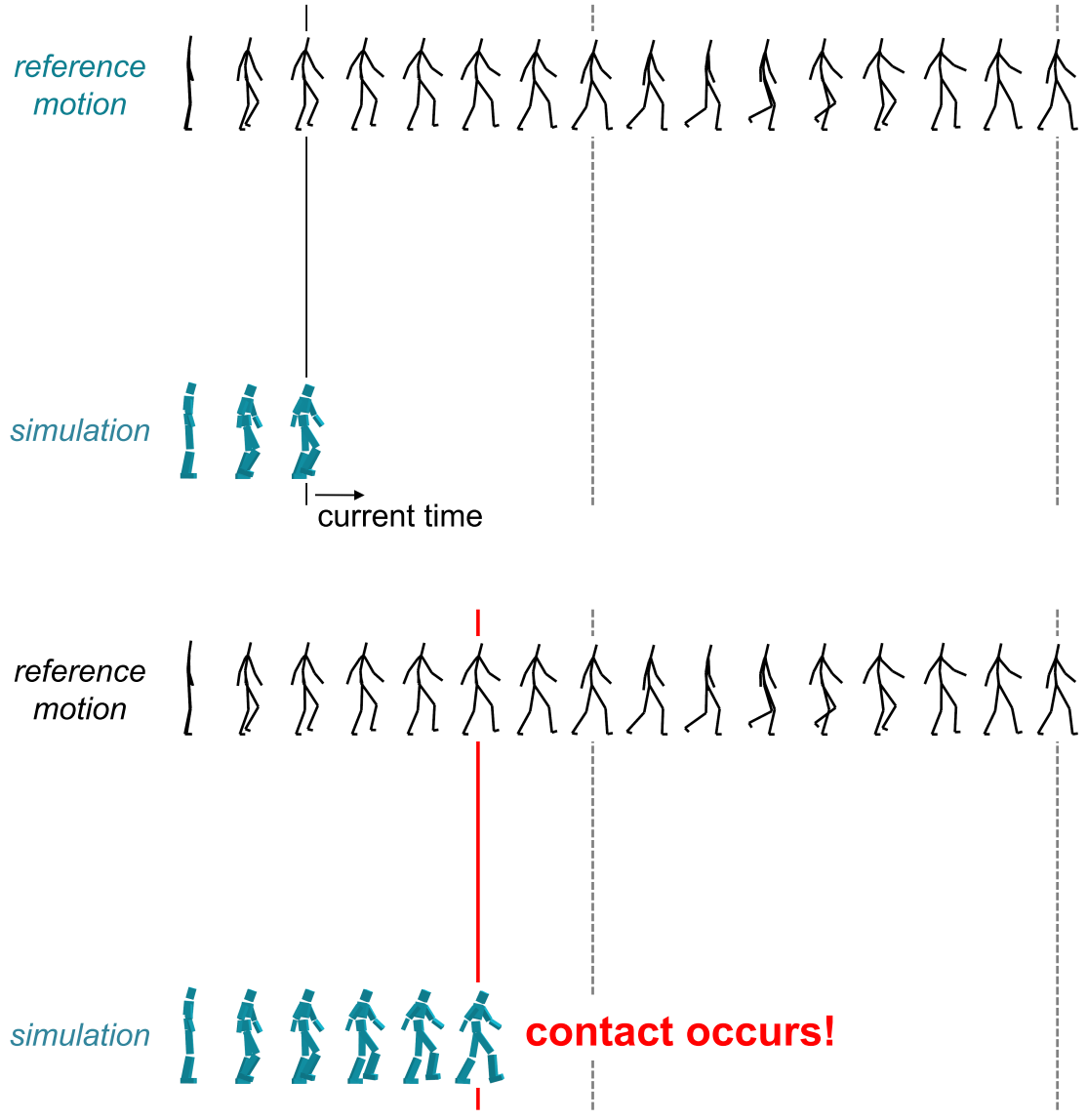


Figure 3.8: *Synchronization in early landing case. In the tracking simulation loop, the swing foot may touch the ground earlier or later than the reference motion indicates even with feedback control. Grey dotted lines show the moments when the swing foot lands on the ground in the reference motion.*

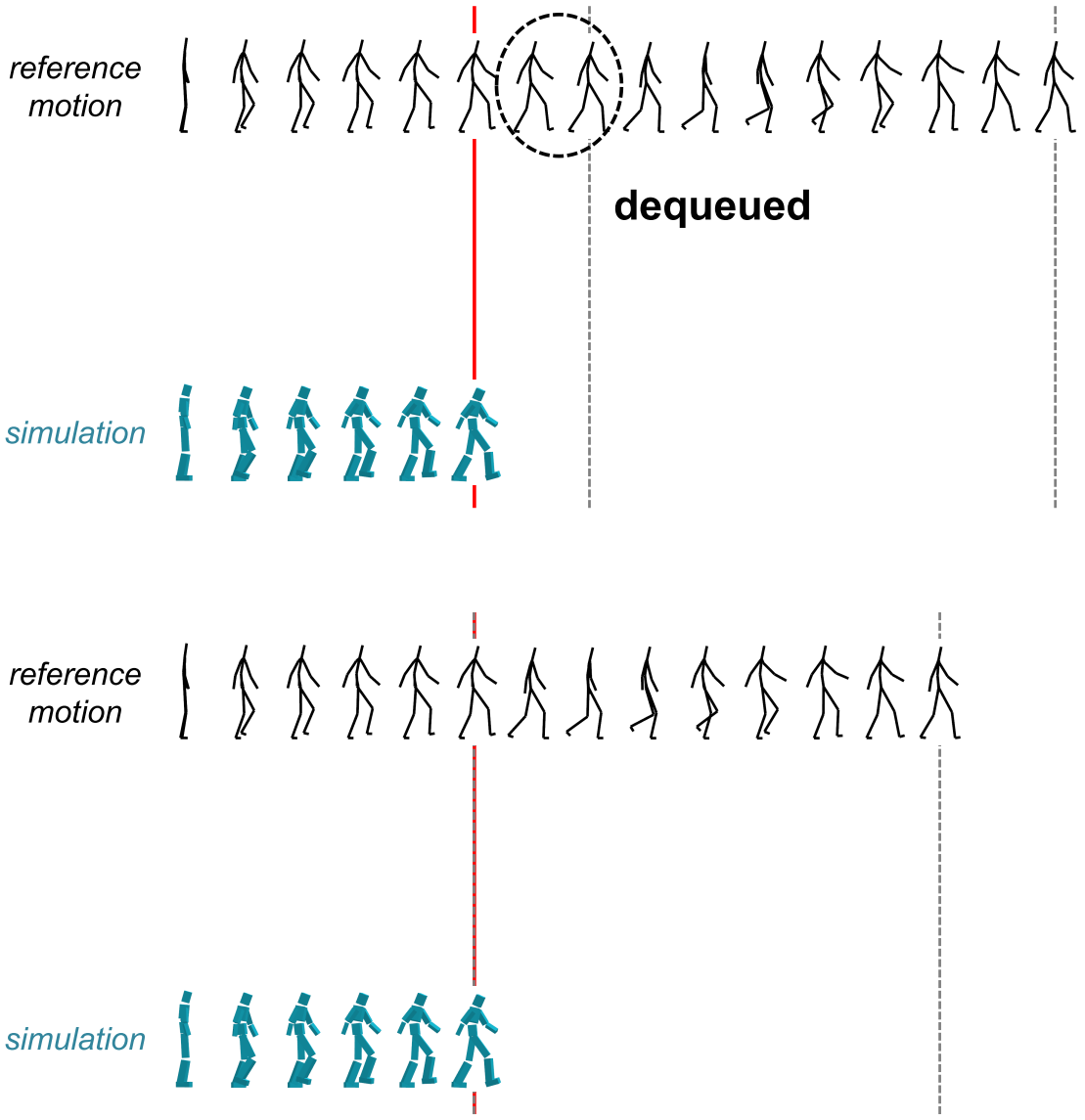


Figure 3.9: *Synchronization in early landing case. When early landing occurs, the remaining frames of the current half-cycle fragment are dequeued.*

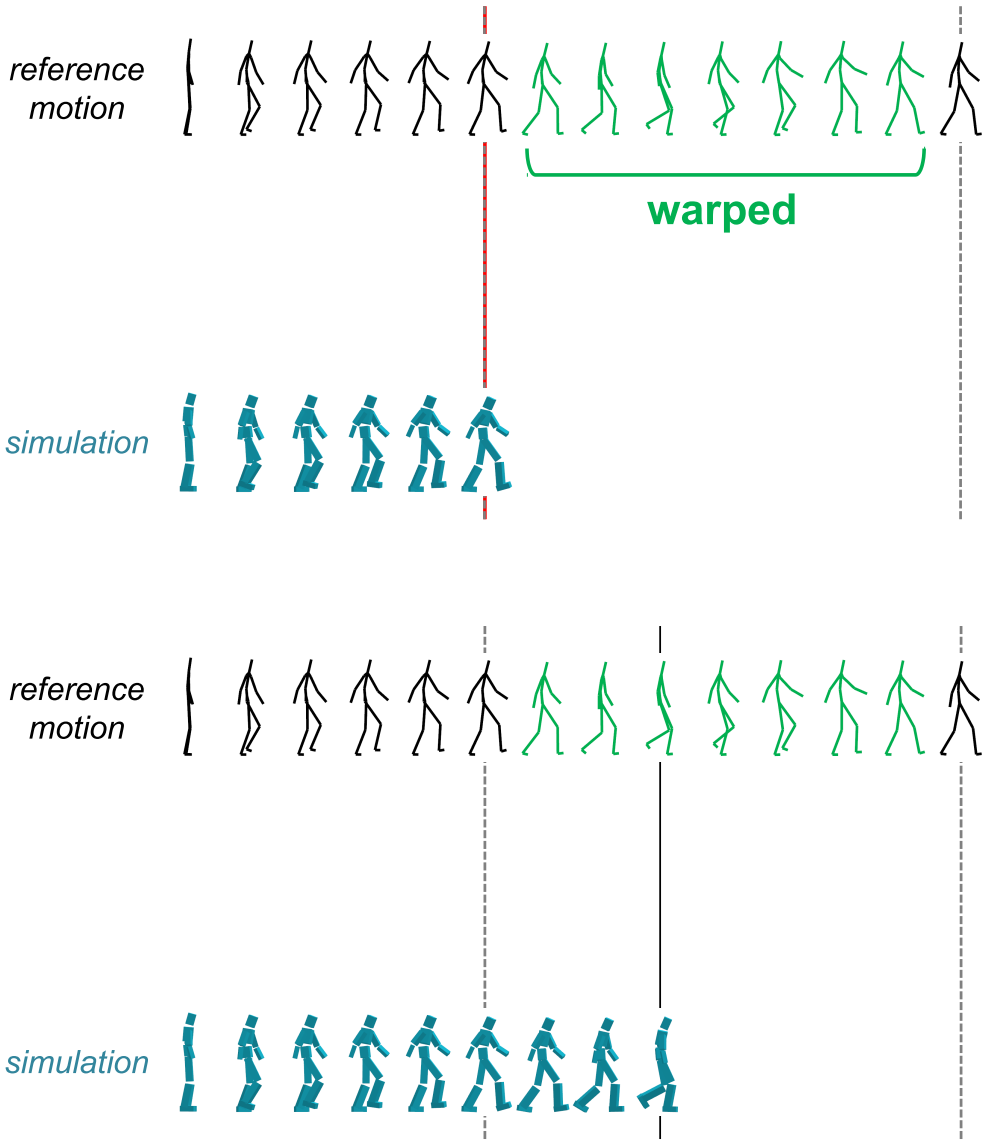


Figure 3.10: *Synchronization in early landing case. After dequeueing, the next fragment is warped to make a smooth transition, and the simulation keeps going on.*

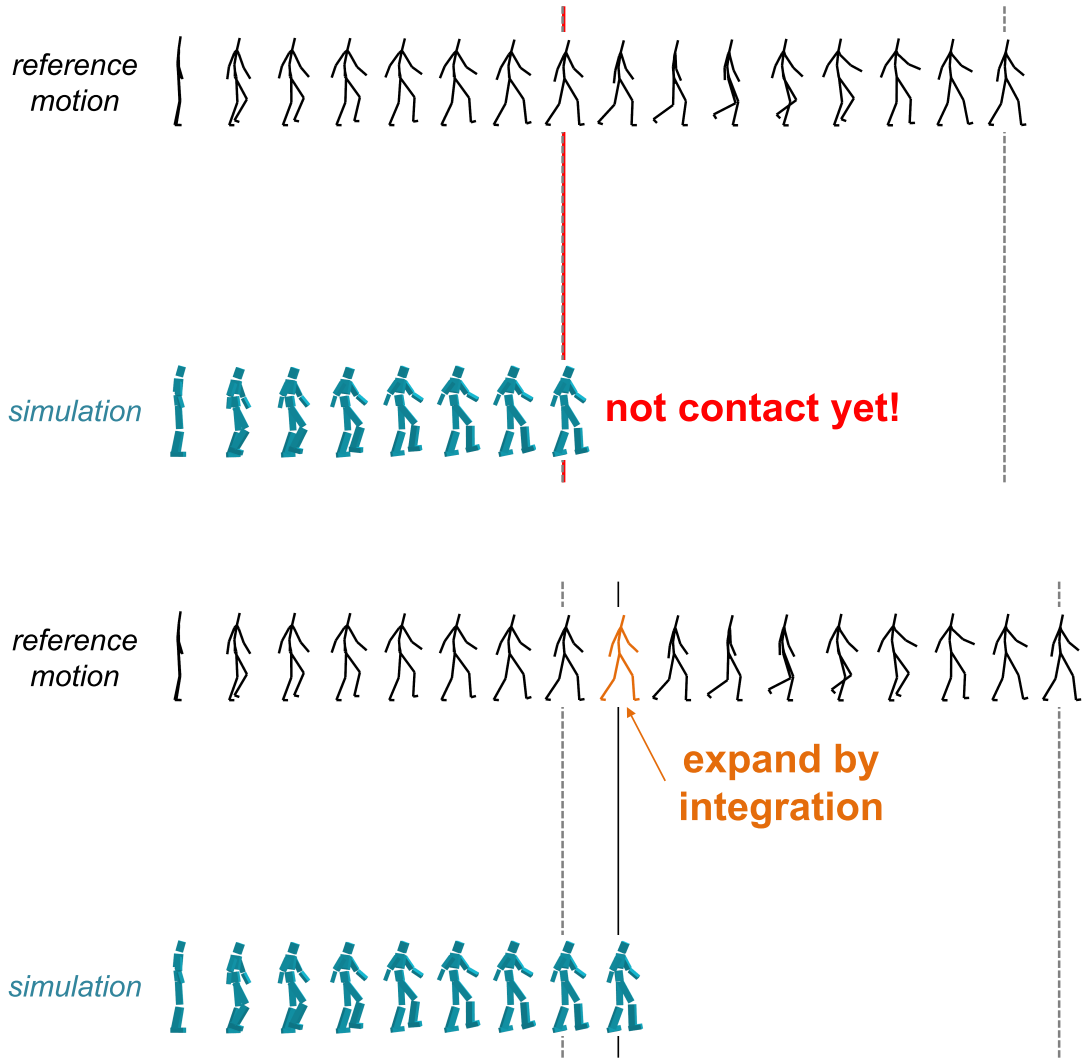


Figure 3.11: *Synchronization in delayed landing case. If the swing foot lands on the ground later than the reference motion, there are no reference frames to follow in this fragment until the swing foot touches the ground. We expands the original reference motion by integrating joint angles with constant velocities at the end.*

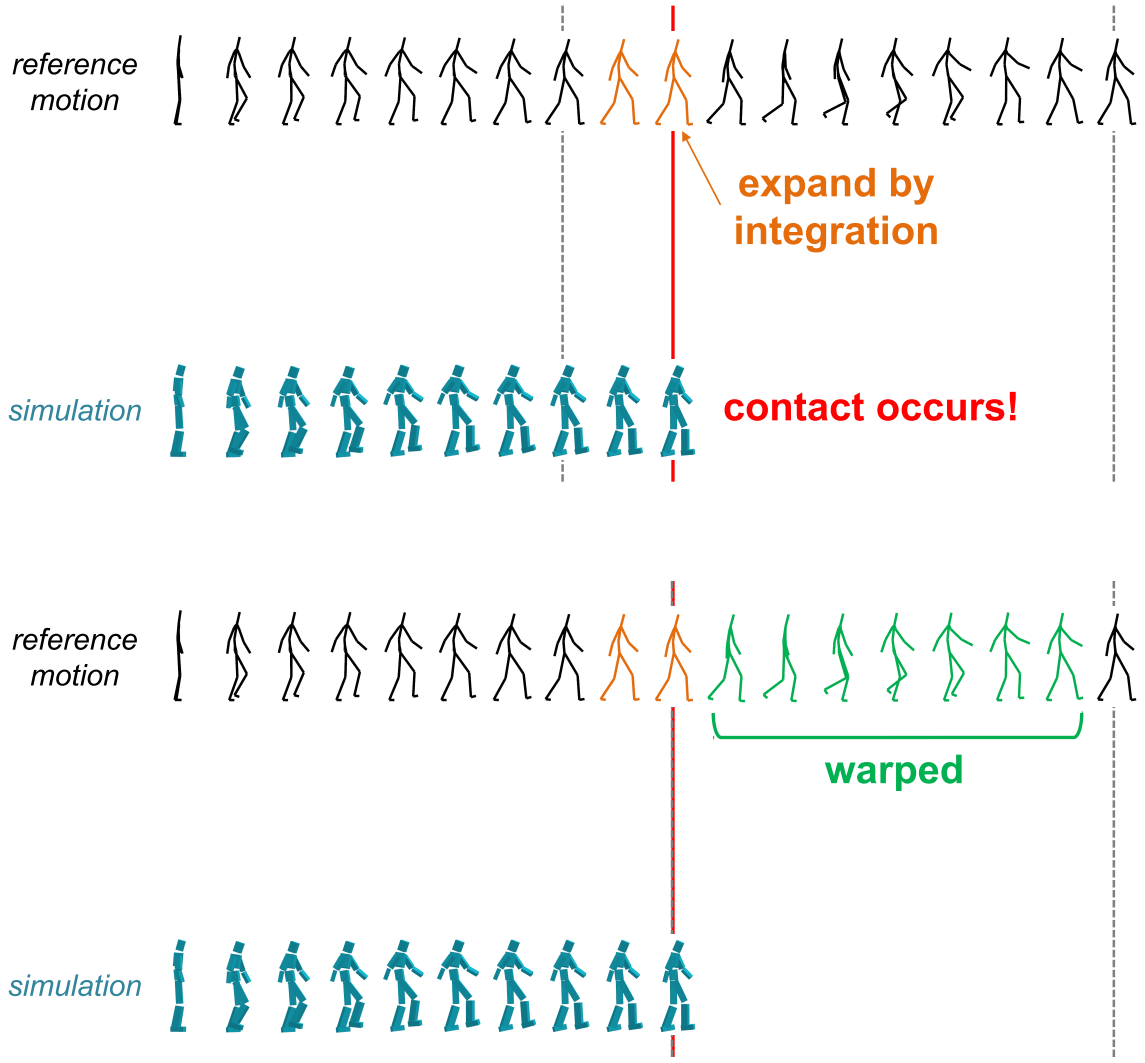


Figure 3.12: *Synchronization in delayed landing case. If the contact occurs, the next fragment is warped to make a smooth transition.*

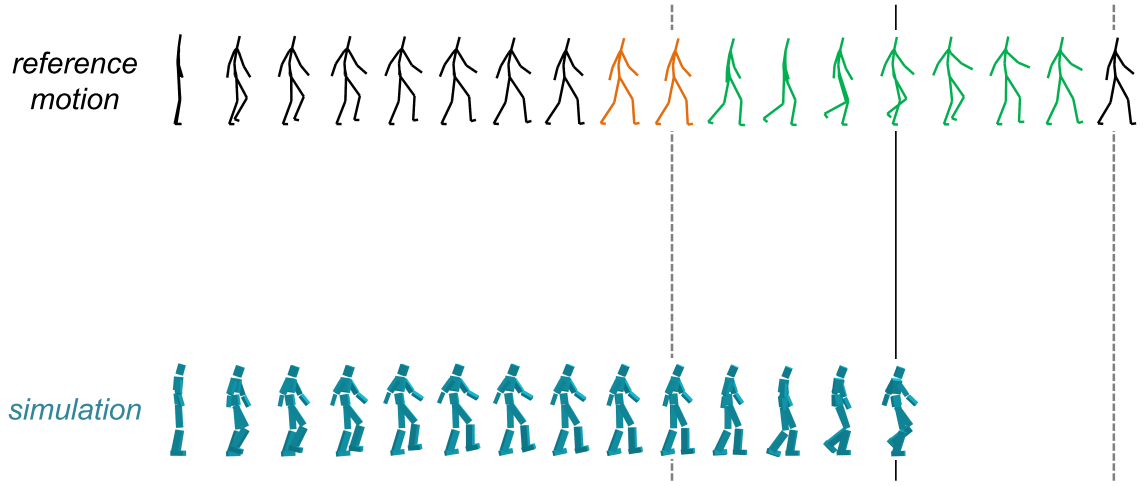


Figure 3.13: *Synchronization in delayed landing case. After warping to make a smooth transition, the simulation keeps going on.*

3.4 Results

All motion data in our experiments[1] are from SNU motion database [61] and were originally captured using a Vicon optical motion capture system at the rate of 120 frames/second and then down-sampled to 30 frames/second. Motion data include walking in a variety of different speeds, turning angles and styles. It also includes sharp U-turning and spinning (see Figure 3.14 for more details). We lifted up the swing foot trajectory slightly in the motion data of turning, spinning and interactive control examples which have low step height. We set dynamics and integrator parameters to achieve robust simulation in a conservative manner while maintaining the performance of realtime simulation at the rate of 30 frames/second. The ground reaction is modeled

[1] You can download all motion data from `http://mrl.snu.ac.kr/research/ProjectDataDrivenBiped/DataDrivenBipedControl_2010_motion.zip`

	c1_sag	c1_cor	c2_sag(+)	c2_sag(-)	c2_cor	c3_sag	c3_cor
Walking							
Walk Forward Normal	0.05	0.2	0.2	0.05	0.2	0.1	0.1
Walk Forward Slow	0.05	0.25	0.5	0.05	0.2	0.1	0.1
Walk Forward Fast	0.05	0.3	0.5	0.05	0.2	0.1	0.1
Walk Forward Gentle	0	0.3	1.2	0.05	0.2	0.1	0.1
Walk Wide Swing	0.05	0.2	0.3	0.05	0.2	0.1	0.1
Walk Brisk	0	0.3	0.2	0	0.2	0.1	0.1
Walk March	0	0.3	0.2	0.05	0.2	0.1	0.1
Walk Backward	0.1	0.3	1	0	0.3	0.3	0.3
Spinning, Turning							
Walk Left 45 Degree	0.1	0.4	0.7	0.05	0.2	0.1	0.1
Walk Left 90 Degree	0	0.3	0	0	0.2	0.1	0.1
U-turn	0.05	0.3	1.2	0.05	0.2	0.1	0.1
Spin	0.2	0.2	0.7	0.05	0.2	0.1	0.1
Robustness to Pushes							
Walk Forward Normal	0.05	0.25	1.7	0.1	0.3	0.1	0.1
Interactive Control							
Stop to Walk	0.1	0.25	1	0.05	0.3	0.1	0.1
Normal Walk	0	0.3	1.8	0	0.2	0.1	0.1
Left / Right Turn 90	0.1	0.3	1.5	0.05	0.4	0.1	0.1
Left / Right Turn 135	0	0.3	2	0.05	0.4	0.1	0.1
Left / Right Turn 180	0.1	0.3	2	0.05	0.3	0.1	0.1
Fast Walk	0	0.25	1	0	0.3	0.1	0.1
Walk Style1	0.1	0.4	0.5	0.05	0.4	0.1	0.1
Walk Style2	0.1	0.3	0.5	0.1	0.3	0.1	0.1

Figure 3.14: *Motion data in our experiments and their corresponding feedback gains. The gains are different for sagittal and coronal planes. "+" is for* $v_d - v > 0$ *or* $d_d - d > 0$ *and "-" is for* $v_d - v < 0$ *or* $d_d - d < 0$. $c_0 = 1$, $c_4 = 0.1$, $c_5 = 0.5$, *and* $c_6 = 0.02$ *for all examples.*

as a damped spring. The ground spring and damping coefficients are $k_s = 2000N/m$ and $k_d = 2\sqrt{k_s} = 89.4Ns/m$, respectively.

Feedback Parameters. All feedback gains are summarized in Figure 3.14. Parameters were manually tuned for each motion data. Parameter tuning was not formidable because each parameter has an intuitive meaning and many of gains are simply constant for all motion data. Most of motion data can be stably reproduced for a wide range of parameter choices excepting for several very challenging examples, such as spinning, which requires careful parameter tuning (see the accompanied video). Most of our controllers generated stable cycles and became resilient to mild pushes without position feedbacks, that is, $c_2 = c_4 = 0$. Without external perturbation, the velocity feedback alone allows the reference trajectory to be followed closely and thus the position feedback has nothing to do with balancing. However, external pushes would make the simulation to deviate from the reference trajectory and the position feedback would play an important role. Therefore, we first tune c_1 and c_3 to achieve stable cycles without any external perturbation and then tune c_2 and c_4 later in the presence of random pushes at the center of mass.

Locomotion Control. Our biped character is able to reproduce various gaits of human walking (see Figure 3.15). Each motion clip recorded a subject standing still, starting to walk, taking 6 to 8 steps, and then stopping. Representing motion data as a motion graph allows us to produce an arbitrarily long sequence of locomotion by splicing walking steps. Our biped can track an arbitrary combination of locomotion steps including slow/fast walking, turning of different angles, and different gaits.

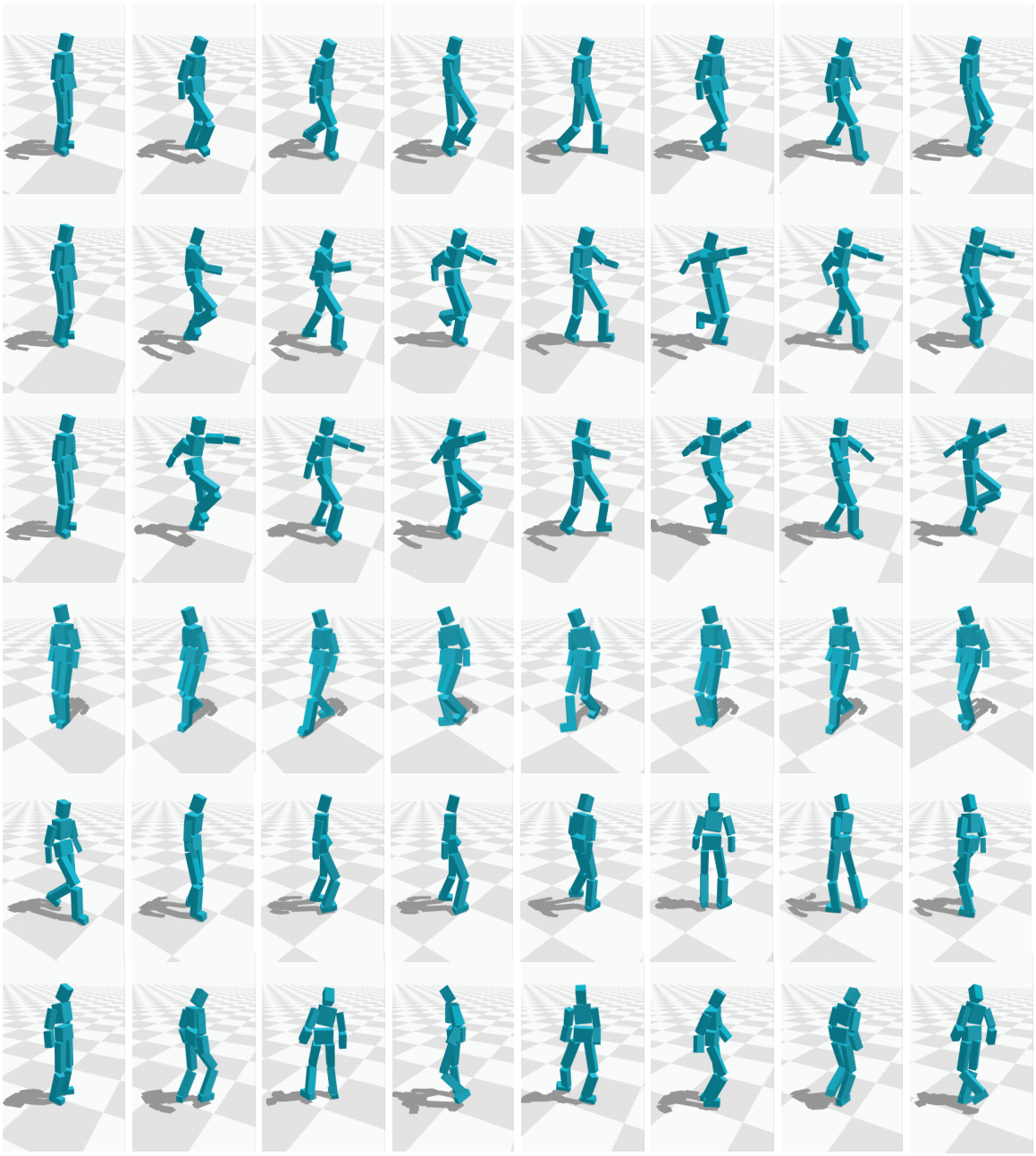


Figure 3.15: *Data-driven biped simulation from motion capture data including (top to bottom) WalkForwardNormal, WalkBrisk, WalkMarch, WalkBackward, U-turn, and Spin.*

Effects of Individual Components. Our controller consists of several components for balancing, tracking, and synchronization. Disabling any of these components would result in either falling over in several steps or the degradation in motion quality. We evaluated the effect of each component by disabling each one at a time for a variety of walking data:

- Disabling synchronization always leads to falling over in 3 to 6 steps.
- Disabling the feedback on a stance hip makes the torso to lean and eventually leads to falling over in 2 to 4 steps.
- Disabling the feedback on a swing hip makes the character to lean to one side and eventually leads to falling over in 6 to 10 steps.
- Disabling the feedback on a stance ankle or swing foot height managed to avoid falling over for some gaits, but the motion looks unnatural.

Robustness under Various Conditions. We tested our walking controller on varied simulation conditions to evaluate its robustness (see Figure 3.16). Our controller generated stable cycles of walking for up to 15 Kg of extra weight on one leg, 50% longer legs, 50% shorter legs, one leg 3% shorter than the other, up and down slopes up to 6 and 4 degrees respectively, 60% to 1200% variations of friction coefficients. These numbers were acquired from the same reference data and the same parameters with those used in walking examples. The limits can be significantly improved if we adapt the reference data kinematically to the varied conditions [39].

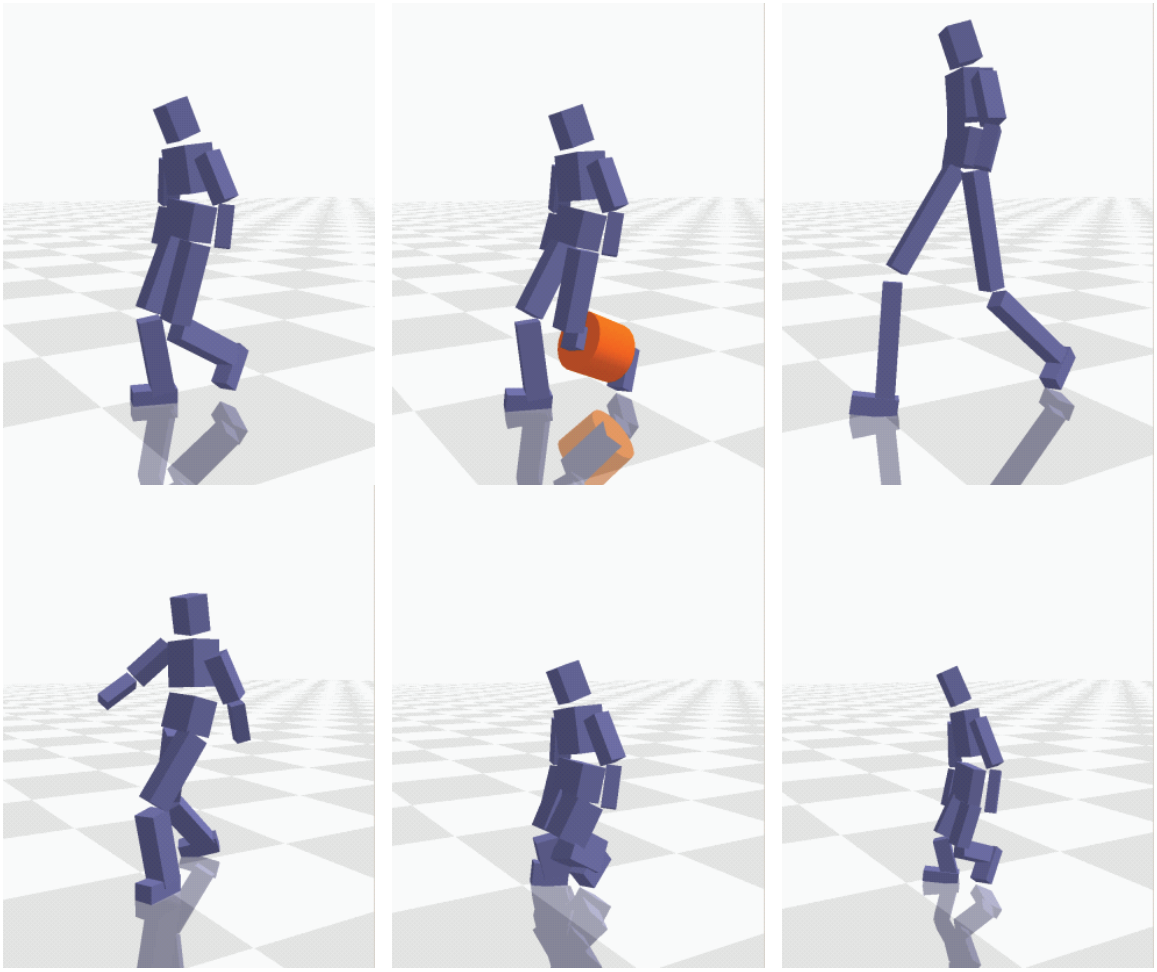


Figure 3.16: *Our walking controller has been tested under varied simulation conditions. (left to right, top to bottom) Original character, extra weight on the leg, longer legs, the left leg shorter than the right leg, shorter height and the same weight, shorter height and lighter weight*

Robustness to Pushes. We quantified the robustness to external disturbances with push experiments similar to Wang et al. [72]. The body mass and simulation coefficients were set to match those of SIMBICON [80] as much as we could. However, we were unable to conduct the comparison test under the exactly same condition. Once the biped entered into stable cycles, we applied forces of 0.4 seconds duration to the center of mass of torso once every 4 seconds for 40 seconds. The controller passes the push-resilience test if the biped is still walking stably after 40 seconds. For Walk Forward Normal data, our controller withstands pushes up to 160N, 130N, 80N and 105N from front, rear, right and left, respectively. Because the robustness is influenced by the size and scale of the body, a type of gaits, walking speed, and many other factors, direct comparison to the previous results would be difficult. Roughly speaking, the results indicate that our controller is about as robust as the controller proposed by Wang et al. [72] and less robust than SIMBICON.

Interactive Control. Our model-free approach allows us to blend a set of motion data on-the-fly and feed inbetween data to the controller. The feedback gains are also interpolated at the same ratio as motion data. The motion set includes turning in 90, 135, and 180 degrees, straight walking at normal/fast speeds and two different gaits. Motion blending and transitioning are computed in the data-driven animation engine and our controller simply tracks a stream of reference data generated at the animation engine. Our biped character can steer in arbitrary turning angles, change its speed, and make transition between different gaits. The user can control the biped character interactively by specifying walking direction, speed and a type of gaits through simple user interfaces. Our controller allows the biped to respond to external perturbations,

such as intentional pushes and collision with stacked boxes (see Figure 3.17).

3.5 Discussion

Biped control requires two essential mechanisms for shaping trajectories and robust balancing/tracking control. Our work is perhaps emphasizing the importance of trajectory shaping. Motion capture reference data allowed our controller to generate realistic human locomotion. Even balance control was achieved in a data-driven manner by modulating reference trajectories. Presumably, advanced control methodologies would improve our work in several directions. Regression-based approaches [62] would allow us to represent natural variations of locomotion in statistical models, which would cope with variations in environments and simulation conditions. Advanced optimal control methods, such as LQR [12] and NQR [49], would allow less stiff systems for tracking control. Even with such expected advantages, we were unable to employ sophisticated control methods because those methods require all reference data be prepared for preprocessing. No reference data generated on-the-fly could be fed into the controller. This restricts the flexibility and versatility of biped control. Designing robust model-free controllers would be an important advance in biped control. We can also think of online model learning that builds a control model incrementally at runtime.

Our controller is more robust if it sacrifices its motion quality by maintaining the direction of the stance foot to match the angle of the ground surface while in contact. It means our controller becomes less robust with more natural stance phases including

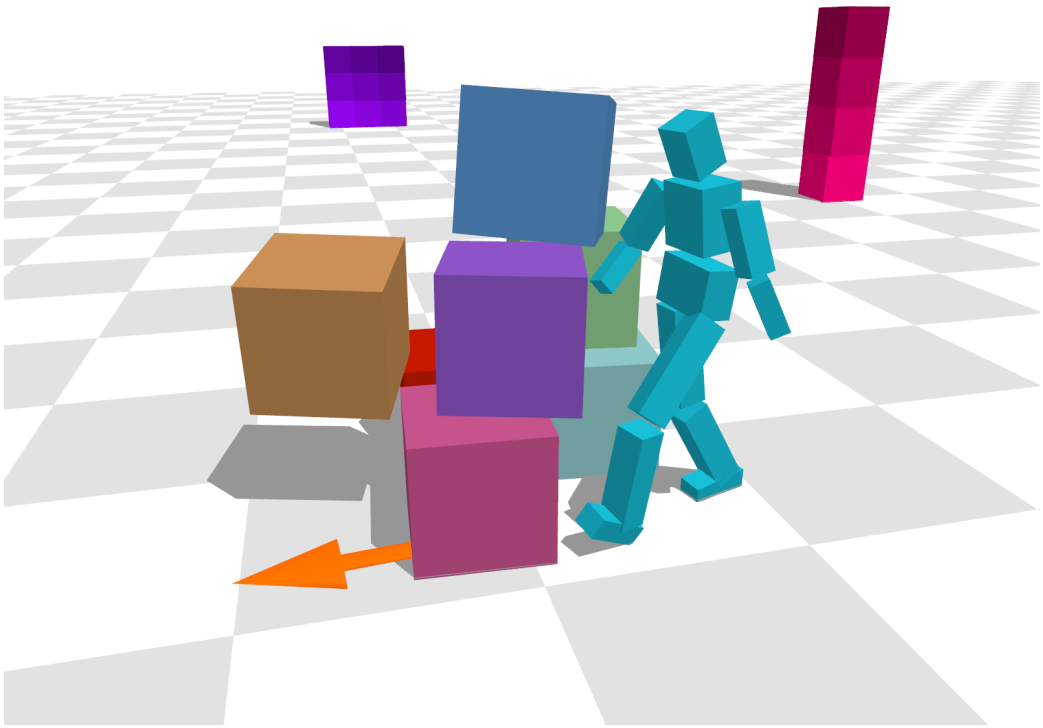


Figure 3.17: *A interactively-controlled biped navigating through stacked boxes.*

heel-strike, midstance, and toe-off. In our push experiments, a controller with its stance foot angle fixed with respect to the ground tends to withstand stronger pushes by 10N to 20N. A similar observation was reported by Wang et al. [72]. The loss of robustness is probably related to inaccurate modeling of the foot. Since our foot model is rigid, it usually have a small contact region on which ground reaction forces are applied. This makes the stance foot to wobble in the simulation. More realistic foot models might improve the robustness of our controller.

Ideally, reference motion data should be physically feasible for best tracking performance, though motion capture data are in general physically imprecise. Some of previous approaches [62, 49] preprocessed motion data to make them physically feasible via spacetime optimization. Spacetime optimization of three-dimensional, full-body motion data is notorious for its challenging nature of numerical instability and heavy computational burden. We did not employ such optimization in our experiments because our feedback rules worked effectively with our test data. However, we suspect that optimized reference data would allow our controller to be more robust and to follow the reference data more precisely. The optimization of reference data would be particularly important if motion data need to be warped, blended, and retargeted.

Chapter 4

Locomotion Control for Many-Muscle Humanoids

We present a biped locomotion controller for humanoid models actuated by more than a hundred Hill-type muscles. The key component of the controller is our novel scalable algorithm that can cope with step-based biped locomotion balancing and the coordination of many nonlinear Hill-type muscles simultaneously. Minimum effort muscle activations are calculated based on muscle contraction dynamics and online quadratic programming. Our controller can faithfully reproduce a variety of realistic biped gaits (e.g., normal walk, quick steps, and fast run) and adapt the gaits to varying conditions (e.g., muscle weakness, tightness, joint dislocation, and external pushes) and goals (e.g., pain reduction and efficiency maximization). We demonstrate the robustness and versatility of our controller with examples that can only be achieved using highly-detailed musculoskeletal models with many muscles.

4.1 Overview

Reproducing realistic human locomotion in physically based simulation has been a long-standing goal of computer graphics research. Many biped controllers in computer graphics assumed a linked structure of rigid bodies connected by idealized joints that can generate arbitrarily large torques along any directions immediately whenever needed. Such simplified body and actuation models made balance control and trajectory tracking plausible for full-body locomotion and even for acrobatic full-body actions. Recently, there have been continuous efforts to simulate and control muscle-actuated mechanisms to reproduce more realistic human locomotion.

The human body has over 700 skeletal muscles and 200 of them are especially important for locomotion because they move large bones. Harmonious coordination of many muscles results in complex human movements. Designing a control law for a many-muscle actuated humanoid poses several challenges: underdetermined control systems, the complexity of muscle contraction dynamics, and integrated controller design. The coordination of many muscles is inherently underdetermined since there are more muscles than the number of body degrees of freedom to actuate. Multiple sets of muscle actuations can lead to the same set of joint torques, and thus the same resulting motion. Moreover, it is often unclear which motion is best suited to given tasks under various intentions or conditions. The best one should be determined in a given situation. Secondly, the contraction dynamics of each individual muscle is a highly nonlinear process, which is often modeled using a three-element Hill-type model. The torque generated by an individual muscle depends on many factors such

as the geometry of joints and bones, the level of muscle activation, muscle length, and the velocity of muscle contraction. It is often impossible to generate an exact torque immediately, which can only be achieved through a dynamic process. Thirdly, due to the complexity of muscle dynamics, it is difficult to simply replace idealized joints with muscle-based actuators in existing biped controllers. Instead, the controllers should be completely redesigned to integrate fully functioning muscle dynamics.

In this chapter, we present a new muscle-actuated biped controller that scales well to cope with highly detailed musculoskeletal models having more than a hundred musculotendons. In biomechanics, such highly detailed musculoskeletal models have often been used for static analysis of human movements, but have not been employed to design locomotion controllers. Our controller is equipped with a step-based balance mechanism to reproduce a variety of human gait patterns, ranging from low-energy normal walk to highly energetic quick steps, while being resilient to external perturbations.

Our controller has two major technical components, muscle optimization and trajectory optimization, which integrate muscle actuators seamlessly into an existing step-based feedback controller. Given a reference motion of arbitrary gait patterns, muscle optimization calculates the optimal coordination of muscle activations to track the reference motion while resolving actuation redundancy by minimizing efforts. Since muscle actuations are computed on a per-frame basis, the solution is optimal only instantaneously. Our trajectory optimization actively modulates the full cycle (half cycle if the gait is symmetric) of the reference motion so that muscle dynamics can be accounted for in a longer horizon. The reference motion is optimized at the pre-

processing phase and the user can choose to reproduce the reference motion as closely as possible or adapt the motion to meet new conditions and intentions at runtime simulation.

We demonstrate the robustness and applicability of our controller using three musculoskeletal models that walk and run in many different gaits. Each individual gait can be adapted to new body conditions (e.g., muscle weakness, tightness, dislocated joints, and external pushes) or new objectives (e.g., pain reduction and efficiency maximization). Our many-muscle controller can reproduce the subtle nuances of pathologic gait conditions to match real patient data.

The key advantage of our many-muscle controller with respect to previous work is its scalability; it can control highly detailed musculoskeletal models, reproduce arbitrary gait patterns, adapt to a wide range of body conditions and optimization objectives. The scalability allows the controller to capture, generate, and adapt detailed nuances of biped gait patterns.

4.2 Humanoid Models

Our humanoid models are based on public-domain musculoskeletal models in the OpenSim file format [18]. We use three humanoid models in our experiments having 25 to 39 degrees of freedom (DOFs) and 62 to 120 muscles. Two of them are without arms as shown in Figure 4.1 (see Appendix B.2 for details). The height and mass of all models are about 180 cm and 75 kg. All DOFs of the models are actuated

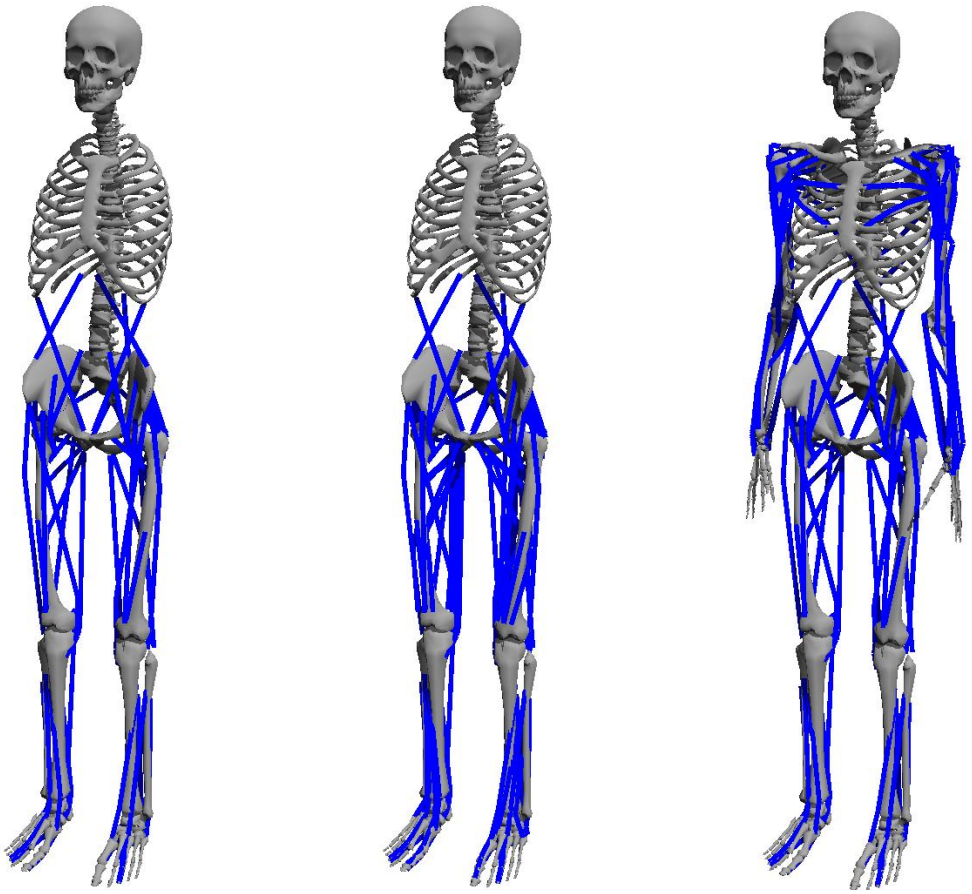


Figure 4.1: *Three musculoskeletal models. The* gait2562 *model on the left has 25 DOFs and 62 muscles, the gait2592 model at the middle has 25 DOFs and 92 muscles, and the fullbody model on the right has 39 DOFs and 120 muscles.*

by Hill-type musculotendon actuators [81], which is commonly used in biomechanics research (Figure 4.2).

A musculotendon actuator generates force depending on its activation level and other internal states. The generated force is transmitted to bones through the attachment points of the musculotendon actuator. The positions of the attachment points determine the lines of action of the transmitted forces. In the remainder of the thesis, we use the term "muscle" to indicate "musculotendon" or "musculotendon actuator" for convenience, except for being used as "muscle fiber".

4.2.1 Muscle Force Generation

The contraction dynamics equation of the Hill-type muscle model describes the behavior of the three components of a musculotendon actuator: a serial element, a contractile element, and a parallel element (Figure 4.2), which represents the tendon, the muscle fiber and the elastic material parallel to the muscle fiber, respectively. The contractile element generates active force f_{ce} based on the activation input, while the tendon force f_{t} and parallel element force f_{pe} are generated passively. According to Zajac [81], the relationship between these forces is described as:

$$f_{\mathrm{mt}} = f_{\mathrm{t}} = f_{\mathrm{m}} \cdot \cos(\alpha) = (f_{\mathrm{ce}} + f_{\mathrm{pe}})\cos(\alpha), \tag{4.1}$$

where f_{mt} is the musculotendon actuator force, f_{m} is the force acting on the muscle fiber, and α is the pennation angle which is the angle between the tendon and the muscle fiber. We take the pennation angle into account to model the musculotendon actuator of a human accurately.

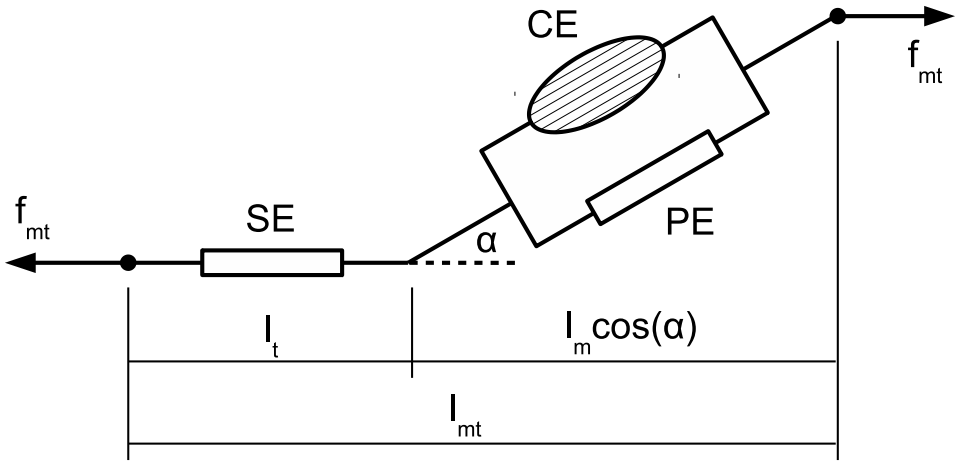


Figure 4.2: *The Hill-type muscle model is composed of a serial element (SE), a contractile element (CE), and a parallel element (PE).*

The contraction dynamics equation transforms the input activation level to the musculotendon force. We can compute the forces of each component in the Hill-type muscle model as follows:

$$f_{\mathrm{t}} = g_{\mathrm{t}}(l_{\mathrm{t}}), \tag{4.2}$$

$$f_{\mathrm{ce}} = a \cdot g_{\mathrm{al}}(l_{\mathrm{m}}) \cdot g_{\mathrm{v}}(\dot{l}_{\mathrm{m}}), \tag{4.3}$$

$$f_{\mathrm{pe}} = g_{\mathrm{pl}}(l_{\mathrm{m}}), \tag{4.4}$$

where g_{t}, g_{al}, g_{pl}, and g_{v} are the force-length relationship of the tendon, the active force-length relationship of the muscle fiber, the passive force-length relationship of the muscle fiber and the velocity-force relationship of the muscle fiber, respectively. Here l_{t}, l_{m}, $\dot{l}_{\mathrm{m}}$, and a are the length of the tendon, the length and the velocity of the muscle fiber and the activation level, respectively. Note that $l_{\mathrm{t}} = l_{\mathrm{mt}} - l_{\mathrm{m}}\cos(\alpha)$, where l_{mt} is the length of the musculotendon actuator which is determined based on

the current pose of the humanoid model. We use a modified version of the contraction dynamics equation proposed by Thelen [67] (see Appendix C.1 for more details about how to calculate g_{t}, g_{al}, g_{pl}, and g_{v}).

By rearranging Equation (4.1), (4.2), (4.3), and (4.4), the muscle fiber force f_{m} can be obtained:

$$f_{\mathrm{m}} = a \cdot g_{\mathrm{al}}(l_{\mathrm{m}}) \cdot g_{\mathrm{v}}(\dot{l}_{\mathrm{m}}) + g_{\mathrm{pl}}(l_{\mathrm{m}}). \tag{4.5}$$

The time-derivative of the lengths of the muscle fibers $\dot{l}_{\mathrm{m}}$ can be written as a function of a, l_{m}, and f_{ce}:

$$\dot{l}_{\mathrm{m}} = g_{\mathrm{v}}^{-1}(a, l_{\mathrm{m}}, f_{\mathrm{ce}}). \tag{4.6}$$

Our system updates the muscle fiber length l_{m} by integrating $\dot{l}_{\mathrm{m}}$ numerically. The initial values of a and l_{m} are provided as described in Appendix C.2.

Additionally, we employ a passive damping element parallel to the contractile element. A nonzero damping term is practically useful to make the simulation stable because Equation (4.6) has a singularity issue when a is close to zero [59]. Adding a damping term, the muscle dynamics equation becomes

$$f_{\mathrm{m}} = a \cdot g_{\mathrm{al}}(l_{\mathrm{m}}) \cdot g_{\mathrm{v}}(\dot{l}_{\mathrm{m}}) + g_{\mathrm{pl}}(l_{\mathrm{m}}) + b \cdot \dot{l}_{\mathrm{m}}, \tag{4.7}$$

$$\dot{l}_{\mathrm{m}} = g_{\mathrm{v}}^{-1}(a, l_{\mathrm{m}}, f_{\mathrm{ce}} - b \cdot \dot{l}_{\mathrm{m}}), \tag{4.8}$$

where b is a damping coefficient that is set to 0.05 for all muscles. (Appendix C.1 describes how we calculate the inverse of the velocity-force relationship equation g_{v}^{-1}).

4.2.2 Muscle Force Transfer

In the human body, each muscle has two ends attached to bones called the *insertion* and the *origin*. A muscle transfers its force to the bones through the insertion and the origin, of which the shapes are points, lines or areas. Our humanoid model simplifies insertions and origins as attachment points assuming all muscles are thin. Some muscles pass over the surfaces of other bones or muscles, and transfer some portion of their forces to them through muscle contact points (Figure 4.3). Contacts points between a muscle and a bone are implemented using *conditional contact points* that simulate the generation and elimination of contact points depending on the joint configurations [18].

Let b^i denote the i^{th} bone of the model, m^j be the j^{th} muscle, $\mathbf{p}_j^k$ be the position of the k^{th} attachment point or conditional contact point of m^j in the global coordinate, b_j^k be the bone on which $\mathbf{p}_j^k$ is attached or contacted. Here $\mathbf{p}_j^k$ is indexed such that $\mathbf{p}_j^{k-1}$ and $\mathbf{p}_j^{k+1}$ are adjacent points of $\mathbf{p}_j^k$ and are included in P_j which is the set of attachment points and contact-maintaining conditional contact points of m^j (Figure 4.3). We note that all elements of P_j are termed attachment points in the remainder of the paper for simplicity.

A muscle m^j exerts f_{mt}^j between $\mathbf{p}_j^1$ on b_j^1 and $\mathbf{p}_j^n$ on b_j^n through the path from $\mathbf{p}_j^1$ to $\mathbf{p}_j^n$. This is physically identical to the situation in which every pair of neighboring attachment points pulls each other with f_{mt}^j because the magnitude of the tensional force is equal at any point on m^j and the direction of the tensional force changes

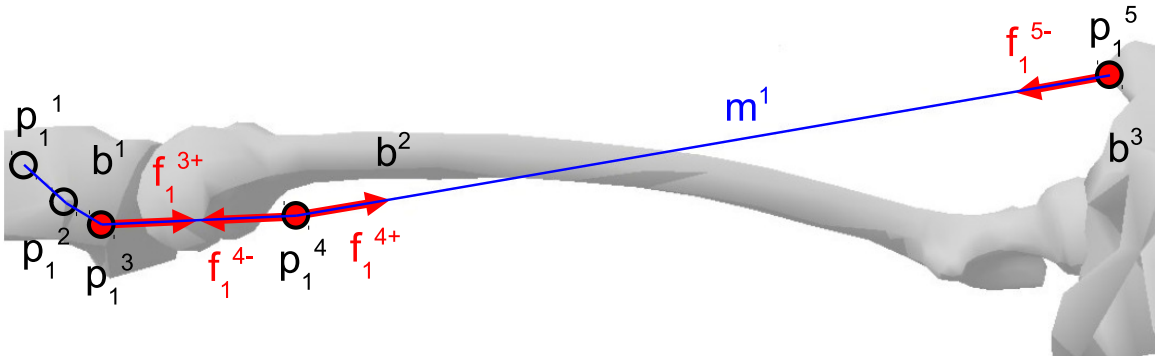
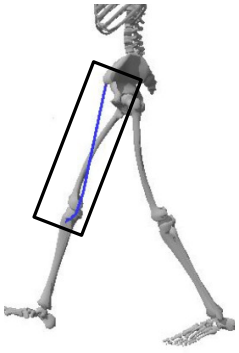


Figure 4.3: *Force transfer of a sartorius muscle which is a biarticular muscle (m^1) connecting the tibia (lower leg), femur (upper leg), and pelvis as shown on the right. Attachment points $\mathbf{p}_1^1$, $\mathbf{p}_1^2$, $\mathbf{p}_1^3$ are on the tibia bone (b^1), $\mathbf{p}_1^4$ is on the femur (b^2) and $\mathbf{p}_1^5$ is on the pelvis (b^3). The muscle force f_{mt}^1 pulls together a pair of points $(\mathbf{p}_1^3, \mathbf{p}_1^4)$, and another pair $(\mathbf{p}_1^4, \mathbf{p}_1^5)$. The same amount of force acts also between pairs $(\mathbf{p}_1^1, \mathbf{p}_1^2)$ and $(\mathbf{p}_1^2, \mathbf{p}_1^3)$ but cancels out because the points are attached to the same bone.*



through the path. The two force vectors $\mathbf{f}_j^{k-}$ and $\mathbf{f}_j^{k+}$ acting on b_j^k at $\mathbf{p}_j^k$ are:

$$\mathbf{f}_j^{k-} = f_{\text{mt}}^j \frac{\mathbf{p}_j^{k-1} - \mathbf{p}_j^k}{\|\mathbf{p}_j^{k-1} - \mathbf{p}_j^k\|}, \qquad \mathbf{f}_j^{k+} = f_{\text{mt}}^j \frac{\mathbf{p}_j^{k+1} - \mathbf{p}_j^k}{\|\mathbf{p}_j^{k+1} - \mathbf{p}_j^k\|}, \tag{4.9}$$

Here $\mathbf{f}_j^{1-}$ and $\mathbf{f}_j^{n+}$ are $\mathbf{0}$ because $\mathbf{p}_j^1$ and $\mathbf{p}_j^n$ are respectively the first and last attachment points of m^j.

If adjacent points $\mathbf{p}_j^k$ and $\mathbf{p}_j^{k+1}$ are attached to the same bone ($b_j^k = b_j^{k+1}$), $\mathbf{f}_j^{k+}$ and $\mathbf{f}_j^{(k+1)-}$ cancel each other out and do not have any effect on the entire body (see Figure 4.3 for an example). We exclude those points from P_j to improve simulation performance.

4.2.3 Equation of Motion

The equation of motion of the humanoid model is written as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_{\text{a}}^T \mathbf{f}_{\text{a}} + \mathbf{J}_{\text{c}}^T \mathbf{f}_{\text{c}}, \tag{4.10}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are the generalized position, velocity and acceleration of all DOFs, $\mathbf{M}$ is the inertia matrix, and $\mathbf{c}$ represents the Coriolis, centrifugal and gravitational force. The Jacobian matrices $\mathbf{J}_{\text{a}}$ and $\mathbf{J}_{\text{c}}$ map the generalized velocity $\dot{\mathbf{q}}$ to the global velocities at the attachment points and ground contact points, respectively. The vector $\mathbf{f}_{\text{a}}$ is the muscle forces at all attachment points which aggregates all $\mathbf{f}_j^{k\{-,+\}}$ of the model and $\mathbf{f}_{\text{c}}$ is the ground contact force, which can be formulated as

$$\mathbf{f}_{\text{c}} = \mathbf{V}_{\text{c}} \boldsymbol{\lambda}, \tag{4.11}$$

where $\mathbf{V}_{\mathrm{c}}$ is the linearized friction cone basis vectors, and $\boldsymbol{\lambda}$ is a coefficient vector. The muscle forces $\mathbf{f}_{\mathrm{a}}$ can be expressed as:

$$\mathbf{f}_{\mathrm{a}} = \mathbf{V}_{\mathrm{a}}\mathbf{C}\mathbf{f}_{\mathrm{mt}}, \tag{4.12}$$

where $\mathbf{V}_{\mathrm{a}}$ is unit direction vectors of all muscle forces $\mathbf{f}_j^{k\{-,+\}}$, $\mathbf{C}$ is the converting matrix that relates the indices of muscles to the indices of attachment points, and $\mathbf{f}_{\mathrm{mt}}$ is the aggregate vector of all scalar musculotendon forces f_{mt}^j.

Substituting the contraction dynamics equation (4.7) into Equation (4.12) transforms the input activation level to the musculotendon force:

$$\mathbf{f}_{\mathrm{a}} = \mathbf{V}_{\mathrm{a}}\mathbf{C}\mathbf{P}(\mathbf{A}\mathbf{a} + \mathbf{p}), \tag{4.13}$$

where $\mathbf{P}$ is a diagonal matrix containing the cosines of the pennation angles, and $\mathbf{a}$ is the muscle activations. $\mathbf{A}$ is a diagonal matrix containing the active force scaling parameters $g_{\mathrm{al}}(l_{\mathrm{m}}) \cdot g_{\mathrm{v}}(\dot{l}_{\mathrm{m}})$ for all muscles. $\mathbf{p}$ is a vector containing coefficients for the passive forces and passive damping elements $g_{\mathrm{pl}}(l_{\mathrm{m}}) + b \cdot \dot{l}_{\mathrm{m}}$ for all muscles. Given the assumption that our Hill-type muscle model is mass-less, the change of activation a immediately affects the muscle fiber lengths l_{m} and its time-derivatives $\dot{l}_{\mathrm{m}}$, which in turn determines the musculotendon force f_{mt} according to the nonlinear muscle dynamics equations (Equation (4.1), (4.2), and (4.7)). $\mathbf{A}$ and $\mathbf{p}$ linearize this highly nonlinear relationship between the activation a and force f_{mt} around the current values of l_{m} and $\dot{l}_{\mathrm{m}}$ such that they can be used in the quadratic program formulation of the muscle optimization. The final form of the equation of motion is:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_{\mathrm{a}}^T\mathbf{V}_{\mathrm{a}}\mathbf{C}\mathbf{P}(\mathbf{A}\mathbf{a} + \mathbf{p}) + \mathbf{J}_{\mathrm{c}}^T\mathbf{V}_{\mathrm{c}}\boldsymbol{\lambda}. \tag{4.14}$$

4.3 Muscle Optimization

The goal of the muscle optimization is to find the optimal coordination of muscle activation levels to control the musculoskeletal model. The muscle optimization is seamlessly integrated as a part of our controller and invoked in a per-frame basis. At every time step at runtime, our many-muscle controller adjusts the reference motion using a balance strategy presented by Kwon and Hodgins [36], which plans the balance-recovering reference motion instantaneously based on the estimated pendulum state, and then optimizes muscle actuations to track the adjusted reference motion.

4.3.1 Objectives

The optimization at runtime uses four objectives to minimize efforts, contact forces, deviation from the reference motion, and deviation from the end-effector trajectories. All terms are instantaneous and formulated as a quadratic form with respect to optimization variables $\ddot{\mathbf{q}}, \mathbf{a}$, and $\boldsymbol{\lambda}$.

Effort. Minimizing effort is important in solving the underdetermined muscle actuation problem in that the best muscle actuations among many possible coordinations can be found. The sum of squared activations measures instantaneous effort at any instance.

$$L_{\text{ef}} = \|\mathbf{a}\|^2. \tag{4.15}$$

Contact Force. Minimizing contact force reduces the impact from the ground and thus improves the stability of control.

$$L_{\text{cf}} = \|\boldsymbol{\lambda}\|^2. \tag{4.16}$$

Tracking. We compute the desired acceleration $\ddot{\mathbf{q}}_{\text{d}}$ to track the balance-recovering reference motion as follows:

$$\ddot{\mathbf{q}}_{\text{d}} = k_{\text{p}} f_{\text{diff}}(\mathbf{q}_{\text{r}}, \mathbf{q}) + k_{\text{d}}(\dot{\mathbf{q}}_{\text{r}} - \dot{\mathbf{q}}) + \ddot{\mathbf{q}}_{\text{r}}, \tag{4.17}$$

where $\mathbf{q}_{\text{r}}$, $\dot{\mathbf{q}}_{\text{r}}$ and $\ddot{\mathbf{q}}_{\text{r}}$ are the reference position, velocity and acceleration of all DOFs, and k_{p} and k_{d} are the tracking gains. The function f_{diff} computes the difference between two positional DOFs depending on the type of the corresponding joints. We use $k_{\text{d}} = 2\sqrt{k_{\text{p}}}$ for critical damping. The tracking objective minimizes the difference between $\ddot{\mathbf{q}}_{\text{d}}$ and $\ddot{\mathbf{q}}$:

$$L_{\text{tr}} = \|\ddot{\mathbf{q}}_{\text{d}} - \ddot{\mathbf{q}}\|^2. \tag{4.18}$$

End Effectors. Foot-step planning is essential for the biped to maintain its balance. The end-effector objectives reinforce the end-effectors to track their desired positional/angular trajectories more accurately. We apply the end-effector objectives for both feet and the torso.

$$L^i_{\text{ee}} = \|\ddot{\mathbf{y}}^i_{\text{d}} - \ddot{\mathbf{y}}^i\|^2, \tag{4.19}$$

where $\ddot{\mathbf{y}}^i_{\text{d}}$ and $\ddot{\mathbf{y}}^i$ are the desired and actual acceleration of the i^{th} body part. The desired acceleration is linearly proportional to the position and velocity differences between the desired and the current end-effector configurations.

4.3.2 Constraints

We use one equality constraint, the equation of motion in (4.14), to make the simulation and control physically plausible. Inequality conditions are used for the Coulomb ground contact model and the ranges of the muscle activations. The ground contacts are formulated as:

$$\boldsymbol{\lambda} \geq \mathbf{0}, \tag{4.20}$$

$$\mathbf{V}_{\mathrm{c}}^T \mathbf{J}_{\mathrm{c}} \ddot{\mathbf{q}} + \mathbf{V}_{\mathrm{c}}^T \dot{\mathbf{J}}_{\mathrm{c}} \dot{\mathbf{q}} + \dot{\mathbf{V}}_{\mathrm{c}}^T \mathbf{J}_{\mathrm{c}} \dot{\mathbf{q}} \geq \mathbf{0}, \tag{4.21}$$

which respectively are the friction cone condition and the non-penetration, non-slipping condition [16]. The muscle activations are in the range of 0 and 1.

$$\mathbf{0} \leq \mathbf{a} \leq \mathbf{1}. \tag{4.22}$$

The muscle activation level is zero when a muscle is fully relaxed exerting no active force, and one when a muscle is exerting its maximum active force.

4.3.3 Quadratic Programming Formulation

The muscle optimization step is formulated as a quadratic program using the objectives and linear constraints stated above:

$$\underset{\ddot{\mathbf{q}},\mathbf{a},\boldsymbol{\lambda}}{\text{minimize}} \qquad l_{\mathrm{ef}} L_{\mathrm{ef}} + l_{\mathrm{tr}} L_{\mathrm{tr}} + \sum_i l_{\mathrm{ee}}^i L_{\mathrm{ee}}^i + l_{\mathrm{cf}} L_{\mathrm{cf}}, \tag{4.23}$$

$$\text{subject to} \qquad \text{Equation (4.14), (4.20), (4.21), and (4.22)}, \tag{4.24}$$

where l_{ef}, l_{tr}, l_{ee}^i and l_{cf} are the weight constants for each objective and $i \in \{$left foot, right foot, torso$\}$.

4.4 Trajectory Optimization

The functionality of our trajectory optimization is twofold. It modulates the reference motion at the preprocessing phase such that our runtime controller can reproduce the original reference motion more accurately and robustly. It also allows us to change the reference motion more aggressively to adapt to new conditions and requirements. In the latter case, the original reference motion serves as an initial guess of the optimized solution.

In the offline trajectory optimization, we optimize only the feet trajectories to adapt since the feet trajectories are the most essential components of full body gaits. Even through only the feet trajectories are optimized, the impact of the change often affects the runtime controller to make a full body change of the simulated motion. Each of the feet trajectories is represented by offsets of the swing and the stance foot position from the corresponding reference foot position at three uniformly distributed keyframes. Assuming the symmetry of the gait, the dimension of the search space is 18 (6 offset points in 3D space). The offsets are applied to the balance-recovering reference motion that is tracked in the muscle optimization.

The trajectory optimization uses five objective terms. The first two terms are mandatory involving essential functionalities (trajectory tracking and balancing) of the controller, while the others are optional terms that we can choose to specify new requirements for motion adaptation. All terms are designed to account for longer horizon of gait patterns.

Pose Difference. Being able to faithfully reproduce any given reference motion is a fundamental objective. We penalize the deviation of the simulated motion from the original reference motion.

$$G_{\mathrm{pd}} = \sum_{1}^{N_{\mathrm{fall}}} f_{\mathrm{pdiff}}(\mathbf{q}_{\mathrm{r}}, \mathbf{q}), \tag{4.25}$$

where N_{fall} is the number of simulation time slots before falling down and f_{pdiff} computes the pose difference by point cloud matching.

Falling Down. The humanoid model should maintain its balance not to fall over. This objective penalizes earlier falling down:

$$G_{\mathrm{fd}} = \sum_{1}^{N_{\mathrm{fall}}} 0 + \sum_{N_{\mathrm{fall}}+1}^{N_{\mathrm{final}}} 1, \tag{4.26}$$

where N_{final} is the total number of simulation time slots. If the model does not fall down during the simulation, G_{fd} is zero.

Efficiency. The metabolic energy consumption has often been employed to measure the effort of locomotion in literature. We found that minimizing energy consumption generally leads to slow walk with a shorter stride. Instead, we use the efficiency term that measures energy consumption per unit moving distance, similarly to those used by Anderson and Pandy [5]:

$$G_{\mathrm{ec}} = -\frac{1}{D} \sum_{1}^{N_{\mathrm{fall}}} \dot{E}, \tag{4.27}$$

where $\dot{E}$ is the current rate of metabolic energy expenditure [74] and D is the total moving distance before falling down.

Contact Force. The contact force objective with longer horizon is motivated by the pain-avoidance behavior. Minimizing ground contact force can reduce pain on injured joints.

$$G_{\mathrm{cf}}^{i} = \sum_{1}^{N_{\mathrm{fall}}} \|\mathbf{f}_{\mathrm{c}}^{i}\|, \tag{4.28}$$

where $\mathbf{f}_{\mathrm{c}}^{i}$ is the resultant contact force of the i^{th} foot, $i \in \{\text{left foot, right foot}\}$.

Muscle Force. We introduce the muscle force objective to simulate another form of pain-avoidance behavior. Minimization of the force of a specific muscle can be interpreted as the pain-avoidance behavior due to strain or injury. The objective is:

$$G_{\mathrm{mf}}^{j} = \sum_{1}^{N_{\mathrm{fall}}} f_{\mathrm{mt}}^{j}, \tag{4.29}$$

where f_{mt}^{j} is the musculotendon force of the j^{th} muscle.

The total objective function is:

$$G = g_{\mathrm{pd}}G_{\mathrm{pd}} + g_{\mathrm{fd}}G_{\mathrm{fd}} + g_{\mathrm{ec}}G_{\mathrm{ec}} + \sum_{i} g_{\mathrm{cf}}^{i}G_{\mathrm{cf}}^{i} + \sum_{j} g_{\mathrm{mf}}^{j}G_{\mathrm{mf}}^{j}, \tag{4.30}$$

where g_{pd}, g_{fd}, g_{ec}, g_{cf}^{i} and g_{mf}^{j} are weight constants. To evaluate Equation (4.30), we run a simulation of 8 to 15 gait cycles using the runtime controller. The landscape of Equation (4.30) is highly-nonlinear and even discontinuous, and thus difficult to optimize. To minimize the objective function, we use the Covariance Matrix Adaption (CMA) algorithm [26], which is a derivative-free, stochastic optimization technique.

4.5 Results

We use our implementation based on the Lie group theory to build the equations of motion of the dynamic system (see [54]). The quadratic program of the muscle actuation optimization is solved at 120 Hz using Quadprog++. The simulation advances by integrating the results from the quadratic program. The simulation runs about eight to twelve times slower than real-time depending on the complexity of the humanoids. The muscle fiber lengths are updated at 840 Hz by using the contraction dynamics equation. The trajectory optimization for each example takes six to nine hours depending on the number of iterations (100 to 200) using 60 cores on a cluster of Intel Xeon E5-2680 machines. Muscles are rendered in blue when their activation level is zero, and in red when their activation level is one, with a linearly interpolated color scheme between them.

In the following examples, the weights g_{pd} and g_{fd} in the trajectory optimization objective G (in Equation (4.30)) are always assigned to positive values to optimize the controllers and g_{ec}, g_{cf}^{i} and g_{mf}^{j} are assigned to positive values when they are required, otherwise assigned to zero. For the pain avoidance, muscle weakness and tightness, and joint dislocation examples, the controller were optimized with *gait2592* model and *normal walk* as the reference motion.

Locomotion Skills. The first set of experiments was conducted to show that our scheme can reproduce captured reference motions. We optimized seven controllers named *normal walk, leaning walk, marching walk, slow run, in-place slow run, in-*

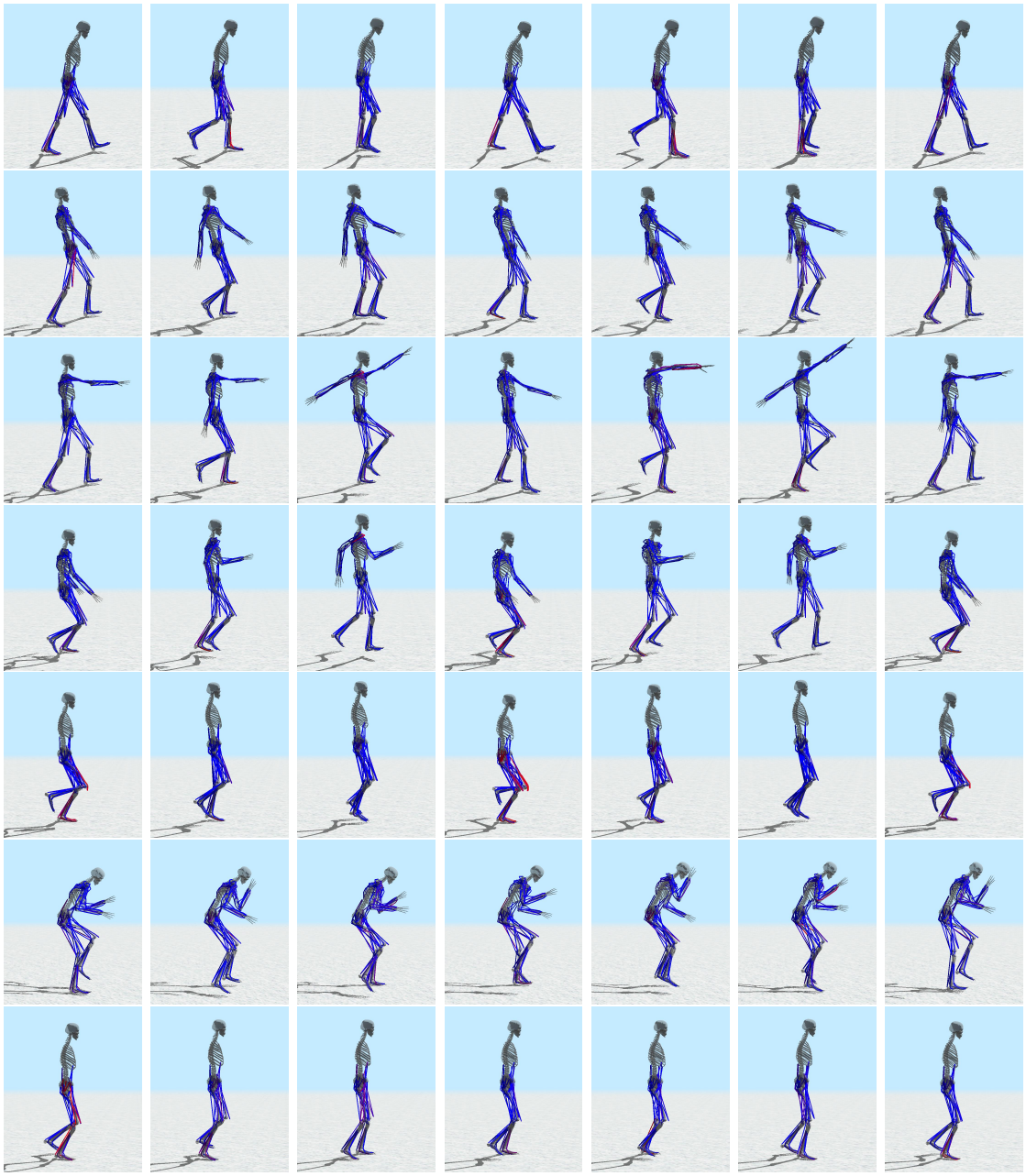


Figure 4.4: *Simulations of locomotion skills (top to bottom): normal walk with gait2592, leaning walk with fullbody, marching walk with fullbody, slow run with fullbody, in-place slow run with gait2592, in-place fast run with fullbody, and quick-step slow run with gait2562.*

place fast run, and *quick-step slow run* (Figure 4.6). Because the *normal walk* motion from the OpenSim distribution was without arm data, it is used only for *gait2592* and *gait2562*. Other reference motions were from public data [42, 61] and were optimized for all three humanoid models. All reference motions were retargeted to match the musculoskeletal models. When the reference motion was not long enough, we created a longer motion by stitching a few cycles together. For *slow run, in-place slow run* and *in-place fast run*, we scaled the isometric maximum forces of all muscles by two because the controller was not able to generate a stable cycle when the original maximum values from the OpenSim model file were used.

Pain Avoidance. People having pain in a limb tend to show asymmetric gait disturbance. Our optimization scheme based on the detailed musculoskeletal models allows us to predict the gait patterns of people with muscle or joint pain.

First, we simulated a unilateral painful ankle plantar flexor. People with such problems tend to reduce the use of the ankle plantar flexor. We gave the weights of the corresponding muscles in Equation (4.30) positive values. When the muscle forces are minimized, the optimal gait simulates an "antalgic" or a painful gait with lower activation of the affected ankle plantar flexor. The total force of the affected ankle plantar flexor is 44% less than that of the contralateral one.

Second, we simulated walking locomotion with "arthralgia" or painful joints of a unilateral limb. We gave a corresponding contact force weight in Equation (4.30) a positive value. Because the pain increases when the pressure inside each joint is increased, we can simply assume that the pain increases as the contact force is increased. The

optimized controller exhibits an "antalgic" gait pattern with a shorter stance duration of the affected limb. The total contact force and total stance duration of the affected limb are 55% and 70% less than those of the contralateral limb, respectively.

Muscle Weakness, Tightness. The next set of experiments was conducted while reducing the maximum muscle strength or increasing the tightness of specific muscles. The controllers were optimized with the efficiency objective in Equation (4.30).

We weakened the uni- or bilateral *gluteus medii* and *gluteus minimi* which act as abductors of the hip joints. For people who have weakness of bilateral *gluteus medii*, a "waddling" gait, a gait with an exaggerated lateral translation of the trunk, is observed. Our experiments reproduced the waddling gait with bilateral *gluteus medii* and *gluteus minimi* of which the maximum isometric force was scaled by 0.4. For people who have weakness of unilateral *gluteus medius*, a Trendelenburg gait, a gait with an exaggerated lateral translation of the trunk only in the direction of the weak muscles, is observed. Our experiments reproduced the Trendelenburg gait with those muscles of which the strength was scaled by 0.2. Our "many-muscle" control scheme allows us to simulate changes of muscles that control lateral movement, such as *gluteus medius*, which is not supported by the controller of Wang et. al. [74].

We also weakened the uni- or bilateral ankle plantar flexors which play an important role in walking by generating propulsion force when an ankle pushes off. Scaling a unilateral ankle plantar flexor by 0.1 resulted in a "limping" gait which does not much depend on the propulsion force of the affected ankle. Scaling the maximum isometric force of the bilateral ankle plantar flexors by 0.2 exhibited a slightly flexed

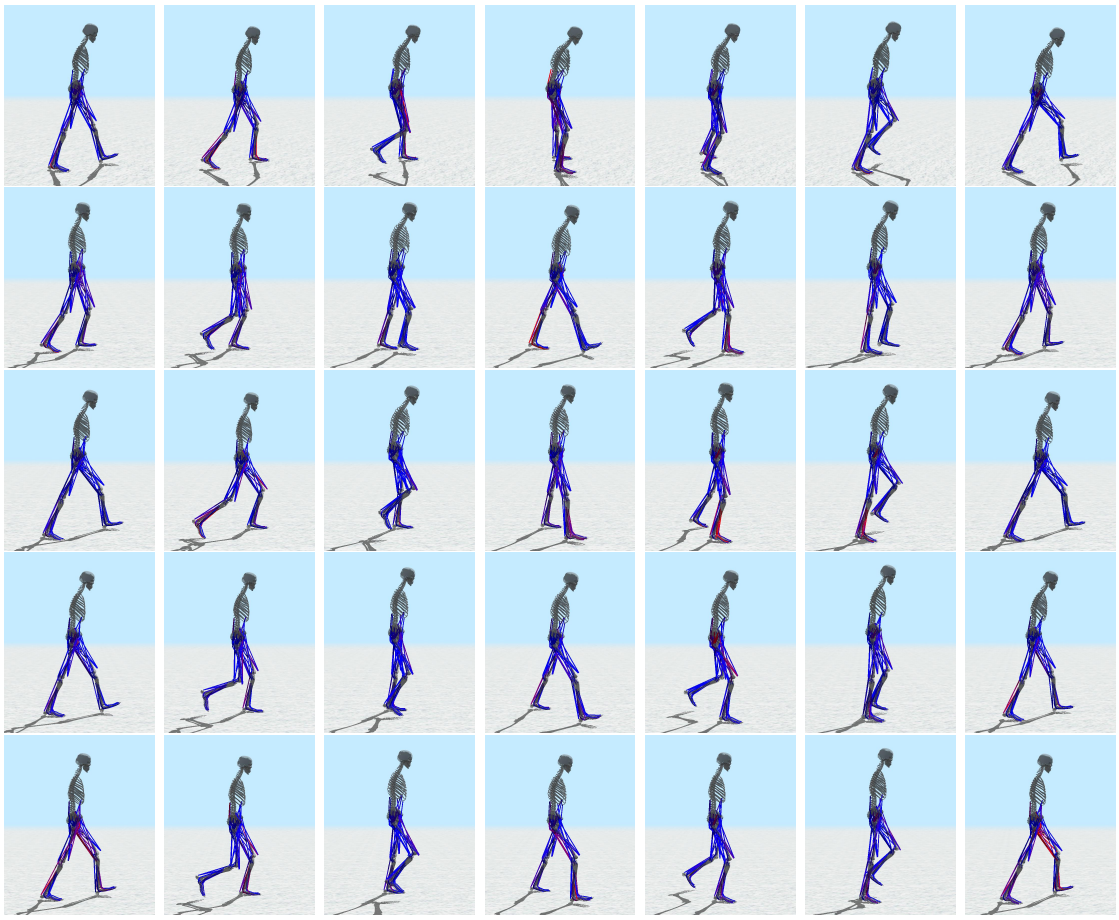


Figure 4.5: *Simulations with a left painful ankle plantar flexor, painful joints on a left limb, weakness of a left ankle plantar flexor, weakness of both of ankle plantar flexors, and tightness of hamstrings and psoai with weakness of ankle plantar flexors (top to bottom).*

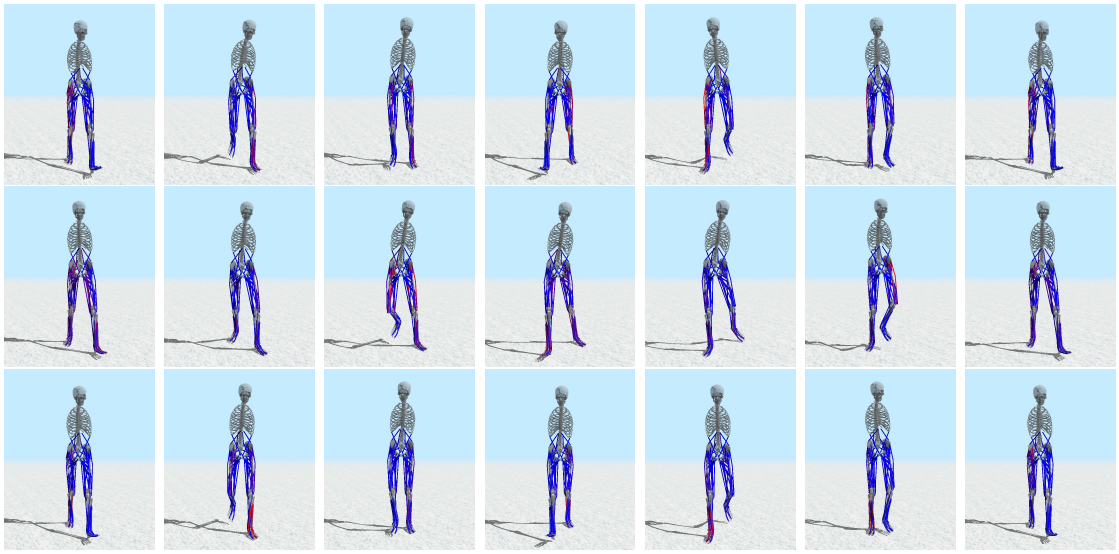


Figure 4.6: *Simulations with bi- and unilateral gluteus medii and gluteus minimi weakness exhibits waddling and Trendelenburg gait, respectively. (row1 and 2). Front view of normal walk simulation is given in the last row for comparison.*

knee gait.

Tightness of the hamstrings and *psoai* with weakness of the ankle plantar flexors is the most common reason for a "crouch" or flexed knee gait in people who suffer from cerebral palsy. We increased the muscle tightness by shortening the tendon slack length which is the rest length of the tendon. By scaling the tendon slack length of the bilateral hamstrings and *psoai* by 0.8 and the maximum isometric force of the bilateral ankle plantar flexors by 0.2, our controller generates a crouch gait with a more flexed knee joint.

Joint Dislocation. In the case of neglected developmental dysplasia of hip (DDH), the hip joint is dislocated in the superolateral direction. We simulated the gait with a

unilateral DDH by moving the hip joint 3 cm in the lateral direction. The optimized gait showed a Trendelenburg gait because the fulcrum of the hip joint moved in the lateral direction, the center of mass of the entire body should move in that direction also, as observed in people with neglected DDH. The controller was also optimized with the efficiency objective.

External Pushes. We tested the robustness of our controllers for external pushes. We applied force with a duration of 0.2 seconds at the connection part between the pelvis and the trunk. From the right, left, rear, and front directions, our controller performs upto 80N for *normal walk* and upto 160N for *slow run*.

Relative Intensity. The Metabolic Equivalent of Task (MET) is a cost measure for physical activities defined as the rate of metabolic energy consumption normalized by body weight [3]. Our controllers can be used to investigate the relative intensities of various exercises by measuring the MET, as it can simulate various locomotion skills with musculotendon actuators.

The simulation results of the MET with *gait2562* for locomotion skills are listed in Table 4.1. The simulated MET of the *normal walk* motion was 7.1. Unfortunately, this value was nearly 2.4 times as large as that measured from an actual human (about 3). This could simply have resulted due to the different energy metrics: the energy consumption of a real person is often measured by analyzing the intake and outtake of the person, while the metabolic energy metric attempts to predict this based on quantities relates to muscles, but these values could differ. Instead, it is also

Locomotion Skill	Average MET
normal walk	7.1
leaning walk	9.2
marching walk	8.4
slow run	10.6
in-place slow run	11.9
in-place fast run	28.4
quick-step slow run	11.3

Table 4.1: *Simulated MET values, which indicate the relative intensity of each locomotion skill.*

possible that the error resulted due to the simplification of the effort metric in the muscle actuation optimization. We would like to further investigate this problem in the future. In any case, the calculated energy consumption is still useful because one can obtain a rough idea about the relative intensity of an exercise.

4.6 Discussion

We presented a locomotion control system for detailed musculoskeletal humanoids. Our controller is not sensitive to model parameters such as the number of muscles, skeletal structures or locomotion styles. It does not rely on any algorithm specifically designed for a given musculoskeletal model, such as the one used by Wang et. al. [74]. Guided by motion capture data, our controller reproduces various styles of locomotion. The muscle activations are instantaneously optimized to choose the best actuations depending on the current state of the model among many redundant choices. The feet trajectories are optimized for a certain period of time to reproduce robust

and natural locomotion and adapt original locomotion styles to various intentions or conditions.

We intend to address important research problems in both graphics and biomechanics. In computer graphics, physics-based motion synthesis has been an important topic to gain physical realism of virtual humanoids. Especially, controller-based approaches produce simulated movements online, allowing humanoids to respond to unexpected disturbances. With hundreds of muscles whose parameters learned from real-human data, our controller can accurately simulate the actuation process of human locomotion while adapting to changes in muscle parameters. This allows us to physically synthesize natural motions and express various locomotion styles simply by changing the optimization criteria or reference motions. On the other hand, many biomechanists have developed detailed musculoskeletal models, and reproduced human movements using trajectory optimization. Our work reformulates these approaches based on highly detailed humanoids and an online control algorithm. The controller approach allows us to predict movements while continuously adapting to new conditions. This has significant practical meaning for medical applications such as understanding and treating muscle disorders. For example, our simulator can be a framework for the virtual surgical planning of gait correcting surgery in patients with cerebral palsy.

We proposed a novel formulation to resolve muscle redundancy, searching for the best actuations, accelerations and contact forces simultaneously. Muscle redundancy could also be resolved solely via static optimization, but we combine it with an online quadratic programming formulation which embeds muscle contraction dynamics to

generate dynamic controllers. To the best of our knowledge, our work builds upon state-of-the-art technology of human locomotion control, with one of the most detailed public domain musculoskeletal models ever used by controllers with the widest variety of gaits ever exhibited.

Our controller used muscle activation levels as control signals, and thus it lacks a delaying effect that comes from the activation dynamics and neural signal transmissions. Simulating such a delay would result in smoother and more realistic trajectories of activation levels.

Chapter 5

Conclusion

This thesis has proposed locomotion control systems for physically simulated humanoid models. We focus on locomotion because it is fundamental skill not only for humans and other animals but for artificial humanoids. Controllers are required for the humanoids to guarantee physical realism of simulated locomotion and interactions with environments while maintaining balance. It has many important implications and practical meanings in various fields, such as computer graphics, robotics, and biomechanics.

Our approach effectively tackles challenges on the locomotion control and the complexity of the human body by a data-driven control algorithm that exploits inherent robustness of human locomotion and reconstructing human musculoskeletal system with musculotendon actuation process. Our locomotion controllers satisfy challenging requirements such as human-likeness, richness, and resilience. The controllers are

capable of reproducing a wide range of natural locomotion motions, responding to internal or external changes, and evaluating metabolic quantities for musculoskeletal models.

In Chapter 3, we have proposed a locomotion controller that exploits human motion capture data and reproduce realistic human locomotion. Our controller takes reference motion data and generate walking simulation that resembles original reference motion. The key is to slightly modulate reference motion such that even a simple tracking controller reproduces them while maintaining balance. The modulation is inspired by pose adjustments of a human responding to changes of body states or ground contacts when walking. This process can be seen as biological concepts, imitation and response. Our humanoids imitates human motion data and response to environmental changes by balance feedback and synchronization rules.

Our approach has several advantages. Our controller does not require a non-linear optimization solver, derivative evaluation of equations of motion, optimal control methods and precomputation. This makes our controller easy-to-implement and computationally efficient. Because our controller doesn't need precomputation, it can take a reference trajectory generated on-the-fly. This means that a number of existing data-driven techniques can be easily combined with our controller, so the range of possible actions can be largely expanded.

Our algorithm is mimicking what we are doing everyday. Each individual exhibits a distinctive gait of walking, which serves as a reference trajectory. The distinctive gait may be learned from experience or may be innate as many biomechanists argued. We

modulate the reference trajectory to cope with changes in environments and external perturbations. We think that mimicking human behavior allows us to control the biped robustly.

In Chapter 4, we have presented a locomotion control system for many-muscle humanoids which are based on detailed human models suggested by biomechanics researchers. The humanoids are actuated by musculotendon actuators to simulate force-generating process of the human body more closely. Muscle actuation signals and movement patterns are optimized per-frame or in a longer horizon by several criteria such as minimum effort or minimum difference with reference motion. Our novel quadratic program formulation combines physics simulation with muscle contraction dynamics, and drives the humanoids to perform desired actions while resolving redundant muscle actuations.

Our controller is not sensitive to model parameters such as the number of muscles, skeletal structures and locomotion styles. It does not rely on any algorithm specifically designed for a given musculoskeletal model. Guided by motion capture data, our controller reproduces various styles of locomotion and adapts the motion to meet new body conditions or new objectives so it can reproduce the subtle nuance of pathologic gait conditions.

We have experimented mostly with locomotion data. Controlling and simulating a wider spectrum of human motions will be an exciting avenue for future research. We anticipate that we will see compelling simulated humanoids equipped with a variety of motor skills spanning from low-energy locomotion to highly-dynamic dancing and

athletic skills that build on data-driven control techniques such as those presented in this thesis. Controlling more repertoire of motor skills would extend our understanding of human moving mechanisms.

A more accurate musculoskeletal model would be another interesting future work for the musculoskeletal controllers. For example, it would be interesting to apply time-varying aspects of real human muscles such as fatigues and injuries. Including such time varying properties would be useful for both animation and medical applications. More accurate modeling of the muscle geometries, volumetric deformations, and of intersections between them would generate a much more appealing simulation of a human body.

Appendix A

Mathematical Definitions

A.1 Definitions of Transition Function

Transition $s(t) : [a, b] \rightarrow [0, 1]$ is a smooth scalar function that satisfies $s(a) = 0$, $s(b) = 1$, and $s'(a) = s'(b) = 0$. Specifically, we use a cubic polynomial:

$$\begin{aligned} s(t) &= -2\left(\frac{x-a}{b-a}\right)^3 + 3\left(\frac{x-a}{b-a}\right)^2, && \text{if } \ a \leq x \leq b \\ &= 0, && \text{if } \ x < a \\ &= 1, && \text{if } \ x > b. \end{aligned}$$

Backward transition function $r(t) : [a, b] \rightarrow [0, 1]$ varies smoothly from one to zero such that $r(a) = 1$, $r(b) = 0$, and $r'(a) = r'(b) = 0$.

Appendix B

Humanoid Models

B.1 Torque-Actuated Biped Models

In Chapter 3, we experimented with a basic torque-actuated biped model (*standard model*) and its variations (Figure B.1). All biped models share the same skeletal structure. They have 13 rigid body parts (head, torso, pelvis, upper arms, lower arms, thighs, shins, and feet) and 12 ball-and-socket joints inbetween the body parts. The total degrees of freedom of each model are 42.

Our *standard model* is used in the tracking examples and the interactive control example. The physical properties of *standard model* are given in Figure B.2.

The *standard model* is altered to have 50% shorter legs with 40% lighter weight, 50% shorter legs with its original weight, 50% longer legs, and a 2.3% and a 4.6% shorter left leg in the robustness examples (Figure B.1).

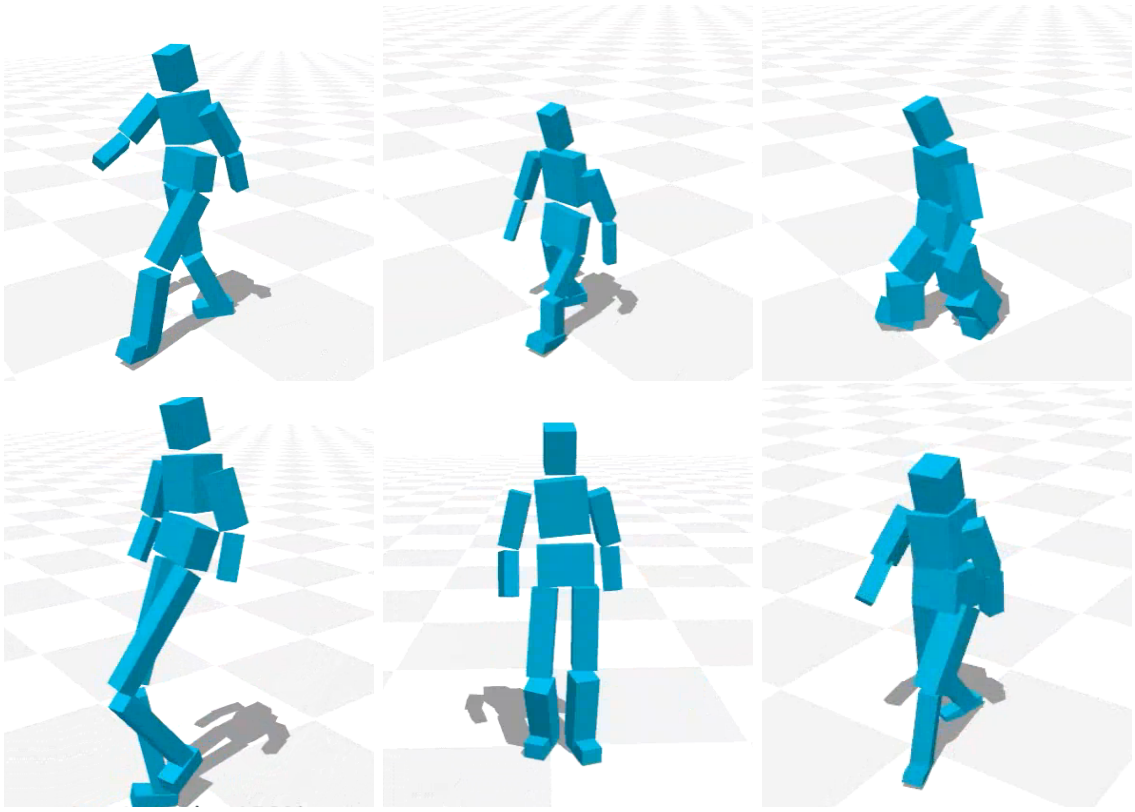


Figure B.1: *Our torque-actuated biped models (left to right, top to bottom): standard model, with 50% shorter legs and 40% lighter weight, 50% shorter legs, 50% longer legs, 4.6% shorter left leg, physical properties of the SIMBICON model.*

	Mass	Length	Inertia ($Kg\,m^2$)		
	(kg)	(m)	Lx	Ly	Lz
head	3	0.180	0.012	0.012	0.008
upper arm	2	0.238	0.011	0.011	0.003
lower arm	1	0.200	0.004	0.004	0.001
torso	8	0.230	0.052	0.068	0.049
hip	6	0.180	0.025	0.047	0.040
thigh	5	0.400	0.072	0.072	0.010
shin	5	0.380	0.065	0.065	0.011
foot	2	0.225	0.010	0.010	0.003

Figure B.2: *The physical properties of our standard model.*

In the push example, the physical properties of the *standard model* are changed to correspond with the biped model used in SIMBICON [80] as much as we could (Figure B.1).

For each variant of *standard model*, the shape and moment of inertia of each changed body part are computed from the mass information based on Figure B.2 under the assumption that the mass is distributed with uniform density ($1000kg/m^3$).

B.2 Many-Muscle Humanoid Models

We use three musculoskeletal models in the OpenSim file format (Figure B.3, Figure B.4). Each model description includes physical properties, relative locations of bodies, joints, and muscles attached to the bodies, and their physiological property such as a maximum isometric force and optimal fiber length. All models have polyg-

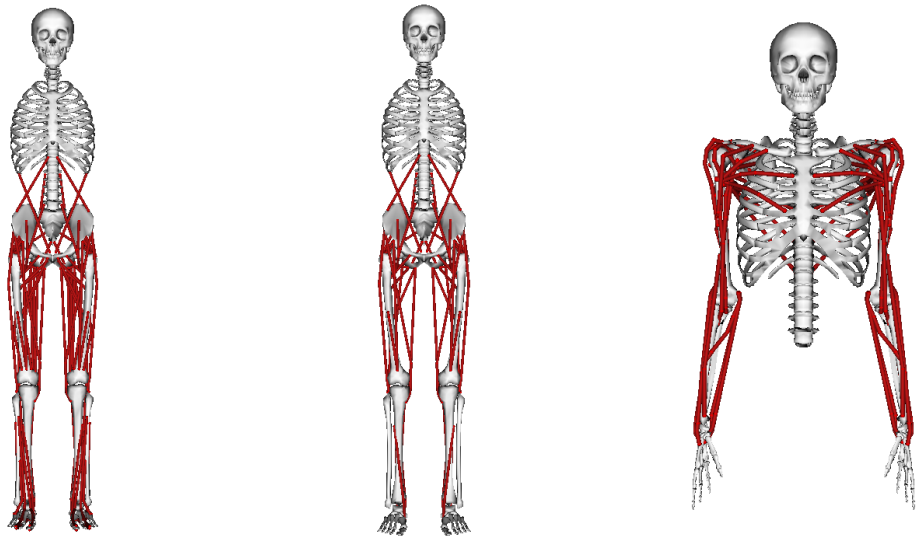


Figure B.3: *The original models which are base of our humanoid models (left to right): gait2392_simbody, gait2354_simbody, Dynamics Arms 2013*

onal mesh data for each bone. The vertices of each mesh are candidates for contact points during simulation.

Gait2592. This model is based on the *gait2392_simbody* model [19, 4] included in the OpenSim distribution. We make minor modifications to the OpenSim model for the easy of implementation. We merge each 1-DOF ankle joint and 1-DOF subtalar joint into one 3-DOF ankle joint because those two joints were very close to each other. The rotational center of the knee joints are assumed to be fixed in the local coordinates of the hip joints although the OpenSim model describes the knee-angle dependencies of the rotational center. The model has 25 DOFs and 92 muscles.

Gait2562. This model is based on the *gait2354_simbody* model, which is also included in the OpenSim distribution and has fewer muscles than *gait2392_simbody*. Along with

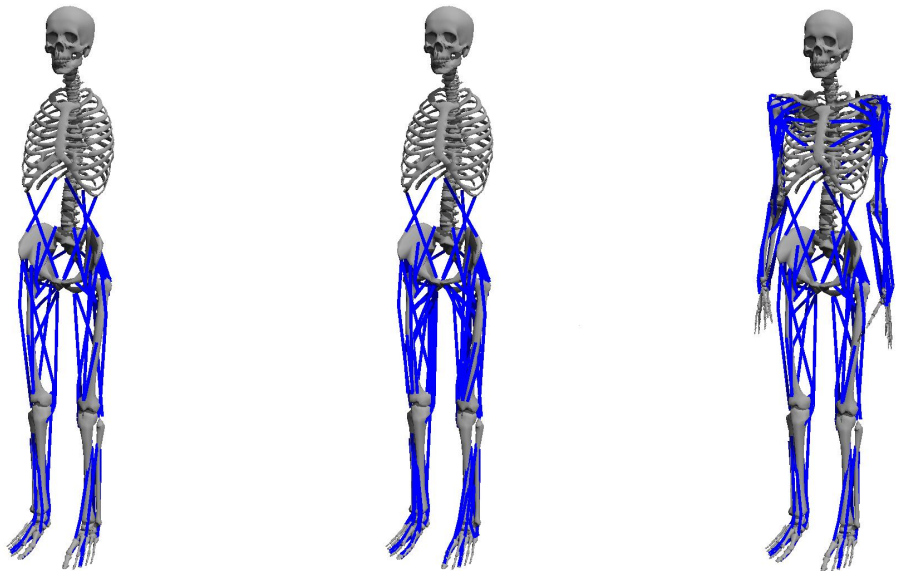


Figure B.4: *Our humanoid models (left to right): gait2562, gait2592, fullbody.*

the modifications on *gait2592*, two ankle evertors (peroneus longus on each leg) and six plantar/dorsi flexors (flexor hallucis longus, flexor digitorum longus and extensor digitorum longus on each leg) muscles are borrowed from *gait2392_simbody*, because *gait2354_simbody* has no ankle evertors and no muscles for the mtp joints. The model has 25 DOFs and 62 muscles.

***Fullbody* model**. We combine *gait2562* and *Dynamics Arms* [63] to make a fullbody model. Similarly to modification made to the ankle joints, the radioulnar joints were merged into the wrist joints. The model has 39 DOFs and 120 muscles.

Appendix C

Dynamics of Musculotendon Actuators

C.1 Contraction Dynamics

We use the contraction dynamics equations proposed by Thelen [67] with some modifications. Note that forces and lengths used in following equations are normalized by the maximum isometric force ($f_{\mathrm{m}}^{\mathrm{o}}$) and the optimal fiber length ($l_{\mathrm{m}}^{\mathrm{o}}$), respectively. The tilde symbols indicate normalized values.

Force-Length Relationship of Tendon. The tendon force-length relationship is defined as:

$$\tilde{g}_{\mathrm{t}}(\epsilon_{\mathrm{t}}) = \begin{cases} \dfrac{\tilde{f}_{\mathrm{t}}^{\mathrm{toe}}}{e^{k_{\mathrm{toe}}} - 1}(e^{k_{\mathrm{toe}} \cdot \epsilon_{\mathrm{t}} / \epsilon_{\mathrm{t}}^{\mathrm{toe}}} - 1), & \epsilon_{\mathrm{t}} \leq \epsilon_{\mathrm{t}}^{\mathrm{toe}} \\ \\ k_{\mathrm{lin}}(\epsilon_{\mathrm{t}} - \epsilon_{\mathrm{t}}^{\mathrm{toe}}) + \tilde{f}_{\mathrm{t}}^{\mathrm{toe}}, & \epsilon_{\mathrm{t}} > \epsilon_{\mathrm{t}}^{\mathrm{toe}} \end{cases} \tag{C.1}$$

where ϵ_{t} is the tendon strain ($\epsilon_{\mathrm{t}} = (l_{\mathrm{t}} - l_{\mathrm{t}}^{\mathrm{sl}})/l_{\mathrm{t}}^{\mathrm{sl}}$ where $l_{\mathrm{t}}^{\mathrm{sl}}$ is the tendon slack length), $\epsilon_{\mathrm{t}}^{\mathrm{toe}}$ and $\tilde{f}_{\mathrm{t}}^{\mathrm{toe}}$ define transition point of the curve from nonlinear to linear, k_{toe} and k_{lin} are shape factors. We use $f_{\mathrm{t}}^{\mathrm{toe}} = 0.33$, $k_{\mathrm{toe}} = 3.0$, $k_{\mathrm{lin}} = 1.712/\epsilon_{\mathrm{t}}^{\mathrm{o}}$, $\epsilon_{\mathrm{t}}^{\mathrm{toe}} = 0.609\epsilon_{\mathrm{t}}^{\mathrm{o}}$ where $\epsilon_{\mathrm{t}}^{\mathrm{o}}$ is the tendon strain due to maximum isometric force [67]. Note that $g_{\mathrm{t}}(l_{\mathrm{t}}) = f_{\mathrm{m}}^{\mathrm{o}} \cdot \tilde{g}_{\mathrm{t}}((l_{\mathrm{t}} - l_{\mathrm{t}}^{\mathrm{sl}})/l_{\mathrm{t}}^{\mathrm{sl}})$.

Passive Force-Length Relationship of Muscle. The original passive muscle force-length relationship used by Thelen [67] is defined as:

$$\tilde{g}_{\mathrm{pl}}^{\circ}(\tilde{l}_{\mathrm{m}}) = \frac{e^{k_{\mathrm{pe}}(\tilde{l}_{\mathrm{m}} - 1)/\epsilon_{\mathrm{m}}^{\mathrm{o}}} - 1}{e^{k_{\mathrm{pe}}} - 1}, \tag{C.2}$$

where $\tilde{l}_{\mathrm{m}}$ is the normalized muscle fiber length, $\epsilon_{\mathrm{m}}^{\mathrm{o}}$ is the passive muscle strain due to maximum isometric force and k_{pe} is a shape factor.

However, Equation (C.2) generates small negative force when $\tilde{l}_{\mathrm{m}}$ is smaller than 1, meaning that muscle fiber generates unrealistic "pushing" force.

We slightly modify the equation as follows:

$$\tilde{g}_{\mathrm{pl}}(\tilde{l}_{\mathrm{m}}) = \begin{cases} 0, & \tilde{l}_{\mathrm{m}} \leq 1 \\ \\ \tilde{g}_{\mathrm{pl}}^{\circ}(\tilde{l}_{\mathrm{m}}), & \tilde{l}_{\mathrm{m}} > 1 \end{cases} \tag{C.3}$$

Note that $g_{\mathrm{pl}}(l_{\mathrm{m}}) = f^{\circ}_{\mathrm{m}} \cdot \tilde{g}_{\mathrm{pl}}(l_{\mathrm{m}}/l^{\circ}_{\mathrm{m}})$.

Active Force-Length Relationship of Muscle. The active muscle force-length relationship is defined as:

$$\tilde{g}_{\mathrm{al}}(\tilde{l}_{\mathrm{m}}) = e^{-(\tilde{l}_{\mathrm{m}}-1)^2/\gamma}, \tag{C.4}$$

where γ is a shape factor.

Force-Velocity Relationship of Muscle. The original inverse function of muscle force-velocity relationship used by Thelen [67] is defined as:

$$\tilde{g}^{\circ -1}_{\mathrm{v}}(a, \tilde{l}_{\mathrm{m}}, \tilde{f}_{\mathrm{ce}}) = \dot{\tilde{l}}^{\max}_{\mathrm{m}}(0.25 + 0.75a)\frac{\tilde{f}_{\mathrm{ce}} - af_{\mathrm{l}}}{c}, \tag{C.5}$$

$$c = \begin{cases} af_{\mathrm{l}} + \tilde{f}_{\mathrm{ce}}/A_{\mathrm{f}}, & \tilde{f}_{\mathrm{ce}} \leq af_{\mathrm{l}} \\ \dfrac{(2 + 2/A_{\mathrm{f}})(af_{\mathrm{l}}\tilde{f}^{\mathrm{len}}_{\mathrm{m}} - \tilde{f}_{\mathrm{ce}})}{\tilde{f}^{\mathrm{len}}_{\mathrm{m}} - 1}, & \tilde{f}_{\mathrm{ce}} > af_{\mathrm{l}} \end{cases} \tag{C.6}$$

where f_{l} means $\tilde{g}_{\mathrm{al}}(\tilde{l}_{\mathrm{m}})$ and $\tilde{f}_{\mathrm{ce}}$ is the normalized contractile element force of muscle fiber, $\tilde{f}_{\mathrm{ce}} = \tilde{g}_{\mathrm{t}}(\epsilon_{\mathrm{t}})/\cos(\alpha) - \tilde{g}_{\mathrm{pl}}(\tilde{l}_{\mathrm{m}})$ by Equation (4.1), (4.2), (4.5) where $\dot{\tilde{l}}_{\mathrm{m}}$ is the normalized velocity of muscle fiber. Here $\dot{\tilde{l}}^{\max}_{\mathrm{m}}$, $\tilde{f}^{\mathrm{len}}_{\mathrm{m}}$ are normalized maximum contraction velocity of muscle fiber and normalized maximum muscle force, respectively, and A_{f} is a shape factor.

The force-velocity relationship g_{v} might be a function of not only $\dot{\tilde{l}}_{\mathrm{m}}$, but a or $\tilde{l}_{\mathrm{m}}$ especially when $\dot{\tilde{l}}_{\mathrm{m}}$ is near $\dot{\tilde{l}}^{\max}_{\mathrm{m}}$ [81]. By finding the inverse of Equation (C.5) analytically, we found g_{v} used by Thelen is a function of a and $\dot{\tilde{l}}_{\mathrm{m}}$. Because dependence

of the force-velocity relationship on a or $\tilde{l}_{\mathrm{m}}$ is not significant due to relatively short duration of near-zero muscle force state [81], we decided to use modified version of Equation (C.5) of which the inverse is a function of only $\tilde{\dot{l}}_{\mathrm{m}}$:

$$\tilde{g}_{\mathrm{v}}^{-1}(a, \tilde{l}_{\mathrm{m}}, \tilde{f}'_{\mathrm{ce}}) = \tilde{\dot{l}}_{\mathrm{m}} = \tilde{\dot{l}}_{\mathrm{m}}^{\max} \frac{\tilde{f}'_{\mathrm{ce}} - a f_{\mathrm{l}}}{c'}, \tag{C.7}$$

where c' can be calculated by Equation (C.6) while substituting f_{ce} to f'_{ce}. Equation (C.7) is a quadratic equation in the variable $\tilde{\dot{l}}_{\mathrm{m}}$ and easily solved with the quadratic formula with checking a possible range of $\tilde{\dot{l}}_{\mathrm{m}}$. Because we employ the parallel damping element, we use $\tilde{f}'_{\mathrm{ce}} = \tilde{f}_{\mathrm{ce}} - b \cdot \tilde{\dot{l}}_{\mathrm{m}}$ instead of $\tilde{f}_{\mathrm{ce}}$ as described in Equation (4.8).

We can find the force-velocity relationship by inverting Equation (C.7) analytically:

$$\tilde{g}_{\mathrm{v}}(\tilde{\dot{l}}_{\mathrm{m}}) = \begin{cases} \dfrac{\tilde{\dot{l}}_{\mathrm{m}} + \tilde{\dot{l}}_{\mathrm{m}}^{\max}}{\tilde{\dot{l}}_{\mathrm{m}}^{\max} - \tilde{\dot{l}}_{\mathrm{m}}/A_{\mathrm{f}}}, & \tilde{\dot{l}}_{\mathrm{m}} \leq 0 \\ \dfrac{f_{\mathrm{m}}^{\mathrm{len}} \tilde{\dot{l}}_{\mathrm{m}}(2 + 2/A_{\mathrm{f}}) + \tilde{\dot{l}}_{\mathrm{m}}^{\max}(\tilde{f}_{\mathrm{m}}^{\mathrm{len}} - 1)}{\tilde{\dot{l}}_{\mathrm{m}}(2 + 2/A_{\mathrm{f}}) + \tilde{\dot{l}}_{\mathrm{m}}^{\max}(\tilde{f}_{\mathrm{m}}^{\mathrm{len}} - 1)}, & \tilde{\dot{l}}_{\mathrm{m}} > 0 \end{cases} \tag{C.8}$$

Note that $f_{\mathrm{ce}} = a \cdot g_{\mathrm{al}}(l_{\mathrm{m}}) \cdot g_{\mathrm{v}}(\dot{l}_{\mathrm{m}}) = a \cdot f_{\mathrm{m}}^{\mathrm{o}} \cdot \tilde{g}_{\mathrm{al}}(\tilde{l}_{\mathrm{m}}) \cdot \tilde{g}_{\mathrm{v}}(\tilde{\dot{l}}_{\mathrm{m}})$.

The muscle property $f_{\mathrm{m}}^{\mathrm{o}}$, $l_{\mathrm{m}}^{\mathrm{o}}$, $l_{\mathrm{t}}^{\mathrm{sl}}$, α^{opt} are specified for each muscle in the humanoid model description. Our humanoids models use $\epsilon_{\mathrm{t}}^{\mathrm{o}} = 0.033$, $\epsilon_{\mathrm{m}}^{\mathrm{o}} = 0.6$, $k_{\mathrm{pe}} = 4.0$, $\gamma = 0.5$, $\tilde{\dot{l}}_{\mathrm{m}}^{\max} = 10$, $\tilde{f}_{\mathrm{m}}^{\mathrm{len}} = 1.8$, $A_{\mathrm{f}} = 0.3$ for all muscles.

C.2 Initial Muscle States

The initial value of a and l_{m} at the start of simulation are not clear because they are invisible internal muscle states. We starts the simulation with fully relaxed muscles, meaning that initial a is zero. Then l_{m} is computed to generate an isometric muscle force which is a force when $\dot{l}_{\mathrm{m}}$ is zero. We compute it by solving Equation (C.7) for $\tilde{l}_m$ given $a = 0$ and $\tilde{\dot{l}}_{\mathrm{m}} = 0$. Using Equation (4.1), (4.2), (4.7) instead of Equation (C.7) gives the same results.

Glossary for Medical Terms

A

abductor

A muscle that makes a motion that pulls a limb away from the midline of the body [75]. 76

ankle plantar flexor

A muscle that makes a movement which decreases the angle between the sole of the foot and the back of the leg. For example, when depressing an car pedal or standing on the tiptoes [75]. vii, 75–78

C

cerebral palsy

A group of permanent disorders of the development of movement and posture, causing activity limitation, that are attributed to non-progressive disturbances that occurred in the developing fetal or infant brain. Its motor disorders are often accompanied by disturbances of sensation, perception, cognition, communi-

cation, and behaviour, by epilepsy, and by secondary musculoskeletal problems [57]. 8, 78, 81

D

distal hamstring lengthening (DHL)

One of the common surgical operations for cerebral palsy patients. This operation lengthens distal parts of hamstring muscles and has been reported to decrease knee flexion and improve knee movements. Now, it is one of the standard surgical operations in single-event multilevel surgery for cerebral palsy patients [8] . 8

F

femoral derotation osteotomy (FDO)

A surgical operation for patients with rotational deformity on lower extremities that causes imperfect lever-arm functionality. It includes corrections of the femur with a metal plate at intertrochanteric or supracondylar part. [8] . 8

G

gluteus medius

One of the three gluteal muscles, which is a broad, thick, radiating muscle, situated on the outer surface of the pelvis. It is related with the abduction of the hip and the medial rotation of thigh. [76] . 76

R

rectus femoris transfer (RFT)

A surgical operation for cerebral palsy patients with spasticity of the hamstring and rectus femoris. It has been reported to be effective for patients with the stiff-knee gait. [8] . 8

S

single-event multilevel surgery

A standard surgical operation for people who suffered from spastic cerebral palsy and are able to walk. It is mainly composed by a combination of muscle tendon lengthening, tendon transfer, and derotation osteotomy. It maximize effectiveness of post-operative rehabilitation while decreasing total duration of hospital stay, needs of additional operations, and reoccurance of the deformity. [8] . 8

T

Tendo Achilles lengthening (TAL)

A surgical operation for tiptoe walking patients. It alleviate over-extension of plantar flexion - knee extension couple. [8] . 8

Bibliography

[1] Yeuhi Abe, Marco da Silva, and Jovan Popović. Multiobjective control with frictional contacts. In *Proc. Symp. on Computer Animation*, pages 249–258, 2007.

[2] Kirsten Acuna. The 15 highest-grossing movies of 2013. `http://www.businessinsider.com/highest-grossing-movies-2013-2013-12`, 2013.

[3] Barbara E Ainsworth, William L Haskell, Stephen D Herrmann, Nathanael Meckes, Jr Bassett, David R, Catrine Tudor-Locke, Jennifer L Greer, Jesse Vezina, Melicia C Whitt-Glover, and Arthur S Leon. 2011 compendium of physical activities: a second update of codes and MET values. *Medicine and science in sports and exercise*, 43(8):1575–1581, 2011.

[4] Frank C. Anderson and Marcus G. Pandy. A dynamic optimization solution for vertical jumping in three dimensions. *Computer Methods in Biomechanics and Biomedical Engineering*, 2(3):201–231, 1999.

[5] Frank C. Anderson and Marcus G. Pandy. Dynamic optimization of human walking. *Journal of Biomechanical Engineering*, 123(5):381–390, 2001.

[6] R Blickhan. The spring-mass model for running and hopping. *Journal of biomechanics*, 22(11-12):1217–1227, 1989.

[7] Karel Čapek. *R.U.R. (Rossum's Universal Robots)*. 1921.

[8] Chin Youb Chung, Kyu-Chang Wang, Moon Seok Park, Moon Suk Bang, Jehee Lee, and Hee Hwang ... *뇌성마비 [Introduction to Cerebral Palsy (in Korean)]*. Koonja, 2013.

[9] Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Generalized biped walking control. *ACM Trans. Graph. (SIGGRAPH 2010)*, 29(4), 2010.

[10] Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Robust task-based control policies for physics-based characters. *ACM Trans. Graph. (SIGGRAPH Asia 2009)*, 28(6), 2009.

[11] Stelian Coros, Philippe Beaudoin, Kang Kang Yin, and Michiel van de Panne. Synthesis of constrained walking skills. *ACM Trans. Graph. (SIGGRAPH Asia 2008)*, 27(6), 2008.

[12] Marco da Silva, Yeuhi Abe, and Jovan Popović. Interactive simulation of stylized human locomotion. *ACM Trans. Graph. (SIGGRAPH 2008)*, 27(3), 2008.

[13] Marco da Silva, Yeuhi Abe, and Jovan Popović. Simulation of human motion data using short-horizon model-predictive control. *Computer Graphics Forum (Eurographics 2008)*, 27(2), 2008.

[14] Marco da Silva, Frédo Durand, and Jovan Popović. Linear bellman combination

for control of character animation. *ACM Trans. Graph. (SIGGRAPH 2009)*, 28(3), 2009.

[15] DARPA Robotics Challenge. `http://www.theroboticschallenge.org`, 2014.

[16] Martin de Lasa, Igor Mordatch, and Aaron Hertzmann. Feature-based locomotion controllers. *ACM Trans. Graph. (SIGGRAPH 2010)*, 29(4), 2010.

[17] Vincent de Sapio, James Warren, Oussama Khatib, and Scott Delp. Simulating the task-level control of human motion: a methodology and framework for implementation. *The Visual Computer*, 21(5):289–302, 2005.

[18] S. L Delp, F. C Anderson, A. S Arnold, P. Loan, A. Habib, C. T John, E. Guendelman, and D. G Thelen. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. on Biomedical Engineering*, 54(11):1940–1950, 2007.

[19] S. L Delp, J. P Loan, M. G Hoy, F. E Zajac, E. L Topp, and J. M Rosen. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Trans. on Biomedical Engineering*, 37(8):757–767, 1990.

[20] Paul C. DiLorenzo, Victor B. Zordan, and Benjamin L. Sanders. Laughing out loud: control for modeling anatomically inspired laughter using audio. *ACM Trans. Graph. (SIGGRAPH Asia 2008)*, 27(5), 2008.

[21] Petros Faloutsos, Michiel van de Panne, and Demetri Terzopoulos. Composable

controllers for physics-based character animation. In *Proc. of SIGGRAPH 2001*, pages 251–260, 2001.

[22] Anthony C. Fang and Nancy S. Pollard. Efficient synthesis of physically valid human motion. *ACM Trans. Graph. (SIGGRAPH 2003)*, 22(3):417–426, 2003.

[23] Gartner, Inc. Gartner says worldwide video game market to total $93 billion in 2013. `http://www.gartner.com/newsroom/id/2614915`, 2013.

[24] Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Trans. Graph. (SIGGRAPH Asia 2013)*, 32(6), 2013.

[25] Michael Gleicher. Retargeting motion to new characters. In *Proc. of SIGGRAPH 98*, pages 33–42, 1998.

[26] Nikolaus Hansen. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*, volume 192 of *Studies in Fuzziness and Soft Computing*, pages 75–102. 2006.

[27] A. V. Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B - Biological Sciences*, 126(843):136–195, 1938.

[28] A. V Hill. *First and last experiments in muscle mechanics*. University Press, Cambridge, 1970.

[29] Jessica K. Hodgins and Nancy S. Pollard. Adapting simulated behaviors for new

characters. In *Proc. of SIGGRAPH 97*, pages 153–162, 1997.

[30] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. Animating human athletics. In *Proc. SIGGRAPH 95*, pages 71–78, 1995.

[31] Jinwook Kim. Virtual Physics: The realtime dynamics simulation library. `http://virtualphysics.imrc.kist.re.kr/`, 2009.

[32] Jung-Yup Kim, Ill-Woo Park, and Jun-Ho Oh. Walking control algorithm of biped humanoid robot on uneven and inclined floor. *J. Intelligent and Robotic Systems*, 48(4):457–484, 2007.

[33] Manmyung Kim, Kyung Lyul Hyun, Jongmin Kim, and Jehee Lee. Synchronized multi-character motion editing. *ACM Trans. Graph. (SIGGRAPH 2009)*, 28(3), 2009.

[34] Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L. Kunii. Creating and retargetting motion by the musculoskeletal human body model. *The Visual Computer*, 16(5):254–270, 2000.

[35] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Trans. Graph. (SIGGRAPH 2002)*, 21(3):473–482, 2002.

[36] Taesoo Kwon and Jessica K. Hodgins. Control systems for human running using an inverted pendulum model and a reference motion capture sequence. In *Proc. Symp. on Computer Animation*, pages 129–138, 2010.

[37] Jehee Lee. Representing rotations and orientations in geometric computing.

IEEE Computer Graphics and Applications, 28(2):75–83, 2008.

[38] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph. (SIGGRAPH 2002)*, 21(3):491–500, 2002.

[39] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proc. of SIGGRAPH 99*, pages 39–48, August 1999.

[40] Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph.*, 28(4), 2009.

[41] Sung-Hee Lee and Demetri Terzopoulos. Heads up!: biomechanical modeling and neuromuscular control of the neck. *ACM Trans. Graph. (SIGGRAPH 2006)*, 25(3), 2006.

[42] Yoonsang Lee, Sungeun Kim, and Jehee Lee. Data-driven biped control. *ACM Trans. Graph. (SIGGRAPH 2010)*, 29(4), 2010.

[43] C. Karen Liu and Zoran Popović. Synthesis of complex dynamic character motion from simple animations. *ACM Trans. Graph. (SIGGRAPH 2002)*, 21(3):408–416, 2002.

[44] Libin Liu, KangKang Yin, Michiel van de Panne, and Baining Guo. Terrain runner: control, parameterization, composition, and planning for highly dynamic motions. *ACM Trans. Graph. (SIGGRAPH Asia 2012)*, 31(6), 2012.

[45] Adriano Macchietto, Victor Zordan, and Christian R. Shelton. Momentum control for balance. *ACM Trans. Graph. (SIGGRAPH 2009)*, 28(3), 2009.

[46] Igor Mordatch, Martin de Lasa, and Aaron Hertzmann. Robust physics-based locomotion using low-dimensional planning. *ACM Trans. Graph. (SIGGRAPH 2010)*, 29(4), 2010.

[47] Igor Mordatch, Jack M. Wang, Emanuel Todorov, and Vladlen Koltun. Animating human lower limbs using contact-invariant optimization. *ACM Trans. Graph. (SIGGRAPH Asia 2013)*, 32(6), 2013.

[48] Jun Morimoto, Garth Zeglin, and Chris Atkeson. Minimax differential dynamic programming: Application to a biped walking robot. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[49] Uldarico Muico, Yongjoon Lee, Jovan Popović, and Zoran Popović. Contact-aware nonlinear control of dynamic characters. *ACM Trans. Graph. (SIGGRAPH 2009)*, 28(3), 2009.

[50] Eadweard Muybridge. *Animal Locomotion. An Electro-Photographic Investigation of Consecutive Phases of Animal Movements.* University of Pennsylvania, 1887.

[51] Jun Nakanishi, Jun Morimoto, Gen Endo, Gordon Cheng, Stefan Schaal, and Mitsuo Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2-3):79–91, 2004.

[52] Masao Nakanishi, Taishin Nomura, and Shunsuke Sato. Stumbling with optimal

phase reset during gait can prevent a humanoid from falling. *Biol. Cybern.*, 95(5):503–515, 2006.

[53] S. Nakaoka, A. Nakazawa, and K. Yokoi. Generating whole body motions for a biped humanoid robot from captured human dances. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3905–3910, 2003.

[54] F. C. Park, J. E. Bobrow, and S. R. Ploen. A lie group formulation of robot dynamics. *The International Journal of Robotics Research*, 14(6):609–618, 1995.

[55] Marc H. Raibert and Jessica K. Hodgins. Animation of dynamic legged locomotion. In *Proc. SIGGRAPH 91*, pages 349–358, 1991.

[56] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics & Applications*, 18(5):32–40, 1998.

[57] Peter Rosenbaum, Nigel Paneth, Alan Leviton, Murray Goldstein, Martin Bax, Diane Damiano, Bernard Dan, Bo Jacobsson, et al. A report: the definition and classification of cerebral palsy april 2006. *Dev Med Child Neurol Suppl*, 109(8-14), 2007.

[58] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph. (SIGGRAPH 2004)*, 23(3):514–521, 2004.

[59] Lisa Margaret Schutte. *Using musculoskeletal models to explore strategies for*

improving performance in electrical stimulation-induced leg cycle ergometry. PhD thesis, Stanford University, 1993.

[60] Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. (SIGGRAPH 2005)*, 24(3), 2005.

[61] SNU Motion Database. `http://mrl.snu.ac.kr/~mdb/`, 2013.

[62] Kwang Won Sok, Manmyung Kim, and Jehee Lee. Simulating biped behaviors from human motion data. *ACM Trans. Graph. (SIGGRAPH 2007)*, 26(3), 2007.

[63] Katherine Steele and Samuel Hamner. Dynamic arms 2013. `https://simtk.org/home/arm2013`, 2013.

[64] Shinjiro Sueda, Andrew Kaufman, and Dinesh K. Pai. Musculotendon simulation for hand animation. *ACM Trans. Graph. (SIGGRAPH 2008)*, 27(3), 2008.

[65] Russ Tedrake, Teresa Weirui Zhang, and H. Sebastian Seung. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2849–2854, 2004.

[66] Joseph Teran, Eftychios Sifakis, Silvia S. Blemker, Victor Ng-Thow-Hing, Cynthia Lau, and Ronald Fedkiw. Creating and simulating skeletal muscle from the visible human data set. *IEEE Trans. Vis. Comput. Graph*, 11(3):317—328, 2005.

[67] Darryl G Thelen. Adjustment of muscle mechanics model parameters to simulate

dynamic contractions in older adults. *Journal of Biomechanical Engineering*, 125(1):70–77, 2003.

[68] 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.

[69] Yao-Yang Tsai, Wen-Chieh Lin, Kuangyou B. Cheng, Jehee Lee, and Tong-Yee Lee. Real-time physics-based 3D biped character animation using an inverted pendulum model. *IEEE Trans. Vis. Comput. Graph*, 16(2):325—337, 2010.

[70] Winnie Tsang, Karan Singh, and Fiu Eugene. Helping hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *Proc. Symp. on Computer Animation*, pages 319–328, 2005.

[71] Kevin Wampler and Zoran Popović. Optimal gait and form from animal locomotion. *ACM Trans. Graph. (SIGGRAPH 2009)*, 28(3), 2009.

[72] J. M Wang, D. J Fleet, and A. Hertzmann. Optimizing walking controllers. *ACM Trans. Graph. (SIGGRAPH Asia 2009)*, 28(5), 2009.

[73] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Optimizing walking controllers for uncertain inputs and environments. *ACM Trans. Graph. (SIGGRAPH 2010)*, 29(4), 2010.

[74] Jack M. Wang, Samuel R. Hamner, Scott L. Delp, and Vladlen Koltun. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Trans. Graph. (SIGGRAPH 2012)*, 31(4), 2012.

[75] Wikipedia. Anatomical terms of motion — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Anatomical_terms_of_motion`, 2014. [Online; accessed 25-April-2014].

[76] Wikipedia. Gluteus medius muscle — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Gluteus_medius_muscle`, 2014. [Online; accessed 25-April-2014].

[77] Jia-chi Wu and Zoran Popović. Terrain-adaptive bipedal locomotion control. *ACM Trans. Graph. (SIGGRAPH 2010)*, 29(4), 2010.

[78] Yuting Ye and C. Karen Liu. Optimal feedback control for character animation using an abstract model. *ACM Trans. Graph. (SIGGRAPH 2010)*, 29(4), 2010.

[79] KangKang Yin, Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Continuation methods for adapting simulated skills. *ACM Trans. Graph. (SIGGRAPH 2008)*, 27(3), 2008.

[80] KangKang Yin, Kevin Loken, and Michiel van de Panne. SIMBICON: simple biped locomotion control. *ACM Trans. Graph. (SIGGRAPH 2007)*, 26(3), 2007.

[81] F E Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering*, 17(4):359–411, 1989.

[82] Victor B. Zordan and Jessica K. Hodgins. Motion capture-driven simulations that hit and react. In *Proc. Symp. on Computer Animation*, pages 89–96, 2002.

초 록

휴머노이드를 제어하여 사람의 자연스러운 이동 동작을 만들어내는 것은 컴퓨터 그래픽스 및 로봇공학 분야에서 중요한 문제로 생각되어 왔다. 하지만, 이는 사람의 이동에서 구동기가 부족한 (underactuated) 특성과 사람의 몸의 복잡한 구조를 모방하고 시뮬레이션해야 한다는 점 때문에 매우 어려운 문제로 알려져왔다. 본 학위논문은 물리 시뮬레이션 기반 휴머노이드가 외부의 변화에 안정적으로 대응하고 실제 사람처럼 자연스럽고 다양한 이동 동작을 만들어내도록 하는 제어 방법을 제안한다. 우리는 실제 사람으로부터 얻을 수 있는 관찰 가능하고 측정 가능한 데이터를 최대한으로 활용하여 문제의 어려움을 극복했다. 우리의 접근 방법은 모션 캡처 시스템으로부터 획득한 사람의 모션 데이터를 활용하며, 실제 사람의 측정 가능한 물리적, 생리학적 특성을 복원하여 사용하는 것이다.

우리는 토크로 구동되는 이족 보행 모델이 다양한 스타일로 걸을 수 있도록 제어하는 데이터 기반 알고리즘을 제안한다. 우리의 알고리즘은 모션 캡처 데이터에 내재된 이동 동작 자체의 강건성을 활용하여 실제 사람과 같은 사실적인 이동 제어를 구현한다. 구체적으로는, 참조 모션 데이터를 재현하는 자연스러운 보행 시뮬레이션을 위한 관절 토크를 계산하게 된다. 알고리즘에서 가장 핵심적인 아이디어는 간단한 추종 제어기만으로도 참조 모션을 재현할 수 있도록 참조 모션을 연속적으로 조절하는 것이다. 우리의 방법은 모션 블렌딩, 모션 와핑, 모션 그래프와 같은 기존

에 존재하는 데이터 기반 기법들을 이족 보행 제어에 활용할 수 있게 한다.

우리는 보다 사실적인 이동 동작을 생성하기 위해 사람의 몸을 세부적으로 모델링한, 근육에 의해 관절이 구동되는 인체 모델을 제어하는 이동 제어 시스템을 제안한다. 시뮬레이션에 사용되는 휴머노이드는 실제 사람의 몸에서 측정된 수치들에 기반하고 있으며 최대 120개의 근육을 가진다. 우리의 알고리즘은 최적의 근육 활성화 정도를 계산하여 시뮬레이션을 수행하며, 참조 모션을 충실히 재현하거나 혹은 새로운 상황에 맞게 모션을 적응시키기 위해 주어진 참조 모션을 수정하는 방식으로 동작한다. 우리의 확장가능한 알고리즘은 다양한 종류의 근골격 인체 모델을 최적의 근육 조합을 사용하며 균형을 유지하도록 제어할 수 있다.

우리는 다양한 스타일로 걷기 및 달리기, 모델의 변화 (근육의 약화, 경직, 관절의 탈구), 환경의 변화 (외력), 목적의 변화 (통증의 감소, 효율성의 최대화)에 대한 대응, 방향 전환, 회전, 인터랙티브하게 방향을 바꾸며 걷기 등과 같은 보다 난이도 높은 동작들로 이루어진 예제를 통해 우리의 접근 방법이 효율적임을 보였다.

주요어: 컴퓨터 애니메이션, 물리 기반 시뮬레이션, 이족 보행 제어, 휴머노이드 이동 제어, 근골격 휴머노이드 제어

학번: 2007-23045