Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation

Seungmoon Song[1]*, Łukasz Kidziński[2], Xue Bin Peng[3], Carmichael Ong[2], Jennifer Hicks[2], Sergey Levine[3], Christopher G. Atkeson[4], and Scott L. Delp[1,2]

[1] *Mechanical Engineering, Stanford University, Stanford, United States*

[2] *Bioengineering, Stanford University, Stanford, United States*

[3] *Electrical Engineering and Computer Science, University of California, Berkeley, United States*

[4] *Robotics Institute, Carnegie Mellon University, United States*

Abstract

Modeling human motor control and predicting how humans will move in novel environments is a grand scientific challenge. Despite advances in neuroscience techniques, it is still difficult to measure and interpret the activity of the millions of neurons involved in motor control. Thus, researchers in the fields of biomechanics and motor control have proposed and evaluated motor control models via neuromechanical simulations, which produce physically correct motions of a musculoskeletal model. Typically, researchers have developed control models that encode physiologically plausible motor control hypotheses and compared the resulting simulation behaviors to measurable human motion data. While such plausible control models were able to simulate and explain many basic locomotion behaviors (e.g. walking, running, and climbing stairs), modeling higher layer controls (e.g. processing environment cues, planning long-term motion strategies, and coordinating basic motor skills to navigate in dynamic and complex environments) remains a challenge. Recent advances in deep reinforcement learning lay a foundation for modeling these complex control processes and controlling a diverse repertoire of human movement; however, reinforcement learning has been rarely applied in neuromechanical simulation to model human control. In this paper, we review the current state of neuromechanical simulations, along with the fundamentals of reinforcement learning, as it applies to human locomotion. We also present a scientific competition and accompanying software platform, which we have organized to accelerate the use of reinforcement learning in neuromechanical simulations. This "Learn to Move" competition, which we have run annually since 2017 at the NeurIPS conference, has attracted over 1300 teams from around the world. Top teams adapted state-of-art deep reinforcement learning techniques to produce complex motions, such as quick turning and walk-to-stand transitions, that have not been demonstrated before in neuromechanical simulations without utilizing reference motion data. We close with a discussion of future opportunities at the intersection of human movement simulation and reinforcement learning and our plans to extend the Learn to Move competition to further facilitate interdisciplinary collaboration in modeling human motor control for biomechanics and rehabilitation research.

Keywords: neuromechanical simulation, deep reinforcement learning, motor control, locomotion, biomechanics, musculoskeletal modeling, academic competition

Introduction

Predictive neuromechanical simulations can produce motions without directly using experimental motion data. If the produced motions reliably match how humans move in novel situations, predictive simulations can be used to accelerate research on assistive devices, rehabilitation treatments, and physical training. Neuromechanical models represent the neuro-musculo-skeletal dynamics of the human body and can be simulated based on physical laws to predict body motions (Fig. 1). Although advancements in musculoskeletal modeling [1, 2] and physics simulation engines [3, 4, 5] allow us to simulate and analyze observed human motions, understanding and modeling human motor control remains a hurdle for accurately predicting motions. In particular, it is difficult to measure and interpret the biological neural circuits that underlie human motor control. To overcome this challenge, one can propose control models based on key features observed in animals and humans and investigate these models in neuromechanical simulations by comparing the simulation results to human data. With such physiologically plausible neuromechanical control models, today we can simulate many aspects of human motions, such as locomotion, in a predictive manner [6, 7, 8]. Despite this progress, developing controllers for more complex tasks, such as adapting to dynamic environments and those that require long-term planning, remains a challenge.

Training artificial neural networks using deep reinforcement learning (RL) in neuromechanical simulations provides an alternative way of developing control models. In contrast to developing a control model that captures certain physiological features and then running simulations to evaluate the results, deep RL can be thought of as training a black-box controller that produces motions of interest. Recent breakthroughs in deep learning make it possible to develop controllers with high-dimensional inputs and outputs that are applicable to human musculoskeletal models. Despite the discrepancy between artificial and biological neural networks, such means of developing versatile controllers could be useful in investigating human motor control [9]. For instance, one could train a controller (i. e., a policy in RL terminology) implemented on an artificial neural network using deep RL in a physiologically plausible simulation environment, and then investigate or reverse engineer the resulting network. One could also

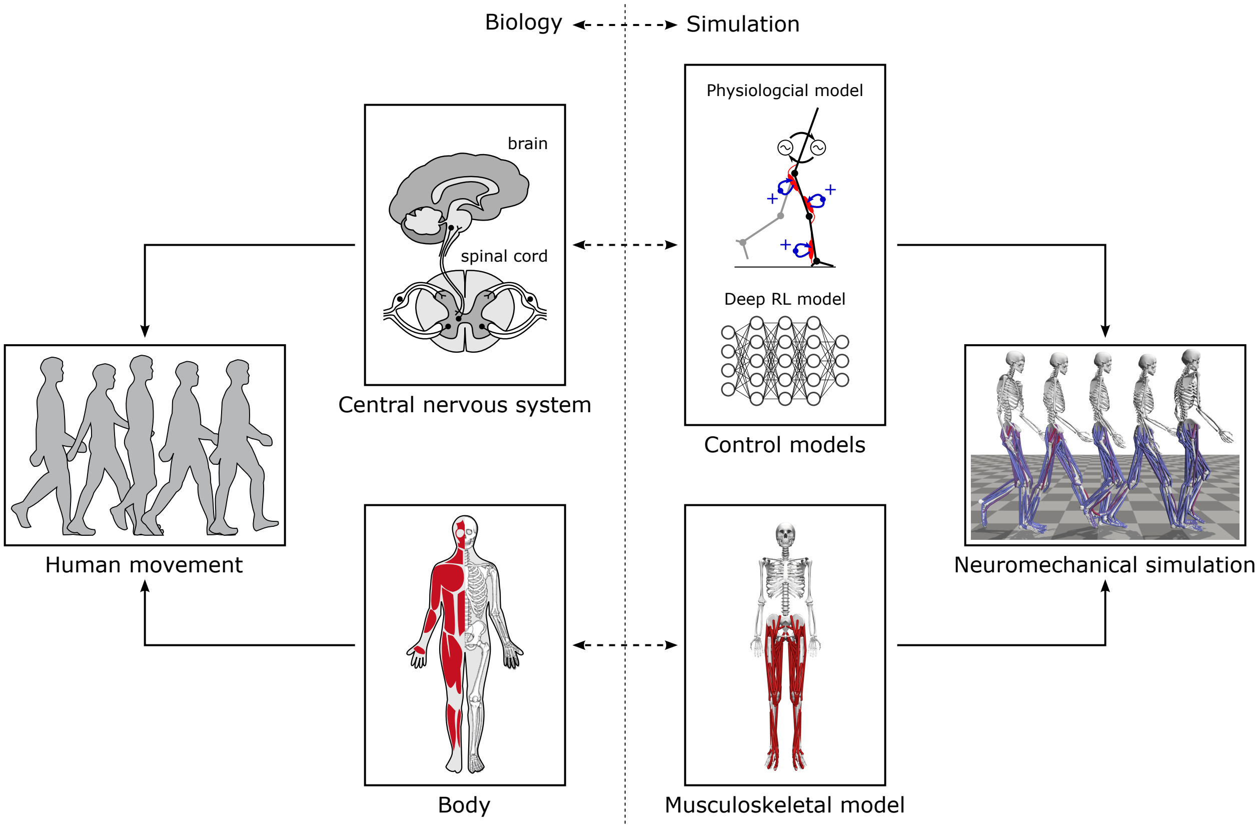

Figure 1: **Neuromechanical simulation.** A neuromechanical simulation consists of a control model and a musculoskeletal model that represent the central nervous system and the body, respectively. The control and musculoskeletal models are forward simulated based on physical laws to produce movements.

train controllers to mimic human motion (e.g., using imitation learning, where a controller is trained to replicate behaviors demonstrated by an expert [10]) or integrate an existing neuromechanical control model with artificial neural networks to study certain aspects of human motor control. While there are recent studies that used deep RL to produce human-like motions with musculoskeletal models [11, 12], little effort has been made to study the underlying control.

We organized the Learn to Move competition series to facilitate developing control models with advanced deep RL techniques in neuromechanical simulation. It has been an official competition at the NeurIPS conference for the past three years. We provided the neuromechanical simulation environment, OpenSim-RL, and participants developed locomotion controllers for a human musculoskeletal model. In the most recent competition, NeurIPS 2019: Learn to Move - Walk Around, the top teams adapted state-of-the-art deep RL techniques and successfully controlled a 3D human musculoskeletal model to follow target velocities by changing walking speed and direction as well as transitioning between walking and standing. Some of these locomotion behaviors were demonstrated in neuromechanical simulations for the first time without using reference motion data. While the solutions were not explicitly designed to model human learning or control, they provide means of developing control models that are capable of producing realistic complex motions.

This paper reviews neuromechanical simulations and deep RL, with a focus on the materials relevant to modeling the control of human locomotion. First, we review control models of human locomotion that have been studied in computer simulations and discuss how to evaluate their physiological plausibility. By reviewing human control models, we hope to inspire future studies using computational approaches, such as deep RL, to encode physiologically plausible features. We also introduce deep RL approaches for continuous control problems (the type of problem we must solve to predict human movement) and review their use in developing locomotion controllers. Then, we present the Learn to Move competition and discuss the successful approaches, simulation results, and their implications for locomotion research. We conclude by suggesting promising future directions for the field and outline our plan to extend the Learn to Move competition.

Computer simulations of human locomotion

This section reviews computer simulation studies that propose control models of human locomotion. We first present the building blocks of musculoskeletal simulations and their use in studying human motion. We next review the biological control hypotheses and neuromechanical control models that embed those hypotheses. We also briefly cover studies in computer graphics that have developed locomotion controllers for human characters. We close by discussing the means of evaluating the plausibility of control models and the limitations of current approaches.

Musculoskeletal simulations

A musculoskeletal model typically represents a human body with rigid segments and muscle-tendon actuators [13, 14, 15] (Fig. 2-a). The skeletal system is often modeled by rigid segments connected by rotational joints. Hill-type muscle models [16] are commonly used to actuate the joints, capturing the dynamics of biological muscles, including both active and passive contractile elements [17, 18, 19, 20] (Fig. 2-b). Hill-type muscle models can be used with models of metabolic energy consumption [21, 22, 23] and muscle fatigue [24, 25, 26] to estimate these quantities in simulations. Musculoskeletal parameter values are determined for average humans based on measurements from a large number of people and cadavers [27, 28, 29, 30] and can be customized to match an individual's height, weight, or CT and MRI scan data [31, 32]. OpenSim [1], which is the basis of the OpenSim-RL package [33] used in the Learn to Move competition, is an open-source software package broadly used in the biomechanics community (e.g., it has about 40,000 unique user downloads) to simulate musculoskeletal dynamics.

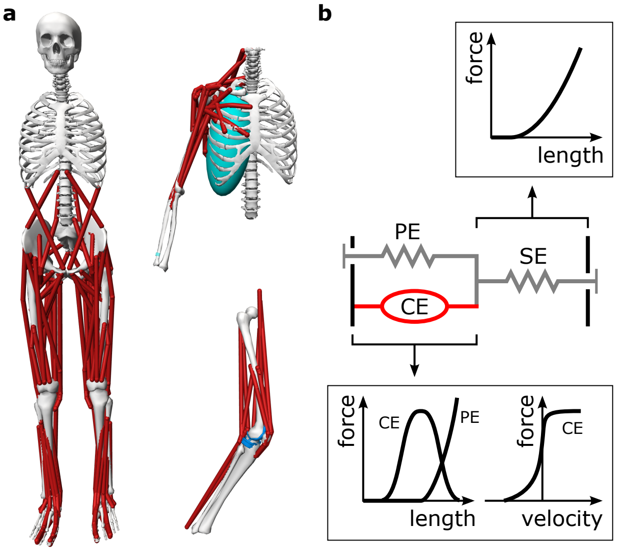

Figure 2: **Musculoskeletal models for studying human movement.** a. Models implemented in OpenSim [1] for a range of studies: lower-limb muscle activity in gait [34], shoulder muscle activity in upper-limb movements [15], and knee contact loads for various motions [35]. b. A Hill-type muscle model typically consists of a contractile element (CE), a parallel elastic element (PE), and a series elastic element (SE). The contractile element actively produces contractile forces that depend on its length and velocity and are proportional to the excitation signal. The passive elements act as non-linear springs where the force depends on their length.

Musculoskeletal simulations have been widely used to analyze recorded human motion. In one common approach, muscle activation patterns are found through various computational methods to enable a musculoskeletal model to track reference motion data, such as motion capture data and ground reaction forces [36, 37, 38], while achieving defined objectives, like minimizing muscle effort. The resulting simulation estimates body states, such as individual muscle forces, that are difficult to directly measure with an experiment. This approach has been extensively used to analyze human locomotion [36, 38], to estimate body state in real-time to control assistive devices [39, 40], and to predict effects of exoskeleton assistance [41] and surgical interventions [42] on muscle coordination. However, such simulations track recorded motions and do not produce new movement.

Alternatively, musculoskeletal simulations can produce motions without reference motion data using trajectory optimization methods. Instead of modeling the motor control system, this approach directly optimizes muscle activations that can actuate the musculoskeletal model and produce desired motions. While this approach does not take into account the structures or constraints of the human nervous system, the approach has been successful in reproducing well-practiced, or well-optimized, motor tasks. For example, normal walking and running motions can be produced by optimizing the muscle activations of a musculoskeletal model to move at a target speed with minimum muscle effort [43, 24, 25, 44]. Although this approach provides biomechanical insights by presenting theoretically optimal gaits for specific objectives, it is not suitable for predicting suboptimal behaviors that emerge from the underlying controller such as walking in unusual environments or reacting to unexpected perturbations.

Neuromechanical control models and simulations

A neuromechanical model includes a representation of a neural controller in addition to the musculoskeletal system (Fig. 1). To demonstrate that a controller can produce stable locomotion, neuromechanical models are typically tested in a forward physics simulation for multiple steps while dynamically interacting with the environment (e.g., the ground and the gravitational force). Researchers have used neuromechanical simulations to test gait assistive devices before developing hardware [45, 46] and to understand how changes in musculoskeletal properties affect walking performance [26, 7]. Moreover,

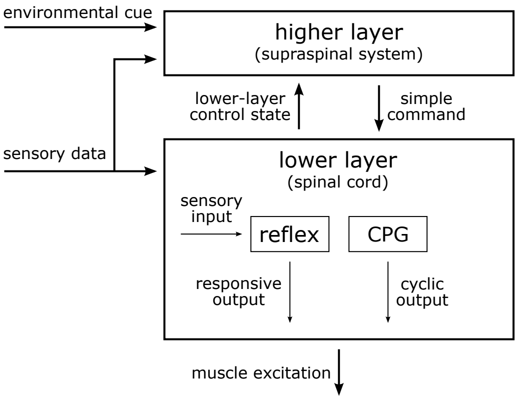

Figure 3: **Locomotion control.** The locomotion controller of animals is generally structured hierarchically with two layers. Reflexes and central pattern generators are the basic mechanisms of the lower layer controller.

the control model can be directly used to control bipedal robots [47, 48, 49] and assistive devices [50, 45, 51].

Modeling human motor control is crucial for a predictive neuromechanical simulation. However, most of our current understanding of human locomotion control is extrapolated from experimental studies of simpler animals [52, 53] as it is difficult to measure and interpret the activity of the millions of neurons involved in human motor control. Therefore, human locomotion control models have been proposed based on a few structural and functional control hypotheses that are shared in many animals (Fig. 3). First, locomotion in many animals can be interpreted as a hierarchical structure with two layers, where the lower layer generates basic motor patterns and the higher layer sends commands to the lower layer to modulate the basic patterns [52]. It has been shown in some vertebrates, including cats and lampreys, that the neural circuitry of the spinal cord, disconnected from the brain, can produce steady locomotion and can be modulated by electrical stimulation to change speed, direction and gait [54, 55]. Second, the lower layer seems to consist of two control mechanisms: reflexes [56, 57] and central pattern generators (CPGs) [58, 59]. In engineering terms, reflexes and CPGs roughly correspond to feedback and feedforward control, respectively. Muscle synergy, where a single pathway co-activates multiple muscles, has also been proposed as a lower layer control mechanism that reduces the degrees of freedom for complex control tasks [60, 61]. Lastly, there is a consensus that humans use minimum effort to conduct well-practiced motor tasks, such as walking [62, 63]. This consensus provides a basis for using energy or fatigue optimization [24, 25, 26] as a principled means of finding control parameter values.

Most neuromechanical control models are focused on lower layer control using spinal control mechanisms, such as CPGs and reflexes. CPG-based locomotion controllers consist of both CPGs and simple reflexes, where the CPGs, often modeled as mutually inhibiting neurons [64], generate the basic muscle excitation patterns. These CPG-based models [65, 66, 67, 68, 69, 8] demonstrated that stable locomotion can emerge from the entrainment between CPGs and the musculoskeletal system, which are linked by sensory feedback and joint actuation. CPG-based models also have been integrated with different control mechanisms, such as muscle synergies [68, 69, 8] and various sensory feedback circuits [66, 68]. On the other hand, reflex-based control models consist of simple feedback circuits without any temporal characteristics and demonstrate that CPGs are not necessary for producing stable locomotion. Reflex-based models [70, 18, 6, 71, 72] mostly use simple feedback laws based on sensory data accessible at the spinal cord such as proprioception (e.g., muscle length, speed and force) and cutaneous (e.g., foot contact and pressure) data [53, 57]. A reflex-based control model combined with a simple higher layer controller that regulates foot placement to maintain balance produced diverse locomotion behaviors including walking, running, and climbing stairs and slopes [6] and reacted to a range of unexpected

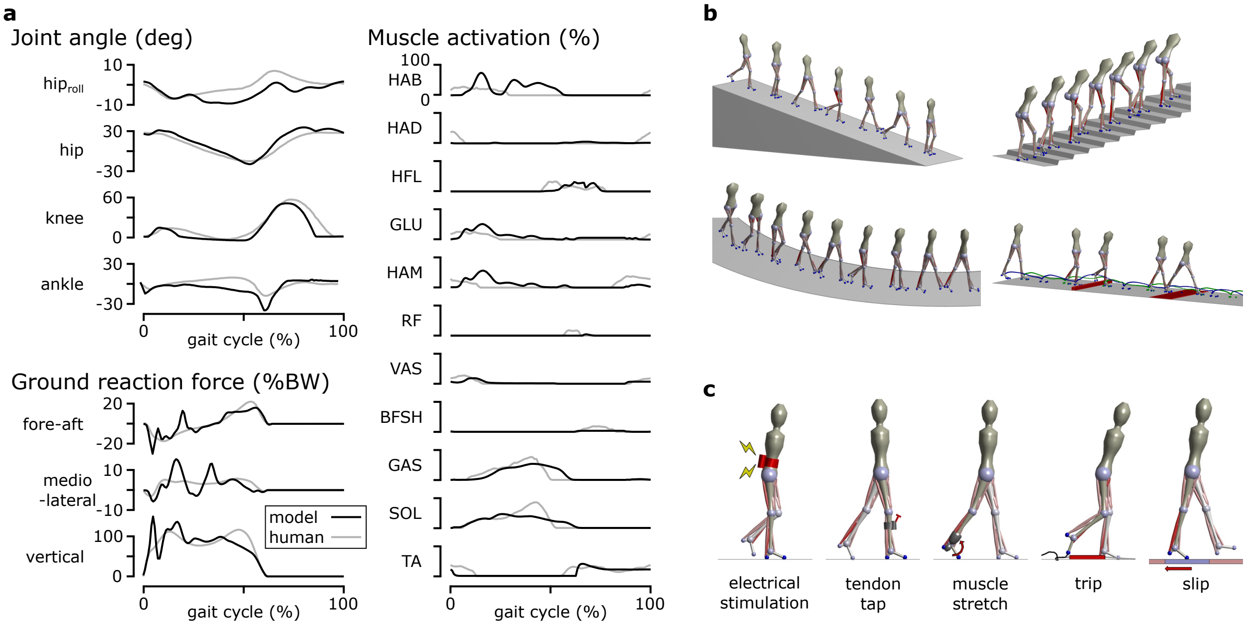

Figure 4: **Reflex-based neuromechanical model.** a. A reflex-based control model produced walking with human-like kinematics, dynamics, and muscle activations when optimized to walking with minimum metabolic energy consumption [6]. b. The model produced diverse locomotion behaviors when optimized at different simulation environments with different objectives. c. The same model optimized for minimum metabolic energy consumption reacted to various disturbances as observed in human experiments [73].

perturbations similarly to humans [73] (Fig. 4). Reflex-based controllers also have been combined with CPGs [71] and a deep neural network that operates as a higher layer controller [72] for more control functions, such as speed and terrain adaptation.

Human locomotion simulations for computer graphics

Human locomotion simulations have been widely developed in computer graphics to automate the process of generating human-like locomotion for computer characters. A number of controllers have been developed for characters to perform desired behaviors in physics simulations [74, 75, 76, 77, 78]. A variety of techniques have been proposed for simulating common behaviors, such as walking and running [79, 80, 81, 82]. Reference motions, such as motion capture data, were often used in the development process to produce more natural behaviors [83, 84, 85, 86]. Musculoskeletal models also have been used to achieve naturalistic motions [87, 88, 89], which makes them very close to neuromechanical simulations. The focus of these studies is producing natural motions rather than accurately representing the underlying biological system. However, the computer graphics studies and physiologically plausible neuromechanical simulations may converge as they progress to produce and model a wide variety of human motions.

Plausibility and limitations of control models

There are several ways to evaluate the physiological plausibility of control models. Plausibility of control models is of vital importance for the credibility of scientific knowledge the model produce and the implications of model predictions for rehabilitation research. Models that include neurophysiological details could be investigated more directly. For instance, a model that represents individual neurons and neural properties can be evaluated at the individual neuron level using neural recordings [90] and at the neuroanatomy level by assessing functions and connections of model parts that represent different areas of the brain and the spinal cord [91]. However, most of the control models tested in neuromechanical simulations focus on how motions emerge from motor control hypotheses rather than capturing the neural details. These models can be evaluated by the neuromechanical simulation results and the control features encoded in the models.

The plausibility of a neuromechanical control model can be assessed by the resulting simulation behavior. First of all, generating stable locomotion in neuromechanical simulations is a challenging control problem [92, 53] and thus has implications for the controller. For instance, a control model that cannot produce stable walking with physiological properties, such as nonlinear muscle dynamics and neural transmission delays, is likely missing some important aspects of human control [93]. Once motions are successfully simulated, they can be compared to measurable human data. We can say a model that produces walking with human-like kinematics, dynamics, and muscle activations is more plausible than one that does not. A model can be further compared with human control by evaluating its reactions to unexpected disturbances [73] and its adaptations in new conditions, such as musculoskeletal changes [26, 7], external assistance [45, 46], and different terrains.

We also can discuss whether a control model is constructed in a plausible manner. It is plausible for a control model to use sensory data that are known to be used in human locomotion [53, 57] and to work with known constraints, such as neural transmission delays. Models developed based on control hypotheses proposed by neuroscientists, such as CPGs and reflexes, partially inherit the plausibility of the corresponding hypotheses. Showing that human-like behaviors emerge from optimality principles that regulate human movements, such as minimum metabolic energy or muscle fatigue, also increases the plausibility of the control models [24, 25, 26].

Existing neuromechanical control models are mostly limited to modeling the lower layer control and producing steady locomotion behaviors. Most aspects of the motor learning process and the higher layer control are thus missing in current neuromechanical models. Motor learning occurs in daily life when acquiring new motor skills or adapting to environmental changes. For example, the locomotion control system adapts when walking on a slippery surface, moving a heavy load, walking on a split-belt treadmill [94, 95], and wearing an exoskeleton [96, 97]. The higher layer control processes environment cues, plans long-term motion strategies, and coordinates basic motor skills to navigate in dynamic and complex environments. While we will discuss other ideas for explicitly modeling motor learning and higher layer control in neuromechanical simulations in the sec:future section, deep RL may be an effective approach to developing controllers for challenging environments and motions.

Deep reinforcement learning for motor control

This section highlights the concepts from deep reinforcement learning relevant to developing models for motor control. We provide a brief overview of the terminology and problem formulations of RL and then cover selected state-of-art deep RL algorithms that are relevant to successful solutions in the Learn to Move competition. We also review human locomotion simulation studies that have used deep RL.

Deep reinforcement learning

Reinforcement learning is a machine learning paradigm for solving decision-making problems. The objective is to learn an optimal policy π that enables an agent to maximize its cumulative reward through interactions with its environment [98] (Fig. 5). For example, in the case of the Learn to Move competition, the environment was the musculoskeletal model and physics-based simulation environment, and higher cumulative rewards were given to solutions that better followed target velocities with lower muscle effort. Participants developed agents, which consists of a policy that controls the musculoskeletal model and a learning algorithms that trains the policy. For the general RL problem, at each timestep t, the agent receives an observation o_t (perception and proprioception data in the case of our competition; perception data includes information on the target velocities) and queries its policy π for an action a_t (excitation values of the muscles in the model) in response to that observation. An

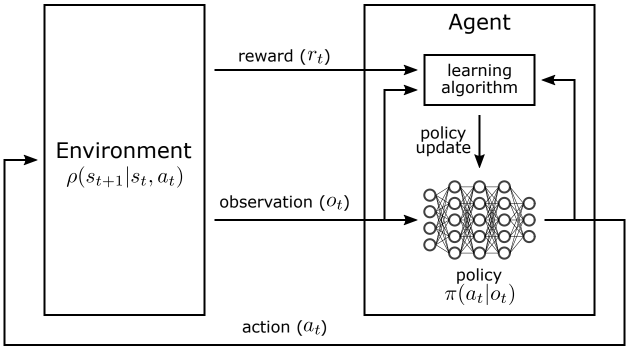

Figure 5: **Reinforcement learning.** In a typical RL process, an agent takes a reward and observation as input and trains a policy that outputs an action to achieve high cumulative rewards.

observation o_t is the full or partial information of the state s_t of the environment. The policy $\pi(a_t|o_t)$ can be either deterministic or stochastic, where a stochastic policy defines a distribution over actions a_t given a particular observation o_t [99]. The agent then applies the action in the environment, resulting in a transition to a new state s_{t+1} and a scalar reward $r_t = r(s_t, a_t, s_{t+1})$. The state transition is determined according to the dynamics model $\rho(s_{t+1}|s_t, a_t)$. The objective for the agent is to learn an optimal policy that maximizes its cumulative reward, or return.

One of the crucial design decisions in applying RL to a particular problem is the choice of policy representation. While a policy can be modeled by any class of functions that maps observations to actions, the recent use of deep neural networks to model policies demonstrated promising results in complex problems and has led to the emergence of the field of deep RL. Policies trained with deep RL methods achieved human-level performance on many of the 2600 Atari video games [100] and overtook world champion human players in the game of Go [101, 102].

State-of-the-art deep RL algorithms used in Learn to Move

Model-free deep RL algorithms (Fig. 6) are widely used for continuous control tasks, such as those considered in the Learn to Move competition where the actions are continuous values of muscle excitations. Model-free algorithms do not learn an explicit dynamics model of state transitions; instead, they directly learn a policy to maximize the expected return. In these continuous control tasks, the policy specifies actions that represent continuous quantities such as control forces or muscle excitations. Policy gradient algorithms incrementally improve a policy by first estimating the gradient of the expected return using trajectories collected from rollouts (forward simulations in our case) of the policy, and then updating the policy via gradient ascent [103]. While simple, the standard policy gradient update has several drawbacks, including stability and sample efficiency. First, the gradient estimator can have high variance, which can lead to unstable learning, and a good gradient estimate may require a large number of training samples. Popular algorithms such as TRPO [104] and PPO [105] improve the stability of policy gradient methods by limiting the change in the policy's behavior after each update step, as measured by the relative entropy between the policies [106]. Another limitation of policy gradient methods is their low sample efficiency, often requiring millions of samples to solve relatively simple tasks. Off-policy gradient algorithms, which utilize rollouts collected from previous policies, have been proposed to substantially reduce the number of samples required to learn effective policies [107, 108, 109]. Off-policy algorithms, such as DDPG [107], typically fit a Q-function, $Q(s, a)$, which is the expected return of performing an action a in the current state s. These methods differentiate the learned Q-function to approximate the policy gradient, then use it to update the policy. Recent off-policy methods, such as TD3 and SAC, build on this approach and propose several modifications that further improve sample efficiency and stability.

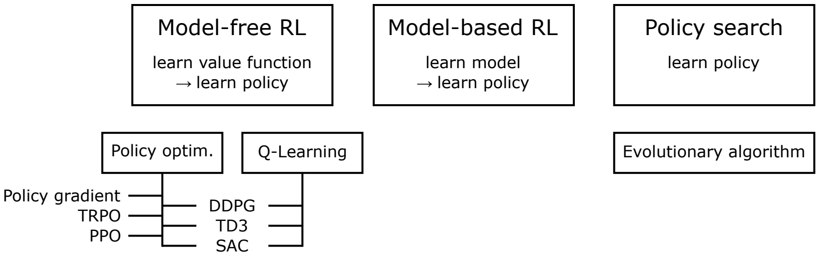

Figure 6: **Reinforcement learning algorithms for continuous action space.** The diagram is adapted from [110] and presents a partial taxonomy of RL algorithms for continuous control, or continuous action space. This focuses on a few modern deep RL algorithms and some traditional RL algorithms that are relevant to the algorithms used by the top teams in our competition. TRPO: trust region policy optimization [104]; PPO: proximal policy optimization [105]; DDPG: deep deterministic policy gradients [107]; TD3: twin delayed deep deterministic policy gradients [108]; SAC: soft-actor critic [109].

Deep RL for human locomotion control

Human motion simulation studies have used various forms of RL (Fig. 6). A number of works in neuromechanical simulation [67, 6] and computer graphics studies [87, 88] reviewed in the sec:simLoco section used policy search methods [111] with derivative-free optimization techniques, such as evolutionary algorithms, to tune their controllers. The control parameters are optimized by repeatedly running a simulation trial with a set of control parameters, evaluating the objective function from the simulation result, and updating the control parameters using an evolutionary algorithm [112]. This approach makes very minimal assumptions about the underlying system and can be effective for tuning controllers to perform a diverse array of skills [113, 6]. However, these algorithms often struggle with high dimensional parameter spaces (i.e., more than a couple of hundred parameters), therefore some care is required to design controllers that expose a relatively low-dimensional but expressive set of parameters for optimization. Selecting an effective set of parameters can require a great deal of expertise, and the selected set of parameters tend to be specific for particular skills, limiting the behaviors that can be reproduced by the character.

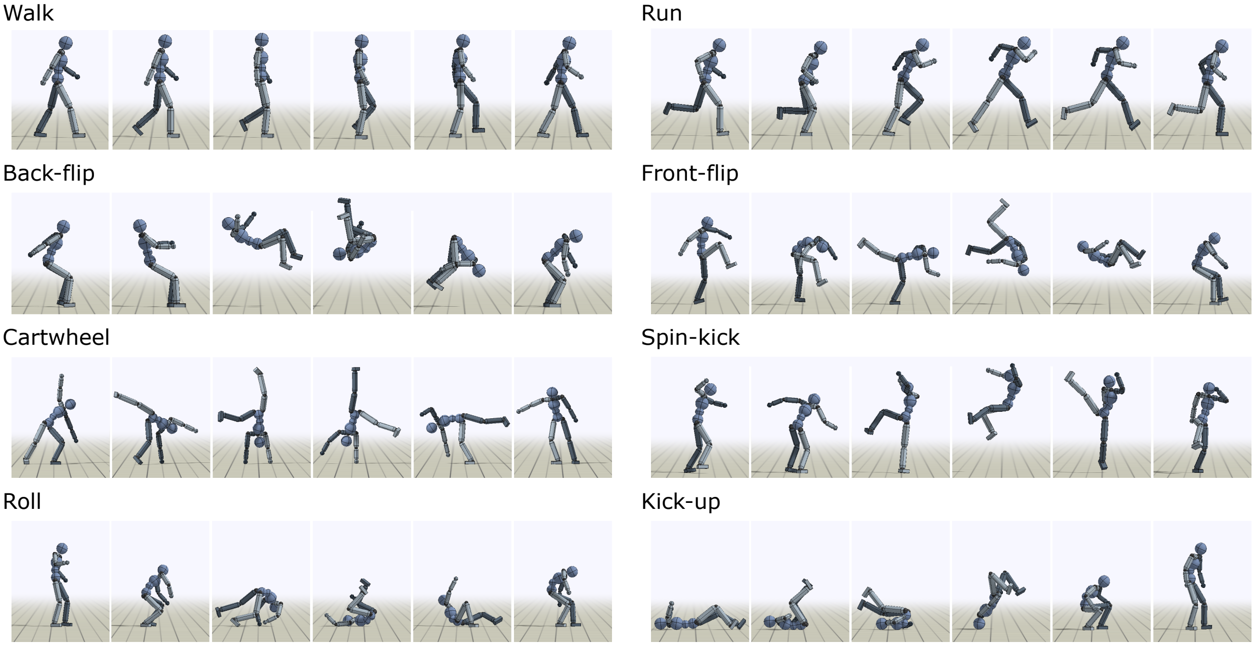

Figure 7: **Computer graphics characters performing diverse human motions.** Dynamic and acrobatic skills learned to mimic motion capture clips with RL in physics simulation [10].

Recently, deep RL techniques have demonstrated promising results for character animation, with policy optimization methods emerging as the algorithms of choice for many of these applications [103, 107, 105]. These methods have been effective for training controllers that can perform a rich repertoire of skills [114, 10, 115, 116, 117, 118]. One of the advantages of deep RL techniques is the ability to learn controllers that operate directly on high-dimensional, low-level representations of the underlying system, thereby reducing the need to manually design compact control representations for each skill. These methods have also been able to train controllers for interacting with complex environments [114, 119, 120], as well as for controlling complex musculoskeletal models [121, 11]. Reference motions continue to play a vital role in producing more naturalistic behavior in deep RL as a form of deep imitation learning, where the objective is designed to train a policy that mimics human motion capture data [10, 117, 11] (Fig. 7). These studies show the potential of using deep RL methods to develop versatile controllers for musculoskeletal models and studying human motor control.

Learn to Move competition

The potential synergy of neuromechanical simulations and deep RL methods in modeling human control motivated us to develop the OpenSim-RL simulation platform and to organize the Learn to Move competition series. OpenSim-RL [33] leverages OpenSim to simulate musculoskeletal models and OpenAI Gym, a widely used RL toolkit [122], to standardize the interface with state-of-the-art RL algorithms. OpenSim-RL is open-source and is provided as a Conda package [123], which has been downloaded about 42,000 times (counting multiple downloads from the same users) over the past three years. Training a controller for a human musculoskeletal model is a difficult RL problem considering the large-dimensional observation and action spaces, delayed and sparse rewards resulting from the highly non-linear and discontinuous dynamics, and the slow simulation of muscle dynamics. Therefore, we organized the Learn to Move competition series to crowd-source machine learning expertise in developing control models of human locomotion. The mission of the competition series is to bridge neuroscience, biomechanics, robotics, and machine learning to model human motor control.

The Learn to Move competition series has been held annually since 2017. It has been one of the official competitions at the NeurIPS conference, a major event at the intersection of machine learning and computational neuroscience. The first competition was NIPS 2017: Learning to Run [33, 124], and the task was to develop a controller for a given 2D human musculoskeletal model to run as fast as possible while avoiding obstacles. In the second competition, NeurIPS 2018: AI for Prosthetics Challenge [125], we provided a 3D human musculoskeletal model, where one leg was amputated and replaced with a passive ankle-foot prosthesis. The task was to develop a walking controller that could follow velocity commands, the magnitude and direction of which varied moderately. These two competitions together attracted about 1000 teams, primarily from the machine learning community, and established successful RL techniques which will be discussed in the sssec:solutions section. We designed the 2019 competition to build on knowledge gained from past competitions. For example, the challenge in 2018 demonstrated the difficulty of moving from 2D to 3D. Thus, to focus on controlling maneuvering in 3D, we designed the target velocity to be more challenging, while we removed the added challenge of simulating movement with a prosthesis. We also refined the reward function to encourage more natural human behaviors.

NeurIPS 2019: Learn to Move - Walk Around

Overview

NeurIPS 2019: Learn to Move - Walk Around was held online from June 6 to November 29 in 2019. The task was to develop a locomotion controller, which was scored based on its ability to meet target velocity vectors when applied in the provided OpenSim-RL simulation environment. The environment repository was shared on Github [126], the submission and grading were managed using the AIcrowd platform [127], and the project homepage provided documentation on the environment and the competition [128]. Participants were free to develop any type of controller that worked in the environment. We encouraged approaches other than brute force deep RL by providing human gait data sets of walking and running [129, 130, 131] and a 2D walking controller adapted from a reflex-based control model [6] that could be used for imitation learning or in developing a hierarchical control structure. There were two rounds. The top 50 teams in Round 1 were qualified to proceed to Round 2 and to participate in a paper submission track. RL experts were invited to review the papers based on the novelty of the approaches, and we selected the best and finalist papers based on the reviews. The prizes included GPUs (NVIDIA), a motion capture suit (Xsens), travel grants, and an invitation to submit a paper to the Journal of NeuroEngineering and Rehabilitation and the NeurIPS 2019 Deep RL Workshop.

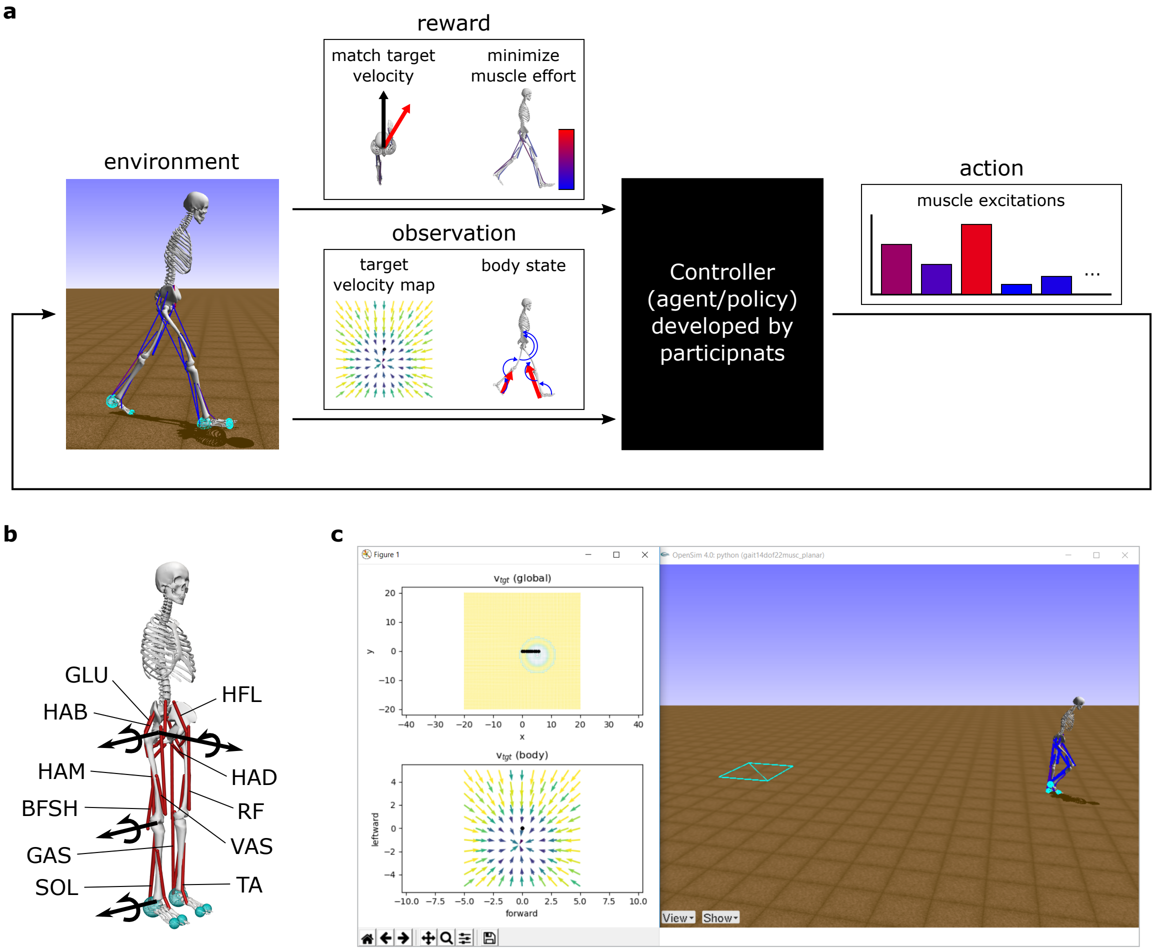

Figure 8: **OpenSim-RL environment for the NeurIPS 2019: Learn to Move – Walk Around competition.** a. A neuromechanical simulation environment is designed for a typical RL framework (Fig. 5). The environment took an action as input, simulated a musculoskeletal model for one time-step, and provided the resulting reward and observation. The action was excitation signals for the 22 muscles. The reward was designed so that solutions following target velocities with minimum muscle effort would achieve high total rewards. The observation consisted of a target velocity map and information on the body state. b. The environment included a musculoskeletal model that represents the human body. Each leg consisted of four rotational joints and 11 muscles. (HAB: hip abductor; HAD: hip adductor; HFL: hip flexor; GLU: glutei, hip extensor; HAM: hamstring, biarticular hip extensor and knee flexor; RF: rectus femoris, biarticular hip flexor and knee extensor; VAS: vastii, knee extensor; BFSH: short head of biceps femoris, knee flexor; GAS: gastrocnemius, biarticular knee flexor and ankle extensor; SOL: soleus, ankle extensor; TA: tibialis anterior, ankle flexor) c. The simulation environment provided a real-time visualization of the simulation to users. The global map of target velocities is shown at the top-left. The bottom-left shows its local map, which is part of the input to the controller. The right visualizes the motion of the musculoskeletal model.

In total, 323 teams participated in the competition and submitted 1448 solutions. In Round 2, the top three teams succeeded in completing the task and received high scores (mean total rewards larger than 1300 out of 1500). Five papers were submitted, and we selected the best paper [132] along with two more finalist papers [133, 134]. The three finalist papers came from the top three teams, where the best paper was from the top team.

Simulation environment

The OpenSim-RL environment included a physics simulation of a 3D human musculoskeletal model, target velocity commands, a reward system, and a visualization of the simulation (Fig. 8). The 3D musculoskeletal model had seven segments connected with eight rotational joints and actuated by 22 muscles. Each foot segment had three contact spheres that dynamically interacted with the ground. A user-developed policy could observe 97-dimensional body sensory data and 242-dimensional target velocity map and produced a 22-dimensional action containing the muscle excitation signals. The reward was designed to give high total rewards for solutions that followed target velocities with minimum muscle effort. The details of the reward were set to result in human-like walking based on previous neuromechanical simulation studies [6, 26]. The mean total reward of five trials with different target velocities was used for ranking. The full description of the simulation environment is provided on our project homepage [135].

Top solutions and results

Various RL techniques have been effectively used since the first competition [124, 125], including frame skipping, discretization of the action space, and reward shaping. These are practical techniques that constrain the problem in certain ways to encourage an agent to search successful regions faster in the initial stages of training. Frame skipping repeats a selected action for a given number of frames instead of operating the controller every frame [133]. This technique reduces the sampling rate and thus computations while maintaining a meaningful representation of observations and control. Discretization of the muscle excitations constrains the action space and thus the search space, which can lead to much faster training. In the extreme case, binary discretization (i.e., muscles were either off or fully activated) was used by some teams in an early stage of training. Reward shaping modifies the reward function provided by the environment to encourage an agent to explore certain regions of the solution space. For example, a term added to the reward function that penalizes crossover steps encouraged controllers to produce more natural steps [133, 134]. Once agents found solutions that seem to achieve intended behaviors with these techniques, they typically were further tuned with the original problem formulation.

A common trend across the finalists in 2019 was the use of off-policy methods. The first place entry by Zhou et al. [132] used DDPG [107], the second place entry by Kolesnikov and Hrinchuk [133] used TD3 [108], and the third place entry by Akimov [134] used SAC [109]. Since off-policy algorithms use data collected with previous policies for training, they can be substantially more sample efficient than their on-policy counterparts and could help to compensate for the computationally expensive simulation. Off-policy algorithms are also more amenable to distributed training, since data-collection and model updates can be performed asynchronously. Kolesnikov and Hrinchuk [133] leveraged this property of off-policy methods to implement a population-based distributed training framework, which used an ensemble of agents whose experiences were collected into a shared replay buffer that stored previously collected *(observation, action, reward, next observation)* pairs. Each agent was configured with different hyperparameter settings and was trained using the data collected from all agents. This, in turn, improved the diversity of the data that was used to train each policy and also improved the exploration of different strategies for solving a particular task.

Curriculum learning [136] was also used by the top teams. Curriculum learning is a training method where a human developer designs a curriculum that consists of a series of simpler tasks that eventually lead to the original task that is challenging to train from scratch. Akimov [134] first trained a policy to walk and then to follow target velocities. Similarly, Zhou et al. [132] trained a policy for normal speed walking by first training it to run at high speed, then to run at slower speeds, and eventually to walk at normal speed. They found that the policy trained through this process resulted in more natural gaits than policies that were directly trained to walk at normal speeds. This is probably because there is a limited set of very high-speed gaits that are close to human sprinting, and starting from this human-like sprinting gait could have guided the solution to a more natural walking gait out of a large variety of slow gaits, some of which are unnatural and ineffective local minima. Then they obtained their final solution policy by training this basic walking policy to follow target velocities and to move with minimum muscle effort.

The winning team, Zhou et al., proposed risk averse value expansion (RAVE), a hybrid approach of model-based and model-free RL [132]. Their method fits an ensemble of dynamics models (i.e. models of the environment) to data collected from the agent's interaction with the environment, and then uses the learned models to generate imaginary trajectories for training a Q-function. This model-based approach can substantially improve sample efficiency by synthetically generating a large volume of data but can also be susceptible to bias from the learned models, which can negatively impact performance. To mitigate potential issues due to model bias, RAVE uses an ensemble of dynamics models to estimate the confidence bound of the predicted values and then trains a policy using DDPG to maximize the confidence lower bound. Their method achieved impressive results on the competition tasks and also demonstrated competitive performance on standard OpenAI Gym benchmarks [122] compared to state-of-the-art algorithms [132].

Implications for human locomotion control

The top solution [132] shows that it is possible to produce many locomotion behaviors with the given 3D human musculoskeletal model, despite its simplifications. The musculoskeletal model simplifies the human body by, for example, representing the upper body and the pelvis as a single segment. Moreover, the whole body does not have any degree of freedom for internal yaw motion (Fig. 8-a). Such a model was selected for the competition as it can produce many locomotion behaviors including walking, running, stair and slope climbing, and moderate turning as shown in a previous study [6]. On the other hand, the missing details of the musculoskeletal model could have been crucial for generating other behaviors like sharp turning motions and gait initiation. However, the top solution was able to initiate walking from standing, quickly turn towards a target (e.g., turn 180° in one step; Fig. 9), walk to the target at commanded speeds, and stop and stand at the target. To our knowledge, it is the first demonstration of rapid turning motions with a musculoskeletal model with no internal yaw degree of freedom. The solution used a strategy that is close to a step-turn rather than a spin-turn, and it will be interesting to further investigate how the simulated motion compares with human turning [137, 138].

The top solutions had some limitations in producing human-like motions. In the top solution [132], the human model first turned to face the target then walked forward towards the target with a relatively natural gait. However, the gait was not as close to human walking as motions produced by previous neuromechanical models and trajectory optimization [6, 44]. This is not surprising as the controllers for the competition needed to cover a broad range of motions, and thus were more difficult to fully optimize

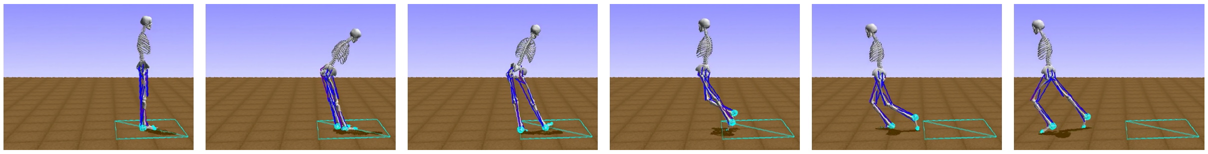

Figure 9: **Rapid turning motion.** The top solution can make the musculoskeletal model with no internal yaw degree of freedom to turn 180° in a single step. Snapshots were taking every 0.4 s.

for specific motions. The second and third solutions [133, 134] were further from human motions as they gradually moved towards a target often using side steps. As policy gradient methods use gradient descent, they often get stuck at local optima resulting in suboptimal motions [120] even though natural gaits are more efficient and agile. Although the top solution overcame some of these suboptimal gaits through curriculum learning, better controllers could be trained by utilizing imitation learning for a set of optimal motions [10, 11, 12] or by leveraging control models that produce natural gaits [18, 6]. Different walking gaits, some of which are possibly suboptimal, are also observed in toddlers during the few months of extensive walking experience [139, 140], and interpreting this process with an RL framework will be instructive to understanding human motor learning.

Future directions

Various deep reinforcement learning approaches, such as imitation learning and hierarchical learning, approaches could be used to produce more complex motions and their coordination. The top solutions in the competitions were able to control standing, walking and running behaviors, which is impressive but only covers a small portion of human motion. Parkour athletes, for example, can plan and execute jumping, vaulting, climbing, rolling, and many other acrobatic motions to move in complex environments, which would be very difficult to perform with brute force RL methods. Imitation learning [10, 117, 11] could be used to train multiple networks to master a set of acrobatic skills (Fig. 7). These networks of motion primitives can then be part of a hierarchical controller [141, 142, 143], where a higher layer controller coordinates the motion skills. More control layers that analyze dynamic scenes and plan longer-term motion sequences [144, 145] can be added if a complex strategy is required for the task. We will promote research in developing all these aspects of control by designing a contest that requires complex motions and long-term motion planning. The task can be something like the World Chase Tag competition, where two athletes take turns to tag the opponent, using athletic movements, in an arena filled with obstacles [146]. To this end, we plan to develop a simulation environment with a faster physics engine that can handle multi-body contacts between human models and obstacles [3, 4, 5], provide a human musculoskeletal model with an articulated upper body [88, 11], and design a visuomotor control interface [147].

As discussed in the ssec:plausibility section, the physiologically plausible control models are currently limited to representing the lower layer control and omit many aspects of the motor learning process and the higher layer control. Motor learning paradigms have been proposed to explain the transient behaviors observed while acquiring new motor skills or adapting to environment changes [148, 149]. Integrating these paradigms into locomotion control models will substantially expand the realm of human motions that neuromechanical simulations can represent. Gait adaptation reported during walking on a split-belt treadmill [94, 95] are a benchmark for evaluating motor learning simulations, and simulations that correctly predict the training process of walking with exoskeletons [96, 97] could change the way we develop assistive devices. Deep RL methods may be able to train control models that perform the functions of both higher layer and lower layer controls. However, integrating physiologically plausible control models with deep RL methods can have both predictive and computational advantages. As there are CPG- and reflex-based lower layer control models proposed to represent some of the physiological control constraints [67, 6], combining these lower-layer control models with a high-layer controller developed with deep RL may have some utility in producing motions closer to humans in novel scenarios. Moreover, it may be easier to train this higher-layer deep neural network than one without such structured lower-layer control as the solution space could be constrained to reasonable movements. Similarly, the action space can be reduced by having a muscle synergy layer between the deep neural network and muscle excitations. In future competitions, we plan to implement and share more physiologically plausible control models in OpenSim-RL and organize a separate paper submission track to acknowledge approaches that encode physiological features.

The importance of physiological plausibility of predictive simulations is not limited to control models and simulation results but also applies to the musculoskeletal simulation environment. For instance, there are a variety of muscle models [20, 150, 151] that represent biological muscles at different levels of detail and focus on different properties, and investigating the effect of these models in whole-body motion will be instructive. Body motion can be affected by other physical and physiological details as well, such as the number of muscles and body segments used to represent the musculoskeletal system [11], soft tissue [152] and its interaction with assistive devices [153], muscle fatigue [154], and musculoskeletal properties that vary across individuals [155] and health conditions [156, 157]. Modeling these details requires sufficient domain knowledge in each field. In the long term, we aim to modularize our simulation framework so that it will be easier for researchers to focus on their expertise, for example in implementing and testing different muscle models, while using other modules (e.g., control, training, and environment modules) in our framework. We could collect various control models and RL algorithms through the Learn to Move competition and share them as part of the modular framework that will make motion simulations easier for non-RL experts.

Conclusion

In this article, we reviewed neuromechanical simulations and deep reinforcement learning with a focus on human locomotion. Neuromechanical simulations provide a means to evaluate control models, and deep RL is a promising tool to develop control models for complex movements. We hope to see more interdisciplinary studies and collaboration that uses advanced machine learning techniques to develop and evaluate physiologically plausible control models. Such studies could significantly advance our ability to model control and predict human behaviors. We plan to continue to develop and disseminate the Learn to Move competition and its accompanying simulation platform to facilitate these advancements toward predictive neuromechanical simulations for rehabilitation treatment and assistive devices.

Abbreviations

CPG: central pattern generators; RAVE: risk averse value expansion; RL: reinforcement learning

Acknowledgements

We thank our partners and sponsors: AIcrowd, Google, Nvidia, Xsens, Toyota Research Institute, the NeurIPS Deep RL Workshop team, and the Journal of NeuroEngineering and Rehabilitation. We thank all the external reviewers who provide us valuable feedback on the papers submitted to the competition. And most importantly, we thank all the researchers who have competed in the Learn to Move competition.

Author's contributions

All authors are members of the team that organized the Learn to Move competition. SS, ŁK, XBP, CO and JH drafted the manuscript. All authors edited and approved the submitted manuscript.

Funding

The Learn to Move competition series were co-organized by the Mobilize Center, a National Institutes of Health Big Data to Knowledge Center of Excellence supported through Grant U54EB020405. The first author was supported by the National Institutes of Health under Grant K99AG065524.

Availability of data and materials

The environment repository and documentation of NeurIPS 2019: Learn to Move - Walk Around is shared on Github [126] and the project homepage [128].

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

References

[1] A. Seth, J. L. Hicks, T. K. Uchida, A. Habib, C. L. Dembia, J. J. Dunne, C. F. Ong, M. S. DeMers, A. Rajagopal, M. Millard, et al., Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement, PLoS computational biology 14 (2018).

[2] C. L. Dembia, N. A. Bianco, A. Falisse, J. L. Hicks, S. L. Delp, Opensim moco: Musculoskeletal optimal control, BioRxiv (2019) 839381.

[3] E. Todorov, T. Erez, Y. Tassa, Mujoco: A physics engine for model-based control, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 5026–5033.

[4] J. Lee, M. Grey, S. Ha, T. Kunz, S. Jain, Y. Ye, S. Srinivasa, M. Stilman, C. Liu, Dart: Dynamic animation and robotics toolkit, Journal of Open Source Software 3 (2018) 500.

[5] J. Hwangbo, J. Lee, M. Hutter, Per-contact iteration method for solving contact dynamics, IEEE Robotics and Automation Letters 3 (2018) 895–902.

[6] S. Song, H. Geyer, A neural circuitry that emphasizes spinal feedback generates diverse behaviours of human locomotion, The Journal of physiology 593 (2015) 3493–3511.

[7] C. F. Ong, T. Geijtenbeek, J. L. Hicks, S. L. Delp, Predicting gait adaptations due to ankle plantarflexor muscle weakness and contracture using physics-based musculoskeletal simulations, BioRxiv (2019) 597294.

[8] D. Tamura, S. Aoi, T. Funato, S. Fujiki, K. Senda, K. Tsuchiya, Contribution of phase resetting to adaptive rhythm control in human walking based on the phase response curves of a neuromusculoskeletal model, Frontiers in Neuroscience (2020).

[9] B. A. Richards, T. P. Lillicrap, P. Beaudoin, Y. Bengio, R. Bogacz, A. Christensen, C. Clopath, R. P. Costa, A. de Berker, S. Ganguli, et al., A deep learning framework for neuroscience, Nature neuroscience 22 (2019) 1761–1770.

[10] X. B. Peng, P. Abbeel, S. Levine, M. van de Panne, Deepmimic: Example-guided deep reinforcement learning of physics-based character skills, ACM Transactions on Graphics (TOG) 37 (2018) 1–14.

[11] S. Lee, M. Park, K. Lee, J. Lee, Scalable muscle-actuated human simulation and control, ACM Transactions on Graphics (TOG) 38 (2019) 1–13.

[12] A. S. Anand, G. Zhao, H. Roth, A. Seyfarth, A deep reinforcement learning based approach towards generating human walking behavior with a neuromuscular model, in: 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), IEEE, pp. 537–543.

[13] E. M. Arnold, S. R. Ward, R. L. Lieber, S. L. Delp, A model of the lower limb for analysis of human movement, Annals of biomedical engineering 38 (2010) 269–279.

[14] A. Rajagopal, C. L. Dembia, M. S. DeMers, D. D. Delp, J. L. Hicks, S. L. Delp, Full-body musculoskeletal model for muscle-driven simulation of human gait, IEEE transactions on biomedical engineering 63 (2016) 2068–2079.

[15] A. Seth, M. Dong, R. Matias, S. L. Delp, Muscle contributions to upper-extremity movement and work from a musculoskeletal model of the human shoulder, Frontiers in neurorobotics 13 (2019) 90.

[16] 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 (1938) 136–195.

[17] F. E. Zajac, Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control., Critical reviews in biomedical engineering 17 (1989) 359–411.

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

[19] M. Millard, T. Uchida, A. Seth, S. L. Delp, Flexing computational muscle: modeling and simulation of musculotendon dynamics, Journal of biomechanical engineering 135 (2013).

[20] D. Haeufle, M. Günther, A. Bayer, S. Schmitt, Hill-type muscle model with serial damping and eccentric force–velocity relation, Journal of biomechanics 47 (2014) 1531–1536.

[21] L. J. Bhargava, M. G. Pandy, F. C. Anderson, A phenomenological model for estimating metabolic energy consumption in muscle contraction, Journal of biomechanics 37 (2004) 81–88.

[22] B. R. Umberger, Stance and swing phase costs in human walking, Journal of the Royal Society Interface 7 (2010) 1329–1340.

[23] A. D. Koelewijn, D. Heinrich, A. J. Van Den Bogert, Metabolic cost calculations of gait using musculoskeletal energy models, a comparison study, PloS one 14 (2019).

[24] M. Ackermann, A. J. Van den Bogert, Optimality principles for model-based prediction of human gait, Journal of biomechanics 43 (2010) 1055–1060.

[25] R. H. Miller, B. R. Umberger, J. Hamill, G. E. Caldwell, Evaluation of the minimum energy hypothesis and other potential optimality criteria for human running, Proceedings of the Royal Society B: Biological Sciences 279 (2012) 1498–1505.

[26] S. Song, H. Geyer, Predictive neuromechanical simulations indicate why walking performance declines with ageing, The Journal of physiology 596 (2018) 1199–1210.

[27] R. Chandler, C. E. Clauser, J. T. McConville, H. Reynolds, J. W. Young, Investigation of inertial properties of the human body, Technical Report, Air Force Aerospace Medical Research Lab Wright-Patterson AFB OH, 1975.

[28] J. Visser, J. Hoogkamer, M. Bobbert, P. Huijing, Length and moment arm of human leg muscles as a function of knee and hip-joint angles, European journal of applied physiology and occupational physiology 61 (1990) 453–460.

[29] G. Yamaguchi, A survey of human musculotendon actuator parameters, Multiple muscle systems: Biomechanics and movement organization (1990).

[30] S. R. Ward, C. M. Eng, L. H. Smallwood, R. L. Lieber, Are current measurements of lower extremity muscle architecture accurate?, Clinical orthopaedics and related research 467 (2009) 1074–1082.

[31] L. Scheys, D. Loeckx, A. Spaepen, P. Suetens, I. Jonkers, Atlas-based non-rigid image registration to automatically define line-of-action muscle models: a validation study, Journal of biomechanics 42 (2009) 565–572.

[32] B. J. Fregly, M. L. Boninger, D. J. Reinkensmeyer, Personalized neuromusculoskeletal modeling to improve treatment of mobility impairments: a perspective from european research sites, Journal of neuroengineering and rehabilitation 9 (2012) 18.

[33] Ł. Kidziński, S. P. Mohanty, C. F. Ong, J. L. Hicks, S. F. Carroll, S. Levine, M. Salathé, S. L. Delp, Learning to run challenge: Synthesizing physiologically accurate motion using deep reinforcement learning, in: The NIPS'17 Competition: Building Intelligent Systems, Springer, 2018, pp. 101–120.

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

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

[36] F. De Groote, A. Van Campen, I. Jonkers, J. De Schutter, Sensitivity of dynamic simulations of gait and dynamometer experiments to hill muscle model parameters of knee flexors and extensors, Journal of biomechanics 43 (2010) 1876–1883.

[37] D. G. Thelen, F. C. Anderson, S. L. Delp, Generating dynamic simulations of movement using computed muscle control, Journal of biomechanics 36 (2003) 321–328.

[38] F. De Groote, A. L. Kinney, A. V. Rao, B. J. Fregly, Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem, Annals of biomedical engineering 44 (2016) 2922–2936.

[39] E. E. Cavallaro, J. Rosen, J. C. Perry, S. Burns, Real-time myoprocessors for a neural controlled powered exoskeleton arm, IEEE Transactions on Biomedical Engineering 53 (2006) 2387–2396.

[40] N. Lotti, M. Xiloyannis, G. Durandau, E. Galofaro, V. Sanguineti, L. Masia, M. Sartori, Adaptive model-based myoelectric control for a soft wearable arm exosuit: A new generation of wearable robot control, IEEE Robotics & Automation Magazine (2020).

[41] T. K. Uchida, A. Seth, S. Pouya, C. L. Dembia, J. L. Hicks, S. L. Delp, Simulating ideal assistive devices to reduce the metabolic cost of running, PloS one 11 (2016).

[42] M. D. Fox, J. A. Reinbolt, S. Õunpuu, S. L. Delp, Mechanisms of improved knee flexion after rectus femoris transfer surgery, Journal of biomechanics 42 (2009) 614–619.

[43] F. C. Anderson, M. G. Pandy, Dynamic optimization of human walking, J. Biomech. Eng. 123 (2001) 381–390.

[44] A. Falisse, G. Serrancolí, C. L. Dembia, J. Gillis, I. Jonkers, F. De Groote, Rapid predictive simulations with complex musculoskeletal models suggest that diverse healthy and pathological human gaits can emerge from similar control strategies, Journal of The Royal Society Interface 16 (2019) 20190402.

[45] N. Thatte, H. Geyer, Toward balance recovery with leg prostheses using neuromuscular model control, IEEE Transactions on Biomedical Engineering 63 (2015) 904–913.

[46] K. Seo, S. Hyung, B. K. Choi, Y. Lee, Y. Shim, A new adaptive frequency oscillator for gait assistance, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 5565–5571.

[47] Z. Batts, S. Song, H. Geyer, Toward a virtual neuromuscular control for robust walking in bipedal robots, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 6318–6323.

[48] N. Van der Noot, A. J. Ijspeert, R. Ronsse, Neuromuscular model achieving speed control and steering with a 3d bipedal walker, Autonomous Robots 43 (2019) 1537–1554.

[49] G. Zhao, F. Szymanski, A. Seyfarth, Bio-inspired neuromuscular reflex based hopping controller for a segmented robotic leg, Bioinspiration & Biomimetics 15 (2020) 026007.

[50] M. F. Eilenberg, H. Geyer, H. Herr, Control of a powered ankle–foot prosthesis based on a neuromuscular model, IEEE transactions on neural systems and rehabilitation engineering 18 (2010) 164–173.

[51] A. R. Wu, F. Dzeladini, T. J. Brug, F. Tamburella, N. L. Tagliamonte, E. H. Van Asseldonk, H. Van Der Kooij, A. J. Ijspeert, An adaptive neuromuscular controller for assistive lower-limb exoskeletons: A preliminary study on subjects with spinal cord injury, Frontiers in neurorobotics 11 (2017) 30.

[52] T. Orlovsky, Orlovskiĭ, Neuronal control of locomotion: from mollusc to man, ????

[53] C. Capaday, The special nature of human walking and its neural control, TRENDS in Neurosciences 25 (2002) 370–376.

[54] D. M. Armstrong, The supraspinal control of mammalian locomotion., The Journal of physiology 405 (1988) 1–37.

[55] M. G. Sirota, G. V. Di Prisco, R. Dubuc, Stimulation of the mesencephalic locomotor region elicits controlled swimming in semi-intact lampreys, European Journal of Neuroscience 12 (2000) 4081–4092.

[56] F. Clarac, Some historical reflections on the neural control of locomotion, Brain research reviews 57 (2008) 13–21.

[57] H. Hultborn, Spinal reflexes, mechanisms and concepts: from eccles to lundberg and beyond, Progress in neurobiology 78 (2006) 215–232.

[58] M. MacKay-Lyons, Central pattern generation of locomotion: a review of the evidence, Physical therapy 82 (2002) 69–83.

[59] K. Minassian, U. S. Hofstoetter, F. Dzeladini, P. A. Guertin, A. Ijspeert, The human central pattern generator for locomotion: Does it exist and contribute to walking?, The Neuroscientist 23 (2017) 649–663.

[60] F. Lacquaniti, Y. P. Ivanenko, M. Zago, Patterned control of human locomotion, The Journal of physiology 590 (2012) 2189–2199.

[61] E. Bizzi, V. C. Cheung, The neural origin of muscle synergies, Frontiers in computational neuroscience 7 (2013) 51.

[62] H. J. Ralston, Energy-speed relation and optimal speed during level walking, Internationale Zeitschrift für Angewandte Physiologie Einschliesslich Arbeitsphysiologie 17 (1958) 277–283.

[63] E. Todorov, Optimality principles in sensorimotor control, Nature neuroscience 7 (2004) 907–915.

[64] K. Matsuoka, Sustained oscillations generated by mutually inhibiting neurons with adaptation, Biological cybernetics 52 (1985) 367–376.

[65] G. Taga, Y. Yamaguchi, H. Shimizu, Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment, Biological cybernetics 65 (1991) 147–159.

[66] N. Ogihara, N. Yamazaki, Generation of human bipedal locomotion by a bio-mimetic neuro-musculo-skeletal model, Biological cybernetics 84 (2001) 1–11.

[67] K. Hase, K. Miyashita, S. Ok, Y. Arakawa, Human gait simulation with a neuromusculoskeletal model and evolutionary computation, The Journal of Visualization and Computer Animation 14 (2003) 73–92.

[68] S. Jo, S. G. Massaquoi, A model of cerebrocerebello-spinomuscular interaction in the sagittal control of human walking, Biological cybernetics 96 (2007) 279–307.

[69] S. Aoi, T. Ohashi, R. Bamba, S. Fujiki, D. Tamura, T. Funato, K. Senda, Y. Ivanenko, K. Tsuchiya, Neuromusculoskeletal model that walks and runs across a speed range with a few motor control parameter changes based on the muscle synergy hypothesis, Scientific reports 9 (2019) 1–13.

[70] M. Günther, H. Ruder, Synthesis of two-dimensional human walking: a test of the λ-model, Biological cybernetics 89 (2003) 89–106.

[71] F. Dzeladini, J. Van Den Kieboom, A. Ijspeert, The contribution of a central pattern generator in a reflex-based neuromuscular model, Frontiers in human neuroscience 8 (2014) 371.

[72] J. Wang, W. Qin, L. Sun, Terrain adaptive walking of biped neuromuscular virtual human using deep reinforcement learning, IEEE Access 7 (2019) 92465–92475.

[73] S. Song, H. Geyer, Evaluation of a neuromechanical walking control model using disturbance experiments, Frontiers in computational neuroscience 11 (2017) 15.

[74] P. Faloutsos, M. Van de Panne, D. Terzopoulos, Composable controllers for physics-based character animation, in: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 251–260.

[75] A. C. Fang, N. S. Pollard, Efficient synthesis of physically valid human motion, ACM Transactions on Graphics (TOG) 22 (2003) 417–426.

[76] K. Wampler, Z. Popović, Optimal gait and form for animal locomotion, ACM Transactions on Graphics (TOG) 28 (2009) 1–8.

[77] S. Coros, A. Karpathy, B. Jones, L. Reveret, M. Van De Panne, Locomotion skills for simulated quadrupeds, ACM Transactions on Graphics (TOG) 30 (2011) 1–12.

[78] S. Levine, J. Popović, Physically plausible simulation for character animation, in: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association, pp. 221–230.

[79] K. Yin, K. Loken, M. Van de Panne, Simbicon: Simple biped locomotion control, ACM Transactions on Graphics (TOG) 26 (2007) 105–es.

[80] C.-C. Wu, V. Zordan, Goal-directed stepping with momentum control, in: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 113–118.

[81] M. De Lasa, I. Mordatch, A. Hertzmann, Feature-based locomotion controllers, ACM Transactions on Graphics (TOG) 29 (2010) 1–10.

[82] S. Coros, P. Beaudoin, M. Van de Panne, Generalized biped walking control, ACM Transactions on Graphics (TOG) 29 (2010) 1–9.

[83] V. B. Zordan, J. K. Hodgins, Motion capture-driven simulations that hit and react, in: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 89–96.

[84] M. Da Silva, Y. Abe, J. Popović, Simulation of human motion data using short-horizon model-predictive control, in: Computer Graphics Forum, volume 27, Wiley Online Library, pp. 371–380.

[85] Y. Lee, S. Kim, J. Lee, Data-driven biped control, in: ACM SIGGRAPH 2010 papers, 2010, pp. 1–8.

[86] S. Hong, D. Han, K. Cho, J. S. Shin, J. Noh, Physics-based full-body soccer motion control for dribbling and shooting, ACM Transactions on Graphics (TOG) 38 (2019) 1–12.

[87] J. M. Wang, S. R. Hamner, S. L. Delp, V. Koltun, Optimizing locomotion controllers using biologically-based actuators and objectives, ACM Transactions on Graphics (TOG) 31 (2012) 1–11.

[88] T. Geijtenbeek, M. Van De Panne, A. F. Van Der Stappen, Flexible muscle-based locomotion for bipedal creatures, ACM Transactions on Graphics (TOG) 32 (2013) 1–11.

[89] Y. Lee, M. S. Park, T. Kwon, J. Lee, Locomotion control for many-muscle humanoids, ACM Transactions on Graphics (TOG) 33 (2014) 1–11.

[90] D. A. McCrea, I. A. Rybak, Organization of mammalian locomotor rhythm and pattern generation, Brain research reviews 57 (2008) 134–146.

[91] H. Schroll, F. H. Hamker, Computational models of basal-ganglia pathway functions: focus on functional neuroanatomy, Frontiers in systems neuroscience 7 (2013) 122.

[92] A. D. Kuo, Stabilization of lateral motion in passive dynamic walking, The International journal of robotics research 18 (1999) 917–930.

[93] G. Obinata, K. Hase, A. Nakayama, Controller design of musculoskeletal model for simulating bipedal walking, in: Annual Conference of the International FES Society, volume 2, p. 1.

[94] J. T. Choi, A. J. Bastian, Adaptation reveals independent control networks for human walking, Nature neuroscience 10 (2007) 1055–1062.

[95] G. Torres-Oviedo, A. J. Bastian, Natural error patterns enable transfer of motor learning to novel contexts, Journal of neurophysiology 107 (2012) 346–356.

[96] I. Cajigas, A. Koenig, G. Severini, M. Smith, P. Bonato, Robot-induced perturbations of human walking reveal a selective generation of motor adaptation, Sci Robot 2 (2017) 1–10.

[97] J. Zhang, P. Fiers, K. A. Witte, R. W. Jackson, K. L. Poggensee, C. G. Atkeson, S. H. Collins, Human-in-the-loop optimization of exoskeleton assistance during walking, Science 356 (2017) 1280–1284.

[98] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[99] S. Levine, Reinforcement learning and control as probabilistic inference: Tutorial and review, arXiv preprint arXiv:1805.00909 (2018).

[100] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533.

[101] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, nature 529 (2016) 484.

[102] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of go without human knowledge, Nature 550 (2017) 354–359.

[103] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: Advances in neural information processing systems, pp. 1057–1063.

[104] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in: International conference on machine learning, pp. 1889–1897.

[105] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[106] S. Kakade, J. Langford, Approximately optimal approximate reinforcement learning, in: ICML, volume 2, pp. 267–274.

[107] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971 (2015).

[108] S. Fujimoto, H. Van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, arXiv preprint arXiv:1802.09477 (2018).

[109] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, arXiv preprint arXiv:1801.01290 (2018).

[110] Openai spinning up documentation, https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#id20, ???? Accessed: 2020-04-29.

[111] M. P. Deisenroth, G. Neumann, J. Peters, et al., A survey on policy search for robotics, Foundations and Trends® in Robotics 2 (2013) 1–142.

[112] N. Hansen, The cma evolution strategy: a comparing review, in: Towards a new evolutionary computation, Springer, 2006, pp. 75–102.

[113] J. M. Wang, D. J. Fleet, A. Hertzmann, Optimizing walking controllers for uncertain inputs and environments, ACM Transactions on Graphics (TOG) 29 (2010) 1–8.

[114] X. B. Peng, G. Berseth, M. Van de Panne, Terrain-adaptive locomotion skills using deep reinforcement learning, ACM Transactions on Graphics (TOG) 35 (2016) 1–12.

[115] W. Yu, G. Turk, C. K. Liu, Learning symmetric and low-energy locomotion, ACM Transactions on Graphics (TOG) 37 (2018) 1–12.

[116] J. Won, J. Park, J. Lee, Aerobatics control of flying creatures via self-regulated learning, ACM Transactions on Graphics (TOG) 37 (2018) 1–10.

[117] L. Liu, J. Hodgins, Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning, ACM Transactions on Graphics (TOG) 37 (2018) 1–14.

[118] A. Clegg, W. Yu, J. Tan, C. K. Liu, G. Turk, Learning to dress: Synthesizing human dressing motion via deep reinforcement learning, ACM Transactions on Graphics (TOG) 37 (2018) 1–10.

[119] X. B. Peng, G. Berseth, K. Yin, M. Van De Panne, Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning, ACM Transactions on Graphics (TOG) 36 (2017) 1–13.

[120] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami, M. Riedmiller, Silver, David, Emergence of locomotion behaviours in rich environments, arXiv preprint arXiv:1707.02286 (2017).

[121] X. B. Peng, M. van de Panne, Learning locomotion skills using deeprl: Does the choice of action space matter?, in: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 1–13.

[122] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, arXiv preprint arXiv:1606.01540 (2016).

[123] Openai spinning up, https://anaconda.org/kidzik/opensim, ???? Accessed: 2020-04-29.

[124] Ł. Kidziński, S. P. Mohanty, C. F. Ong, Z. Huang, S. Zhou, A. Pechenko, A. Stelmaszczyk, P. Jarosik, M. Pavlov, S. Kolesnikov, et al., Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments, in: The NIPS'17 Competition: Building Intelligent Systems, Springer, 2018, pp. 121–153.

[125] Ł. Kidziński, C. Ong, S. P. Mohanty, J. Hicks, S. Carroll, B. Zhou, H. Zeng, F. Wang, R. Lian, H. Tian, et al., Artificial intelligence for prosthetics: Challenge solutions, in: The NeurIPS'18 Competition, Springer, 2020, pp. 69–128.

[126] osim-rl, https://github.com/stanfordnmbl/osim-rl/tree/v3.0.11, ????. Accessed: 2020-04-29.

[127] Aicrowd — challenges - neurips 2019: Learn to move, https://www.aicrowd.com/challenges/neurips-2019-learning-to-move-walk-around, ????. Accessed: 2020-04-29.

[128] osim-rl, http://osim-rl.stanford.edu/, ????. Accessed: 2020-04-29.

[129] M. H. Schwartz, A. Rozumalski, J. P. Trost, The effect of walking speed on the gait of typically developing children, Journal of biomechanics 41 (2008) 1639–1650.

[130] S. R. Hamner, S. L. Delp, Muscle contributions to fore-aft and vertical body mass center accelerations over a range of running speeds, Journal of biomechanics 46 (2013) 780–787.

[131] C. T. John, F. C. Anderson, J. S. Higginson, S. L. Delp, Stabilisation of walking by intrinsic muscle properties revealed in a three-dimensional muscle-driven simulation, Computer methods in biomechanics and biomedical engineering 16 (2013) 451–462.

[132] B. Zhou, H. Zeng, F. Wang, Y. Li, H. Tian, Efficient and robust reinforcement learning with uncertainty-based value expansion, arXiv preprint arXiv:1912.05328 (2019).

[133] S. Kolesnikov, O. Hrinchuk, Sample efficient ensemble learning with catalyst.rl (2019).

[134] D. Akimov, Distributed soft actor-critic with multivariate reward representation and knowledge distillation, arXiv preprint arXiv:1911.13056 (2019).

[135] osim-rl: environment, http://osim-rl.stanford.edu/docs/nips2019/environment/, ???? Accessed: 2020-04-29.

[136] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th annual international conference on machine learning, pp. 41–48.

[137] K. Hase, R. Stein, Turning strategies during human walking, Journal of Neurophysiology 81 (1999) 2914–2922.

[138] M. J. D. Taylor, P. Dabnichki, S. Strike, A three-dimensional biomechanical comparison between turning strategies during the stance phase of walking, Human movement science 24 (2005) 558–573.

[139] A. Hallemans, D. De Clercq, P. Aerts, Changes in 3d joint dynamics during the first 5 months after the onset of independent walking: a longitudinal follow-up study, Gait & posture 24 (2006) 270–279.

[140] K. E. Adolph, W. G. Cole, M. Komati, J. S. Garciaguirre, D. Badaly, J. M. Lingeman, G. L. Chan, R. B. Sotsky, How do you learn to walk? thousands of steps and dozens of falls per day, Psychological science 23 (2012) 1387–1394.

[141] N. Heess, G. Wayne, Y. Tassa, T. Lillicrap, M. Riedmiller, D. Silver, Learning and transfer of modulated locomotor controllers, arXiv preprint arXiv:1610.05182 (2016).

[142] K. Frans, J. Ho, X. Chen, P. Abbeel, J. Schulman, Meta learning shared hierarchies, arXiv preprint arXiv:1710.09767 (2017).

[143] J. Merel, M. Botvinick, G. Wayne, Hierarchical motor control in mammals and machines, Nature Communications 10 (2019) 1–12.

[144] J.-C. Latombe, Robot motion planning, volume 124, Springer Science & Business Media, 2012.

[145] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, E. Frazzoli, A survey of motion planning and control techniques for self-driving urban vehicles, IEEE Transactions on intelligent vehicles 1 (2016) 33–55.

[146] World chase tag^{TM} - home, https://www.worldchasetag.com/, ???? Accessed: 2020-02-16.

[147] M. Nakada, T. Zhou, H. Chen, T. Weiss, D. Terzopoulos, Deep learning of biomimetic sensorimotor control for biomechanical human animation, ACM Transactions on Graphics (TOG) 37 (2018) 1–15.

[148] K. P. Körding, D. M. Wolpert, Bayesian integration in sensorimotor learning, Nature 427 (2004) 244–247.

[149] M. A. Smith, A. Ghazizadeh, R. Shadmehr, Interacting adaptive processes with different timescales underlie short-term motor learning, PLoS biology 4 (2006).

[150] M. Günther, D. F. Haeufle, S. Schmitt, The basic mechanical structure of the skeletal muscle machinery: One model for linking microscopic and macroscopic scales, Journal of theoretical biology 456 (2018) 137–167.

[151] M. Millard, A. L. Emonds, M. Harant, K. Mombaur, A reduced muscle model and planar musculoskeletal model fit for the simulation of whole-body movements, Journal of biomechanics 89 (2019) 11–20.

[152] K. E. Zelik, A. D. Kuo, Human walking isn't all hard work: evidence of soft tissue contributions to energy dissipation and return, Journal of Experimental Biology 213 (2010) 4257–4264.

[153] K. M. Moerman, D. Solav, D. Sengeh, H. Herr, Automated and data-driven computational design of patient-specific biomechanical interfaces (2016).

[154] L. Ma, D. Chablat, F. Bennis, W. Zhang, A new simple dynamic muscle fatigue model and its validation, International journal of industrial ergonomics 39 (2009) 211–220.

[155] A. Falisse, S. Van Rossom, I. Jonkers, F. De Groote, Emg-driven optimal estimation of subject-specific hill model muscle–tendon parameters of the knee joint actuators, IEEE Transactions on Biomedical Engineering 64 (2016) 2253–2262.

[156] M. Sreenivasa, C. J. G. Chamorro, D. Gonzalez-Alvarado, O. Rettig, S. I. Wolf, Patient-specific bone geometry and segment inertia from mri images for model-based analysis of pathological gait, Journal of biomechanics 49 (2016) 1918–1925.

[157] M. Wesseling, F. De Groote, C. Meyer, K. Corten, J.-P. Simon, K. Desloovere, I. Jonkers, Subject-specific musculoskeletal modelling in patients before and after total hip arthroplasty, Computer methods in biomechanics and biomedical engineering 19 (2016) 1683–1691.