

REVIEW

Open Access



# Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation

Seungmoon Song<sup>1\*</sup> , Łukasz Kidziński<sup>2</sup>, Xue Bin Peng<sup>3</sup>, Carmichael Ong<sup>2</sup>, Jennifer Hicks<sup>2</sup>, Sergey Levine<sup>3</sup>, Christopher G. Atkeson<sup>4</sup> and Scott L. Delp<sup>1,2</sup>

## Abstract

Modeling human motor control and predicting how humans will move in novel environments is a grand scientific challenge. 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 was an official competition at the NeurIPS conference from 2017 to 2019 and attracted over 1300 teams from around the world. Top teams adapted state-of-the-art deep reinforcement learning techniques and produced 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 could be used to accelerate research on assistive devices, rehabilitation treatments, and physical

\*Correspondence: smsong@stanford.edu

<sup>1</sup> Department of Mechanical Engineering, Stanford University, Stanford, CA, USA

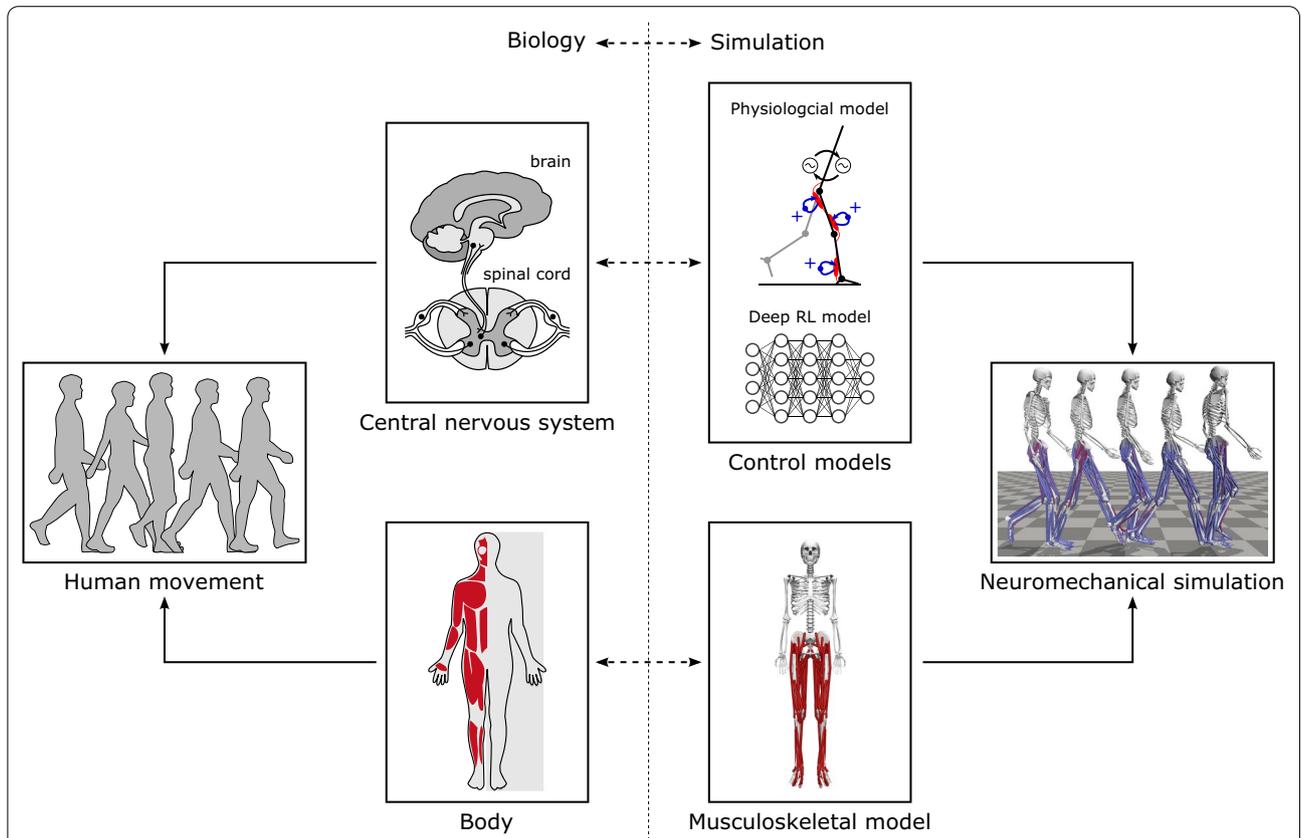
Full list of author information is available at the end of the article



© The Author(s) 2021. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

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 [3–5] 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 very 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 evaluate 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 steady walking, in a predictive manner [6–8]. 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 may allow us to overcome some of the limitations in current 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 controllers that can produce motions of interest, resulting in controllers that are often treated as a black-box due to their complexity. 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, a black-box controller that has been validated to produce human-like neuromechanical simulations could be useful in predicting responses to assistive devices or therapies like targeted strength-training. Or one could gain some insight about human motor control by training a controller with deep RL in



**Fig. 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

certain conditions (i.e., objective functions, simulation environment, etc.) and by analyzing the controller. One could also 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 from 2017 to 2019. 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 complex motions.

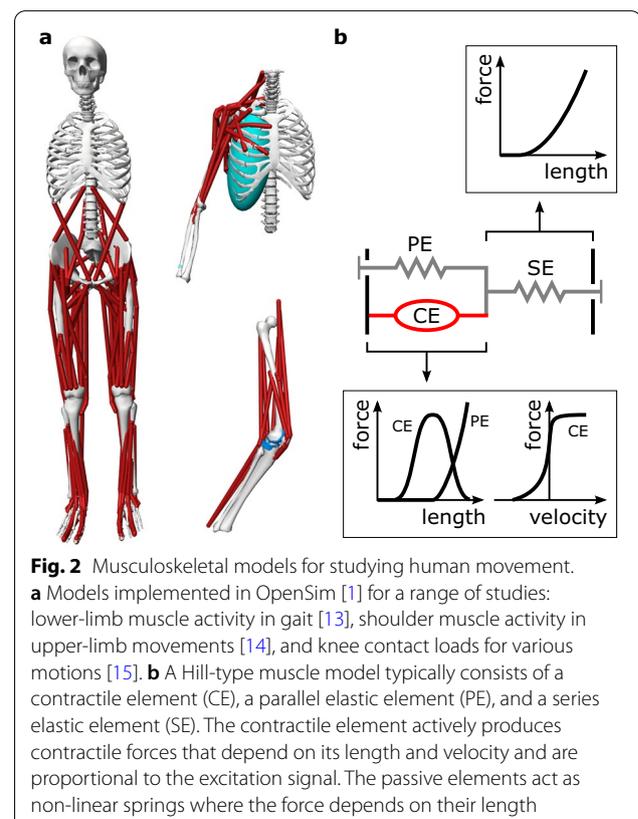
This paper reviews neuromechanical simulations and deep RL, with a focus on the materials relevant to modeling the control of human locomotion. First, we provide background on neuromechanical simulations of human locomotion and discuss how to evaluate their physiological plausibility. We also introduce 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. Our goal with this review is to provide a primer for researchers who want to apply deep RL approaches to study control of human movement in neuromechanical simulation and to demonstrate how deep RL is a powerful complement to traditional physiologically plausible control models.

## Background on neuromechanical simulations of human locomotion

This section provides background on neuromechanical simulations 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 this section 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 [14, 16, 17] (Fig. 2a). The skeletal system is often modeled by rigid segments connected by rotational joints. Hill-type muscle models [18] are commonly used to actuate the joints, capturing the dynamics of biological muscles, including both active and passive contractile elements [19–22] (Fig. 2b). Hill-type muscle models can be used with models of metabolic energy consumption [23–25] and muscle fatigue [26–28] 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 [29–32] and can be customized to match an individual's height, weight, or CT and MRI scan data [33, 34]. OpenSim [1], which is the basis of the OpenSim-RL package [35] used in the Learn to Move competition, is an open-source software package broadly used in the biomechanics community (e.g., it has about 60,000 unique user downloads as of 2021 [36]) to simulate musculoskeletal dynamics.

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, while achieving defined objectives, like minimizing muscle effort [37–39]. The resulting simulation estimates body states, such as individual muscle forces, that are difficult to directly measure with an experiment. Such an approach has been validated for human walking and running by comparing the simulated muscle activations with recorded electromyography data [40, 41], and for animal locomotion by comparing simulated muscle forces, activation levels, and muscle-tendon length changes with *in vivo* measurements during cat locomotion [42]. These motion tracking approaches have been used to analyze human locomotion [37, 39], to estimate body state in real-time to control assistive devices [43, 44], and to predict effects of exoskeleton assistance and surgical interventions on muscle coordination [45, 46]. While these simulations that track reference data provide a powerful tool to analyze recorded motions, they do not produce new motions and thus cannot predict movement in novel scenarios.

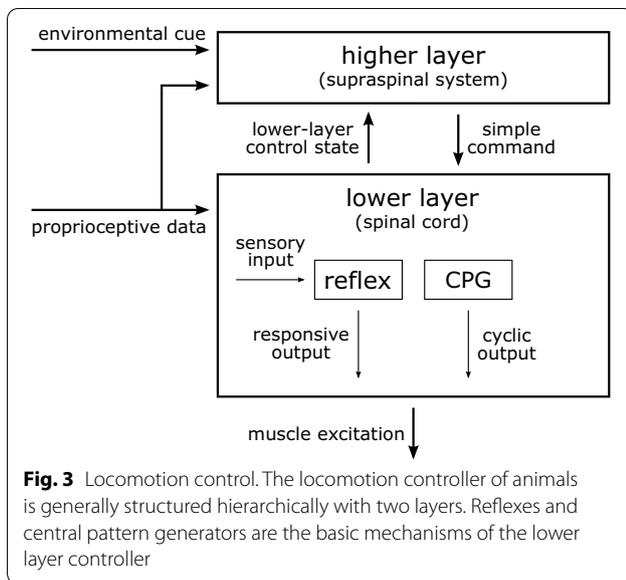
Alternatively, musculoskeletal simulations can produce motions without reference motion data using trajectory optimization methods [47]. This approach finds muscle activation patterns that produce a target motion through trajectory optimization with a musculoskeletal model based on an assumption that the target motion is well optimized for a specific objective. Thus, this approach has been successful in producing well-practiced motor tasks, such as normal walking and running [48, 49] and provides insights into the optimal gaits for different objectives [26, 27], biomechanical features [50], and assistive devices [51]. However, it is not straightforward to apply this approach to behaviors that are not well trained and thus functionally suboptimal. For instance, people initially walk inefficiently in lower limb exoskeletons and adapt to more energy optimal gaits over days and weeks [52]; therefore, trajectory optimization with an objective such as energy minimization likely would not predict the initial gaits. These functionally suboptimal behaviors

in humans are produced by the nervous system that is probably optimized for typical motions, such as normal walking, and is also limited by physiological control constraints, such as neural transmission delays and limited sensory information. A better representation of the underlying controller may be necessary to predict these kinds of emergent behaviors that depart from typical minimum effort optimal behaviors.

### 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). Neuromechanical simulations have been used to test gait assistive devices before developing hardware [53, 54] and to understand how changes in musculoskeletal properties affect walking performance [7, 28]. Moreover, the control model implemented in neuromechanical simulations can be directly used to control bipedal robots [55–57] and assistive devices [53, 58, 59].

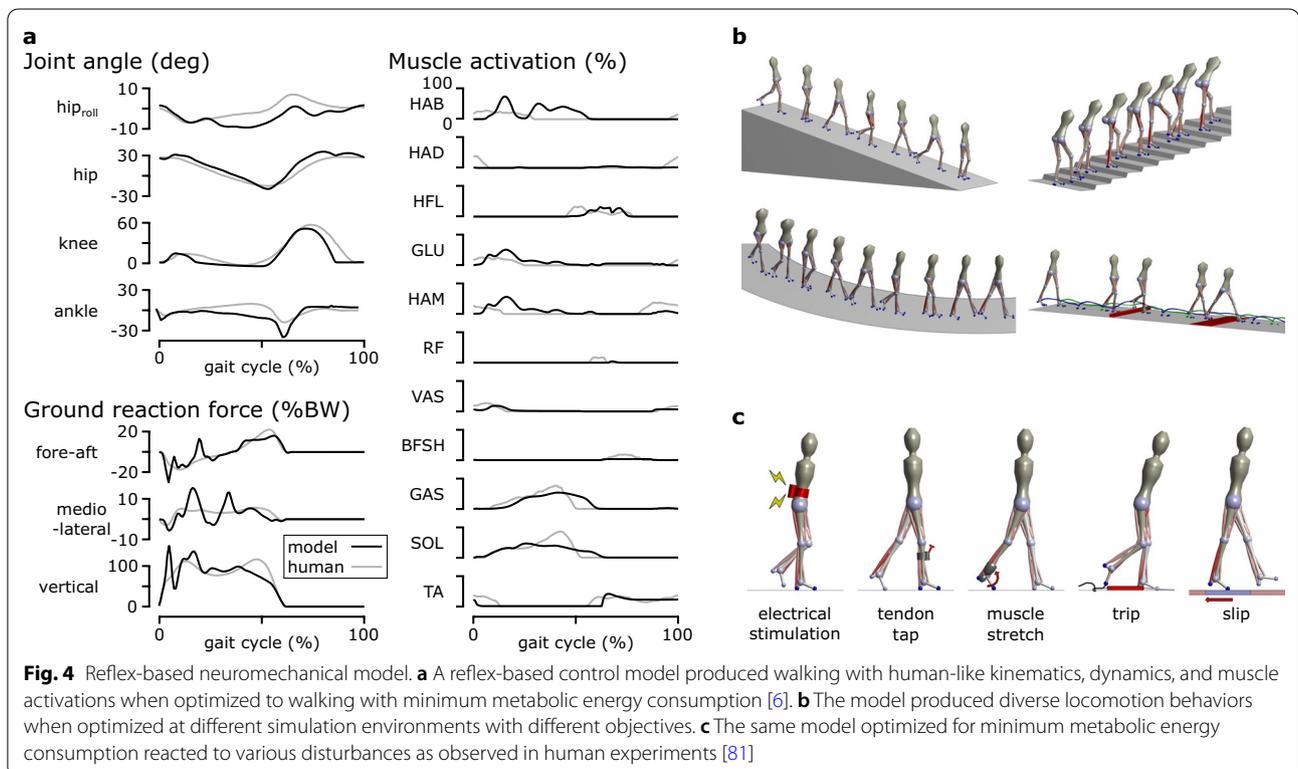
Modeling human motor control is crucial for a predictive neuromechanical simulation. However, most of our current understanding of human locomotion control is extrapolated from experimental studies of simpler animals [60, 61] as it is extremely difficult to measure and interpret the biological neural circuits. 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 [60]. 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 stereotypical locomotion behaviors and can be modulated by electrical stimulation to change speed, direction and gait [62, 63]. Second, the lower layer seems to consist of two control mechanisms: reflexes [64, 65] and central pattern generators (CPGs) [66, 67]. In engineering terms, reflexes and CPGs roughly correspond to feedback and feedforward control, respectively. Muscle synergies, where a single pathway co-activates multiple muscles, have also been proposed as a lower layer control mechanism that reduces the degrees of freedom for complex control tasks [68, 69]. Lastly, there is a consensus that humans use minimum effort to conduct well-practiced motor tasks, such as walking [70, 71]. This consensus provides a basis



for using energy or fatigue optimization [26–28] 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

[72], generate the basic muscle excitation patterns. These CPG-based models [8, 73–77] demonstrated that stable locomotion can emerge from the entrainment between CPGs and the musculoskeletal system, which are linked by sensory feedback and joint actuation. A CPG-based model that consists of 125 control parameters produced walking and running with a 3D musculoskeletal model with 60 muscles to walk and run [75]. CPG-based models also have been integrated with different control mechanisms, such as muscle synergies [8, 76, 77] and various sensory feedback circuits [74, 76]. 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 [6, 20, 78–80] 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 [61, 65]. A reflex-based control model with 80 control parameters combined with a simple higher layer controller that regulates foot placement to maintain balance produced diverse locomotion behaviors with a 3D musculoskeletal model with 22 muscles, including walking, running, and climbing stairs and slopes [6] and reacted to a range of unexpected perturbations similarly to humans [81] (Fig. 4). Reflex-based controllers also have been combined with CPGs [79] and a



deep neural network that operates as a higher layer controller [80] for more control functions, such as speed and terrain adaptation.

#### Human locomotion simulations for computer graphics

A number of controllers have been developed in computer graphics to automate the process of generating human-like locomotion for computer characters [82–86]. A variety of techniques have been proposed for simulating common behaviors, such as walking and running [87–90]. Reference motions, such as motion capture data, were often used in the development process to produce more natural behaviors [91–94]. Musculoskeletal models also have been used to achieve naturalistic motions [95–97], which makes them very close to neuromechanical simulations. The focus of these studies is producing natural-looking 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

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 [61, 98] 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 [99]. 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 [81] and its adaptations in new conditions, such as musculoskeletal changes [7, 28], external assistance [53, 54], and different terrains [6].

We can also assess the plausibility of control features that are encoded in a model. It is plausible for a control model to use sensory data that are known to be used in human locomotion [61, 65] 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 [26–28].

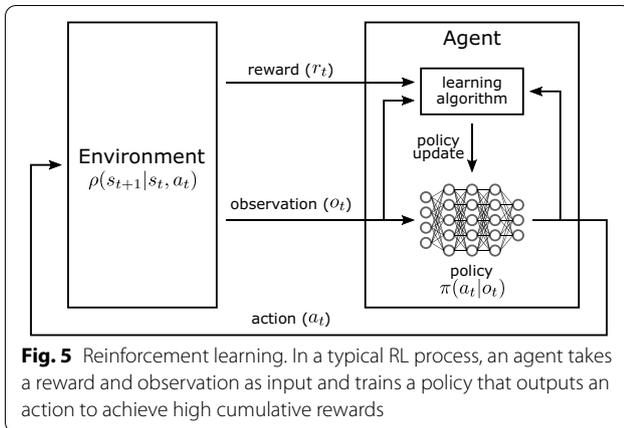
Existing neuromechanical control models are mostly limited to modeling the lower layer control and producing steady locomotion behaviors. Most aspects of the motor learning process and the higher layer control are thus missing in current neuromechanical models. Motor learning occurs in daily life when 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, wearing an exoskeleton [52, 100], and in experimentally constructed environments such as on a split-belt treadmill [101, 102] and with perturbation forces [103, 104]. 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 Future directions 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-the-art deep RL algorithms that are relevant to successful solutions in the Learn to Move competition. We also review studies that used deep RL to control human locomotion in physics-based simulation.

#### Deep reinforcement learning

Reinforcement learning is a machine learning paradigm for solving decision-making problems. The objective is to learn an optimal policy  $\pi$  that enables an agent to maximize its cumulative reward through interactions with its environment [105] (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 consist of a policy that controls the musculoskeletal model and a learning algorithm 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  $\pi$  for an action  $a_t$  (excitation values of the muscles in the model) in response to that observation. An 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



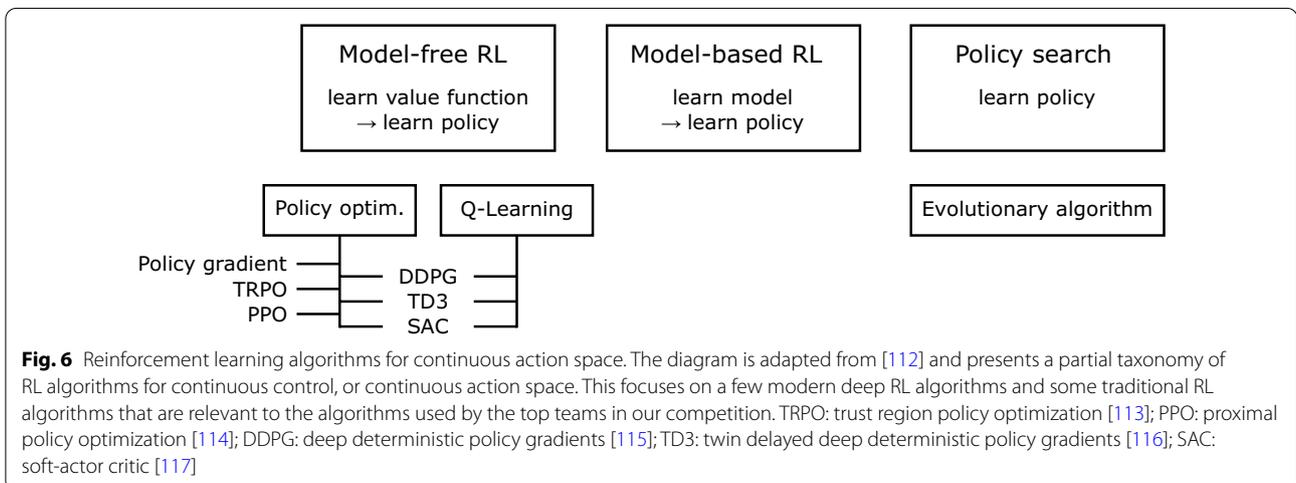
policy defines a distribution over actions  $a_t$  given a particular observation  $o_t$  [106]. Stochastic policies allow gradients to be computed for non-differentiable objective functions [107], such as those computed from the results of a neuromechanical simulation, and the gradients can be used to update the policies using gradient ascent. 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.

One of the crucial design decisions in applying RL to a particular problem is the choice of policy representation. Deep RL is the combination of RL with deep neural network function approximators. While a policy can be modeled by any class of functions that maps observations to actions, the 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 [108], overtook world champion human players in the game of Go [109, 110], and reached the highest league in a popular professional computer game that requires long-term strategies [111].

**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, or reward. 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 forward simulations of the policy, and then updating the policy via gradient ascent [118]. 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. Algorithms such as TRPO [113] and PPO [114] 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 [119]. Another limitation of policy gradient methods is their low sample efficiency. Standard policy gradient algorithms use a new batch of data collected with the current policy to estimate a gradient when updating the current policy at each



iteration. Thus, each batch of data is used to perform a small number of updates, then discarded, and millions of samples are often required to solve relatively simple tasks. Off-policy gradient algorithms can substantially reduce the number of samples required to learn effective policies by allowing the agent to reuse data collected from previous iterations of the algorithm when updating the latest policy [115–117]. Off-policy algorithms, such as DDPG [115], 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. More recent off-policy methods, such as TD3 and SAC, build on this approach and propose several modifications that further improve sample efficiency and stability.

### 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 [6, 75] and computer graphics studies [95, 96] reviewed in the Background on neuromechanical simulations of human locomotion section used policy search methods [120] 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 [121]. This optimization approach makes very minimal assumptions about the underlying system and can be effective for tuning controllers to perform a diverse array of skills [6, 122]. However, these algorithms often struggle with high dimensional parameter spaces (i.e., more than a couple of hundred parameters) [123]. Therefore, researchers developed controllers with a relatively low-dimensional set of parameters that could produce desired motions, which require a great deal of expertise and human insight. Also, the selected set of parameters tend to be specific for particular skills, limiting the behaviors that can be reproduced by the character.

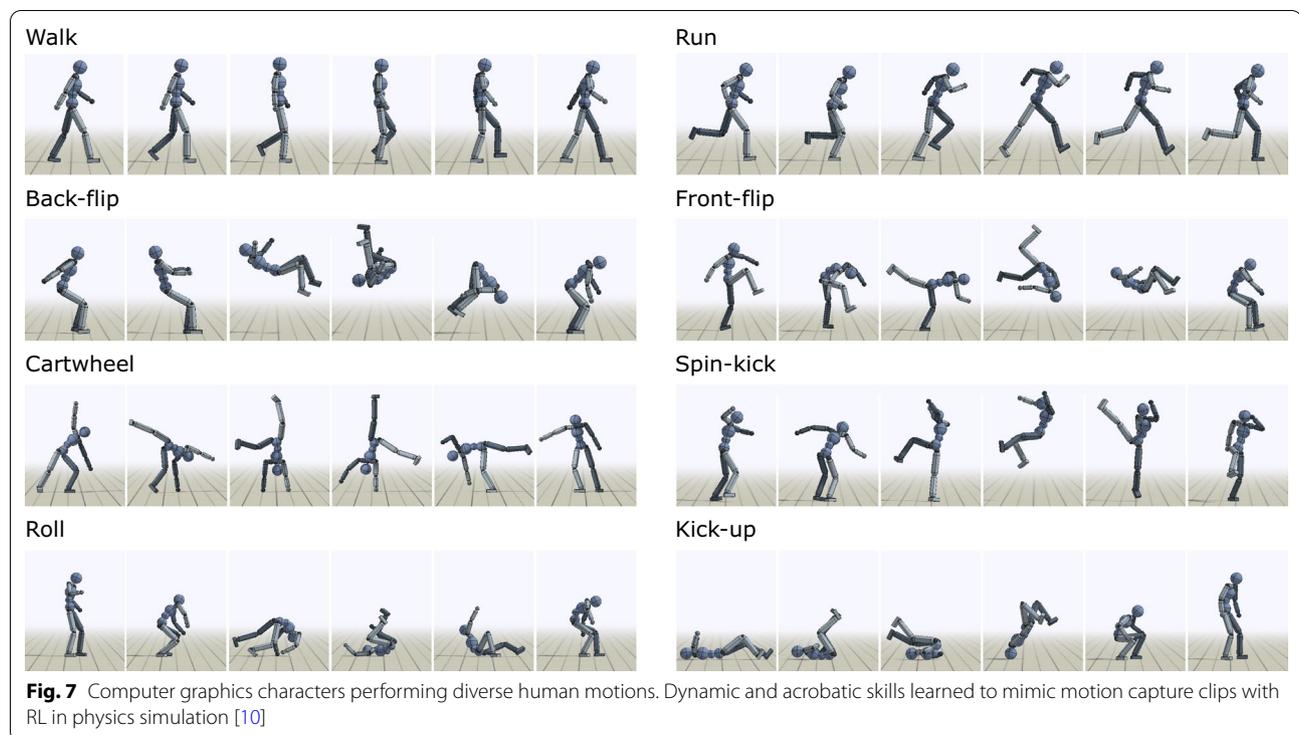
Recently, deep RL techniques have demonstrated promising results for character animation, with policy optimization methods emerging as the algorithms of choice for many of these applications [114, 115, 118]. These methods have been effective for training controllers that can perform a rich repertoire of skills [10, 124–127]. 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 [124, 128, 129], as well as for controlling complex musculoskeletal models [11, 130]. 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, 11, 126] (Fig. 7). As these studies using reference motion data show the potential of using deep RL methods in developing versatile controllers, it would be worth testing various deep RL approaches in neuromechanical simulations.

### 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 [35] leverages OpenSim to simulate musculoskeletal models and OpenAI Gym, a widely used RL toolkit [131], to standardize the interface with state-of-the-art RL algorithms. OpenSim-RL is open-source and is provided as a Conda package [132], which has been downloaded about 42,000 times from 2017 to 2019. 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 was held annually from 2017 to 2019. It was 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 [35, 133], 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 [134], 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 Top solutions and results 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 (Appendix - Reward function).

### NeurIPS 2019: Learn to Move - Walk Around

#### Overview

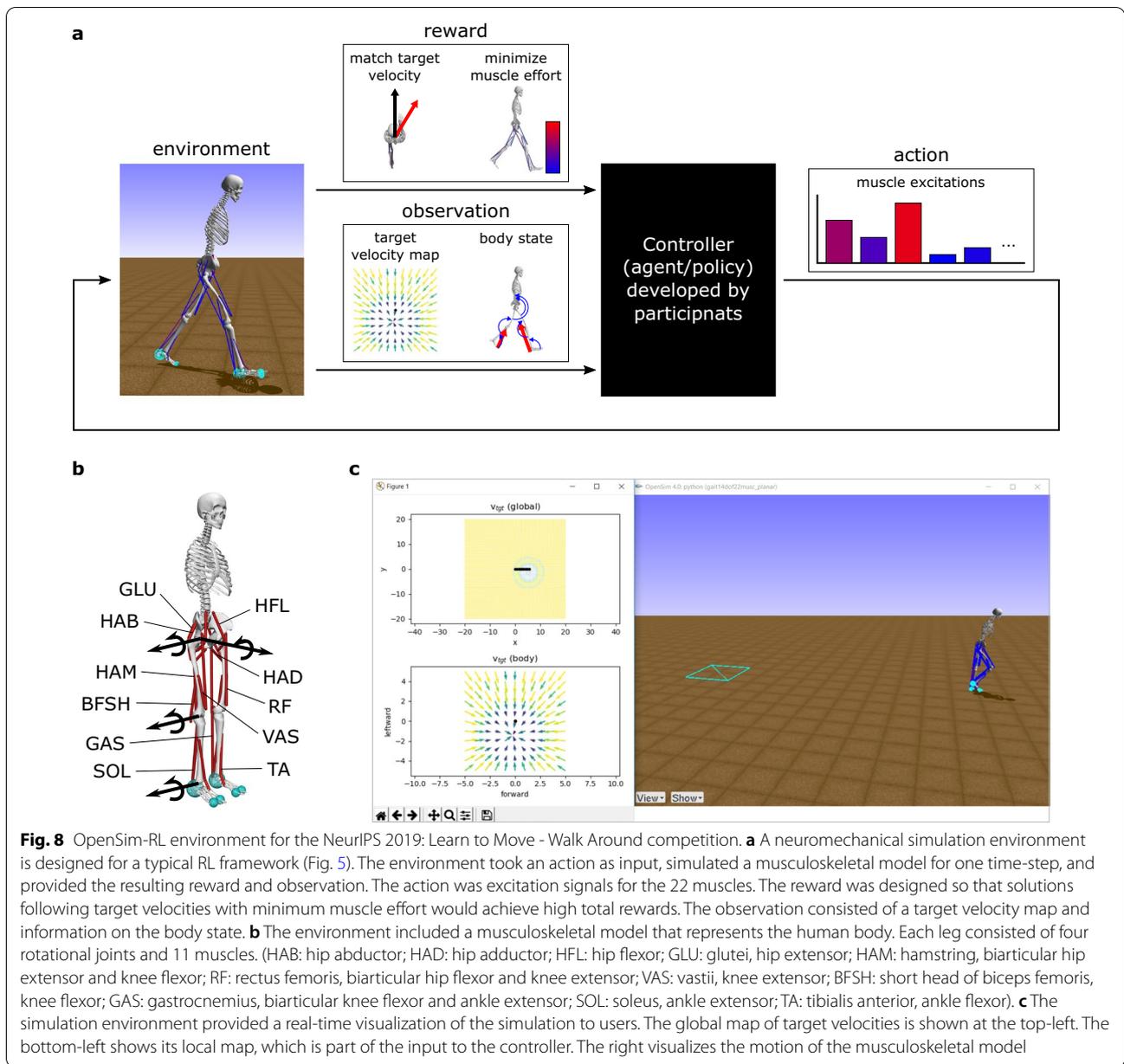
NeurIPS 2019: Learn to Move - Walk Around was held online from June 6 to November 29 in 2019. The task was to develop a locomotion controller, which was scored based on its ability to meet target velocity vectors when applied in the provided OpenSim-RL simulation environment. The environment repository was shared on Github [135], the submission and grading were managed using the AICrowd platform [136], and the project homepage provided documentation on the environment and the competition [137]. 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 [138–140] 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. More details on the competition can be found on the competition homepage [136].

In total, 323 teams participated in the competition and submitted 1448 solutions. In Round 2, the top three teams [141–143] 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 [141] along with two more finalist papers [142, 143]. 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 (Appendix – Reward function). The mean total reward of five trials with different target velocities was used for ranking.

**Top solutions and results**

All of the top three teams that succeeded in completing the task used deep reinforcement learning [141–143]. None of the teams utilized reference motion data for training or used domain knowledge in designing the

policy. The only part of the training process that was specific to locomotion was using intermediate rewards that induced effective gaits or facilitated the training process. The top teams used various heuristic RL techniques that have been effectively used since the first competition [133, 134] and adapted state-of-the-art deep RL training algorithms.

Various heuristic RL techniques were used, 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 [142]. 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 [142, 143]. Once agents found solutions that seem to achieve intended behaviors with these techniques, they typically were further tuned with the original problem formulation.

Curriculum learning [144] 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. Zhou et al. [141] 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.

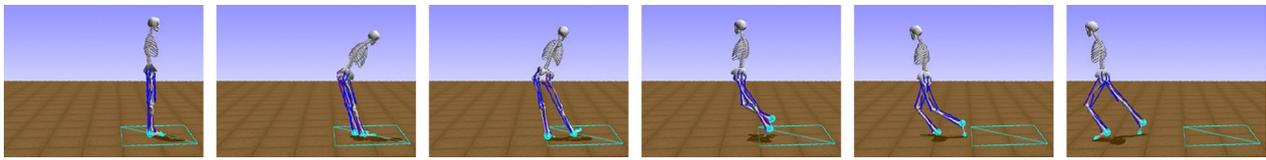
All of the top teams used off-policy deep RL algorithms. The first place entry by Zhou et al. [141] used DDPG [115], the second place entry by Kolesnikov and Hrinchuk [142] used TD3 [116], and the third place entry by Akimov [143] used SAC [117]. Since off-policy algorithms allow updating the policy using data collected in previous iterations, 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 [142] 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.

The winning team, Zhou et al., proposed risk averse value expansion (RAVE), a hybrid approach of model-based and model-free RL [141]. 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 [131] compared to state-of-the-art algorithms [141].

#### **Implications for human locomotion control**

The top solution 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. 8a). 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. 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



**Fig. 9** Rapid turning motion. The top solution can make the musculoskeletal model with no internal yaw degree of freedom to turn  $180^\circ$  in a single step. Snapshots were taken every 0.4 s

how the simulated motion compares with human turning [145, 146].

The top solutions had some limitations in producing human-like motions. In the top solution [141], 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, 49]. 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 for specific motions. The second and third solutions [142, 143] were further from human motions as they gradually moved towards a target often using side steps. As policy gradient methods use gradient ascent, they often get stuck at local optima resulting in suboptimal motions [129] 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 [10–12] or by leveraging control models that produce natural gaits [6, 20]. Different walking gaits, some of which are possibly suboptimal, are also observed in toddlers during the few months of extensive walking experience [147, 148], and interpreting this process with an RL framework will be instructive to understanding human motor learning.

### Future directions

Deep reinforcement learning could be a powerful tool in developing neuromechanical control models. The best solutions of the Learn to Move competition, which used deep RL without reference motion data, produced rapid turning and walk-to-stand motions that had not previously been demonstrated with physiologically plausible models. However, it is difficult to fully optimize a deep neural network, suggesting that it is very challenging to train a single network that can produce a wide range of human-like motions. Moreover, only the top three teams in the competition were able to conduct the task of following target velocities [141–143], and such brute force deep RL may not easily extend for tasks that require long-term motion planning, such as navigating in a dynamic and complex environment.

Various deep reinforcement learning approaches, such as imitation learning and hierarchical learning, could be used to produce more optimized and complex motions. Humans can perform motions that are much more challenging than just walking at various speeds and directions. 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, 11, 126] could be used to train multiple separate networks to master a set of acrobatic skills (Fig. 7). These networks of motion primitives can then be part of the lower layer of a hierarchical controller [149–151], where a higher-layer network could be trained to coordinate the motion skills. A physiologically plausible control models that produces human-like walking, for instance, can also be part of the lower layer. More control layers that analyze dynamic scenes and plan longer-term motion sequences [152, 153] can be added if a complex strategy is required for the task. We will design future competitions to promote research in these directions of performing motions that would be difficult to produce with brute force deep RL. 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 [154].

Deep RL could also help to advance our understanding of human motor control. First, RL environments and solutions could have implications for human movement. While interpreting individual connections and weights of general artificial neural networks in terms of biological control circuits may not be plausible, rewards and policies that generate realistic motions could inform us about the objectives and control structures that underlie human movement. Also, perturbation responses of the trained policies that signify sensory-motor connections could be used to further analyze the physiological plausibility of the policies by comparing the responses to those observed in human experiments [81, 155–157]. Second, we could use deep RL as a means of training a black-box controller that complements a physiologically plausible model in simulating motions of interest. For instance, one could test an ankle control model in the context of

walking if there is a black-box controller for the other joints that in concert produces walking. Third, we may be able to use data-driven deep RL, such as imitation learning, to train physiologically plausible control models. We could establish such a training framework by using existing plausible control models as baseline controllers to produce (simulated) training data, and then determining the size and scope of gait data needed to train policies that capture the core features of the baseline controllers. Once the framework is established and validated with these existing control models, we could train new policies using human motion data. These control models could better represent human motor control than the ones that have been developed through imitation RL with only target motions as reference data. Also, these models could produce reliable predictions and could be customized to individuals.

While this paper focuses on the potential synergy of neuromechanical simulations and deep reinforcement learning, combining a broader range of knowledge, models, and methodologies will be vital in further understanding and modeling human motor control. For instance, regarding motor learning, there are a number of hypotheses and models of the signals that drive learning [102, 158], the dynamics of the adaptation process [103, 159], and the mechanisms of constructing and adapting movements [160–163]. Most of these learning models seek to capture the net behavioral effects, where a body motion is often represented by abstract features; implementing these learning models together with motion control models (such as those discussed in this paper) could provide a holistic evaluation of both motor control and learning models [164–166]. There are also different types of human locomotion models, including simple dynamic models and data-driven mathematical models. These models have provided great insights into the dynamic principles of walking and running [167–170], the stability and optimality of steady and non-steady gaits [171–177], and the control and adaptation of legged locomotion [166, 178–181]. As these models often account for representative characteristics, such as the center of mass movement and foot placement, they could be used in modeling the higher layer of hierarchical controllers.

## 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. Despite some success of using controllers based on deep RL to produce coordinated body motions in physics-based simulations, producing more complex

motions involving long-term planning and learning physiologically plausible models remain as future research challenges. Progress in these directions might be accelerated by combining domain expertise in modeling human motor control and advanced machine learning techniques. We hope to see more interdisciplinary studies and collaborations that are able to explain 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.

## Appendix—Reward function

The reward function, in the NeurIPS 2019: Learn to Move competition, was designed based on previous neuromechanical simulation studies [6, 28] that produced human-like walking. The total reward,  $J(\pi)$ , consisted of three reward terms:

$$\begin{aligned} J(\pi) &= R_{alive} + R_{step} + R_{target} \\ &= \sum_{i_{sim}} b_{alive} + \sum_{i_{step}} (w_s b_s - w_v c_v - w_e c_e) + \sum_{i_{target}} b_{target} \end{aligned} \quad (1)$$

where  $R_{alive}$  (with  $b_{alive} = 0.1$ ) was for not falling down,  $R_{step}$  was for making footsteps with desired velocities and small effort, and  $R_{target}$  was for reaching target locations. The indexes  $i_{sim} \in \{1, 2, \dots, 2500\}$ ,  $i_{step} \in \{1, 2, \dots\}$  and  $i_{target} \in \{1, 2\}$  were for the simulation step, footstep, and target location, respectively. The step reward,  $R_{step}$  consists of one bonus term and two cost terms. The step bonus,  $b_s = \Delta t_{i_{step}} = \sum_{i \text{ in } i_{step}} \Delta t$ , where  $\Delta t = 0.01$  s is the simulation time step, is weighted heavily with  $w_s$  to ensure the step reward is positive for every footstep. The velocity cost,  $c_v = \left\| \sum_{i \text{ in } i_{step}} (v_{pel} - v_{tgt0}) \Delta t \right\|_2$ , penalizes the deviation of average velocity during the footstep from the average of the target velocities given during that step. As the velocity cost is calculated with average velocity, it allows instantaneous velocity to naturally fluctuate within a footstep as in human walking [182]. The effort cost,  $c_e = \sum_{i \text{ in } i_{step}} \sum_m A_m^2 \Delta t$ , penalizes the use of muscles, where  $A_m$  is the activation level of muscle  $m \in \{HAB_L, HAD_L, \dots, TAR\}$ . The time integration of muscle activation square approximates muscle fatigue and is often minimized in locomotion simulations [26, 28]. The step bonus and costs are proportioned by the simulation time step so that the total reward does not favor many small footsteps over fewer large footsteps or vice versa. The weights were  $w_s = 10$ ,  $w_v = 3$ , and  $w_e = 1$ . Lastly, the target reward,  $R_{target}$ , with high bonuses of  $b_{target} = 500$  were to reward solutions that successfully follow target velocities. At the beginning of a simulation

trial, target velocities pointed toward the first target location ( $i_{target} = 1$ ), and if the human model reached the target location and stayed close to it ( $\leq 0.3$  m) for a while ( $2 \sim 4$  s),  $b_{target}$  was awarded. Then the target velocities were updated to point toward a new target location ( $i_{target} = 2$ ) with another bonus  $b_{target}$ . The hypothetical maximum total reward of a trial, with zero velocity and effort costs, is  $\max(R_{alive}) + \max(R_{step}) + \max(R_{target}) = 250 + 250 + 1000 = 1500$ .

### Abbreviations

CPG: Central pattern generators; RAVE: Risk averse value expansion; RL: Reinforcement learning.

### Acknowledgements

We thank our partners and sponsors: Alcrowd, 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.

### Authors' contributions

SS, ŁK, XBP, CO, JK, SL, CGA and SLD organized the Learn to Move competition. SS, ŁK, XBP, CO and JH drafted the manuscript. SS, ŁK, XBP, CO, JK, SL, CGA and SLD edited and approved the submitted manuscript. All authors read and approved the final 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 [135] and the project homepage [137].

### Declarations

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>Department of Mechanical Engineering, Stanford University, Stanford, CA, USA. <sup>2</sup>Department of Bioengineering, Stanford University, Stanford, CA, USA. <sup>3</sup>Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, CA, USA. <sup>4</sup>Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Received: 11 August 2020 Accepted: 29 July 2021

Published online: 16 August 2021

### References

- Seth A, Hicks JL, Uchida TK, Habib A, Dembia CL, Dunne JJ, Ong CF, DeMers MS, Rajagopal A, Millard M, et al. Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS Comput Biol*. 2018;14(7).
- Dembia CL, Bianco NA, Falisse A, Hicks JL, Delp SL. Opensim moco: Musculoskeletal optimal control. *PLoS Comput Biol*. 2019;16(12).
- Todorov E, Erez T, Mujoco TY. A physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012;5026–33.
- Lee J, Grey M, Ha S, Kunz T, Jain S, Ye Y, Srinivasa S, Stilman M, Liu C. Dart: Dynamic animation and robotics toolkit. *J Open Source Softw*. 2018;3(22):500.
- Hwangbo J, Lee J, Hutter M. Per-contact iteration method for solving contact dynamics. *IEEE Robot Autom Lett*. 2018;3(2):895–902.
- Song S, Geyer H. A neural circuitry that emphasizes spinal feedback generates diverse behaviours of human locomotion. *J Physiol*. 2015;593(16):3493–511.
- Ong CF, Geijtenbeek T, Hicks JL, Delp SL. Predicting gait adaptations due to ankle plantarflexor muscle weakness and contracture using physics-based musculoskeletal simulations. *PLoS Comput Biol*. 2019;15(10).
- Tamura D, Aoi S, Funato T, Fujiki S, Senda K, Tsuchiya K. Contribution of phase resetting to adaptive rhythm control in human walking based on the phase response curves of a neuromusculoskeletal model. *Front Neurosci*. 2020;14:17.
- Richards BA, Lillicrap TP, Beaudoin P, Bengio Y, Bogacz R, Christensen A, Clopath C, Costa RP, de Berker A, Ganguli S, et al. A deep learning framework for neuroscience. *Nature Neurosci*. 2019;22(11):1761–70.
- Peng XB, Abbeel P, Levine S, van de Panne M. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans Graph*. 2018;37(4):1–14.
- Lee S, Park M, Lee K, Lee J. Scalable muscle-actuated human simulation and control. *ACM Trans Graph*. 2019;38(4):1–13.
- Anand AS, Zhao G, Roth H, Seyfarth A. 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), 2019;537–43.
- Delp SL, Loan JP, Hoy MG, Zajac FE, Topp EL, Rosen JM. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Trans Biomed Eng*. 1990;37(8):757–67.
- Seth A, Dong M, Matias R, Delp SL. Muscle contributions to upper-extremity movement and work from a musculoskeletal model of the human shoulder. *Front Neurobot*. 2019;13:90.
- Lerner ZF, DeMers MS, Delp SL, Browning RC. How tibiofemoral alignment and contact locations affect predictions of medial and lateral tibiofemoral contact forces. *J Biomech*. 2015;48(4):644–50.
- Arnold EM, Ward SR, Lieber RL, Delp SL. A model of the lower limb for analysis of human movement. *Ann Biomed Eng*. 2010;38(2):269–79.
- Rajagopal A, Dembia CL, DeMers MS, Delp DD, Hicks JL, Delp SL. Full-body musculoskeletal model for muscle-driven simulation of human gait. *IEEE Trans Biomed Eng*. 2016;63(10):2068–79.
- Hill AV. The heat of shortening and the dynamic constants of muscle. *Proc R Soc London*. 1938;126(843):136–95.
- Zajac FE. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit Rev Biomed Eng*. 1989;17(4):359–411.
- Geyer H, Herr H. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Trans Neural Syst Rehab Eng*. 2010;18(3):263–73.
- Millard M, Uchida T, Seth A, Delp SL. Flexing computational muscle: modeling and simulation of musculotendon dynamics. *J Biomech Eng*. 2013;135:2.
- Haeufle D, Günther M, Bayer A, Schmitt S. Hill-type muscle model with serial damping and eccentric force-velocity relation. *J Biomech*. 2014;47(6):1531–6.
- Bhargava LJ, Pandy MG, Anderson FC. A phenomenological model for estimating metabolic energy consumption in muscle contraction. *J Biomech*. 2004;37(1):81–8.
- Umberger BR. Stance and swing phase costs in human walking. *J R Soc Interface*. 2010;7(50):1329–40.
- Koelewijn AD, Heinrich D, Van Den Bogert AJ. Metabolic cost calculations of gait using musculoskeletal energy models, a comparison study. *PLoS ONE*. 2019;14:9.
- Ackermann M, Van den Bogert AJ. Optimality principles for model-based prediction of human gait. *J Biomech*. 2010;43(6):1055–60.
- Miller RH, Umberger BR, Hamill J, Caldwell GE. Evaluation of the minimum energy hypothesis and other potential optimality criteria for human running. *Proc R Soc B*. 2012;279(1733):1498–505.

28. Song S, Geyer H. Predictive neuromechanical simulations indicate why walking performance declines with ageing. *J Physiol*. 2018;596(7):1199–210.
29. Chandler R, Clauser CE, McConville JT, Reynolds H, Young JW. Investigation of inertial properties of the human body. Air Force Aerospace Medical Research Lab Wright-Patterson AFB OH: Technical report; 1975.
30. Visser J, Hoogkamer J, Bobbert M, Huijing P. Length and moment arm of human leg muscles as a function of knee and hip-joint angles. *Eur J Appl Physiol Occup Physiol*. 1990;61(5–6):453–60.
31. Yamaguchi G. A survey of human musculotendon actuator parameters. Multiple muscle systems: Biomechanics and movement organization; 1990.
32. Ward SR, Eng CM, Smallwood LH, Lieber RL. Are current measurements of lower extremity muscle architecture accurate? *Clin Orthopaed Related Res*. 2009;467(4):1074–82.
33. Scheys L, Loeckx D, Spaepen A, Suetens P, Jonkers I. Atlas-based non-rigid image registration to automatically define line-of-action muscle models: a validation study. *J Biomech*. 2009;42(5):565–72.
34. Fregly BJ, Boninger ML, Reinkensmeyer DJ. Personalized neuromusculoskeletal modeling to improve treatment of mobility impairments: a perspective from european research sites. *J Neuroeng Rehabil*. 2012;9(1):18.
35. Kidziński Ł, Mohanty SP, Ong CF, Hicks JL, Carroll SF, Levine S, Salathé M, Delp SL. Learning to run challenge: Synthesizing physiologically accurate motion using deep reinforcement learning. In: *The NIPS'17 Competition: Building Intelligent Systems*, 2018;101–120. Springer.
36. SimTK: OpenSim. [https://simtk.org/plugins/reports/index.php?type=group&group\\_id=91&reports=reports](https://simtk.org/plugins/reports/index.php?type=group&group_id=91&reports=reports). Accessed 07 Aug 2021.
37. De Groote F, Van Campen A, Jonkers I, De Schutter J. Sensitivity of dynamic simulations of gait and dynamometer experiments to hill muscle model parameters of knee flexors and extensors. *J Biomech*. 2010;43(10):1876–83.
38. Thelen DG, Anderson FC, Delp SL. Generating dynamic simulations of movement using computed muscle control. *J Biomech*. 2003;36(3):321–8.
39. De Groote F, Kinney AL, Rao AV, Fregly BJ. Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. *Ann Biomed Eng*. 2016;44(10):2922–36.
40. Liu MQ, Anderson FC, Schwartz MH, Delp SL. Muscle contributions to support and progression over a range of walking speeds. *J Biomech*. 2008;41(15):3243–52.
41. Hamner SR, Seth A, Delp SL. Muscle contributions to propulsion and support during running. *J Biomech*. 2010;43(14):2709–16.
42. Karabulut D, Dogru SC, Lin Y-C, Pandey MG, Herzog W, Arslan YZ. Direct validation of model-predicted muscle forces in the cat hindlimb during locomotion. *J Biomech Eng*. 2020;142:5.
43. Cavallaro EE, Rosen J, Perry JC, Burns S. Real-time myoprocessors for a neural controlled powered exoskeleton arm. *IEEE Trans Biomed Eng*. 2006;53(11):2387–96.
44. Lotti N, Xiloyannis M, Durandau G, Galofaro E, Sanguineti V, Masia L, Sartori M. Adaptive model-based myoelectric control for a soft wearable arm exosuit: A new generation of wearable robot control. *IEEE Robotics & Automation Magazine*. 2020.
45. Uchida TK, Seth A, Pouya S, Dembia CL, Hicks JL, Delp SL. Simulating ideal assistive devices to reduce the metabolic cost of running. *PLoS ONE*. 2016;11:9.
46. Fox MD, Reinbolt JA, Öunpuu S, Delp SL. Mechanisms of improved knee flexion after rectus femoris transfer surgery. *J Biomech*. 2009;42(5):614–9.
47. De Groote F, Falisse A. Perspective on musculoskeletal modelling and predictive simulations of human movement to assess the neuromechanics of gait. *Proc R Soc B*. 2021;288(1946):20202432.
48. Anderson FC, Pandy MG. Dynamic optimization of human walking. *J Biomech Eng*. 2001;123(5):381–90.
49. Falisse A, Serrancolí G, Dembia CL, Gillis J, Jonkers I, De Groote F. Rapid predictive simulations with complex musculoskeletal models suggest that diverse healthy and pathological human gaits can emerge from similar control strategies. *J R Soc Interface*. 2019;16(157):20190402.
50. Miller RH, Umberger BR, Caldwell GE. Limitations to maximum sprinting speed imposed by muscle mechanical properties. *J Biomech*. 2012;45(6):1092–7.
51. Handford ML, Srinivasan M. Energy-optimal human walking with feedback-controlled robotic prostheses: a computational study. *IEEE Trans Neural Syst Rehabil Eng*. 2018;26(9):1773–82.
52. Zhang J, Fiers P, Witte KA, Jackson RW, Poggensee KL, Atkeson CG, Collins SH. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science*. 2017;356(6344):1280–4.
53. Thatté N, Geyer H. Toward balance recovery with leg prostheses using neuromuscular model control. *IEEE Trans Biomed Eng*. 2015;63(5):904–13.
54. Seo K, Hyung S, Choi, BK, Lee Y, Shim Y. A new adaptive frequency oscillator for gait assistance. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015;5565–71.
55. Batts Z, Song S, Geyer H. Toward a virtual neuromuscular control for robust walking in bipedal robots. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015;6318–23.
56. Van der Noot N, Ijspeert AJ, Ronsse R. Neuromuscular model achieving speed control and steering with a 3d bipedal walker. *Auton Robots*. 2019;43(6):1537–54.
57. Zhao G, Szymanski F, Seyfarth A. Bio-inspired neuromuscular reflex based hopping controller for a segmented robotic leg. *Bioinspir Biomimet*. 2020;15(2):026007.
58. Eilenberg MF, Geyer H, Herr H. Control of a powered ankle-foot prosthesis based on a neuromuscular model. *IEEE Trans Neural Syst Rehabil Eng*. 2010;18(2):164–73.
59. Wu AR, Dzeladini F, Brug TJ, Tamburella F, Tagliamonte NL, Van Asseldonk EH, Van Der Kooij H, Ijspeert AJ. An adaptive neuromuscular controller for assistive lower-limb exoskeletons: a preliminary study on subjects with spinal cord injury. *Front Neurobot*. 2017;1:130.
60. Orlovsky T, Orlovskii GN, Deliagina T, Grillner S. *Neuronal Control of Locomotion: from Mollusc to Man*. Oxford: Oxford University Press; 1999.
61. Capaday C. The special nature of human walking and its neural control. *TRENDS Neurosci*. 2002;25(7):370–6.
62. Armstrong DM. The supraspinal control of mammalian locomotion. *J Physiol*. 1988;405(1):1–37.
63. Sirota MG, Di Prisco GV, Dubuc R. Stimulation of the mesencephalic locomotor region elicits controlled swimming in semi-intact lampreys. *Eur J Neurosci*. 2000;12(11):4081–92.
64. Clarac F. Some historical reflections on the neural control of locomotion. *Brain Res Rev*. 2008;57(1):13–21.
65. Hultborn H. Spinal reflexes, mechanisms and concepts: from eccles to lundberg and beyond. *Progr Neurobiol*. 2006;78(3–5):215–32.
66. MacKay-Lyons M. Central pattern generation of locomotion: a review of the evidence. *Phys Ther*. 2002;82(1):69–83.
67. Minassian K, Hofstoetter US, Dzeladini F, Guertin PA, Ijspeert A. The human central pattern generator for locomotion: Does it exist and contribute to walking? *Neuroscientist*. 2017;23(6):649–63.
68. Lacquaniti F, Ivanenko YP, Zago M. Patterned control of human locomotion. *J Physiol*. 2012;590(10):2189–99.
69. Bizzi E, Cheung VC. The neural origin of muscle synergies. *Front Comput Neurosci*. 2013;7:51.
70. Ralston HJ. Energy-speed relation and optimal speed during level walking. *Internationale Zeitschrift für Angewandte Physiologie Einschliesslich Arbeitsphysiologie*. 1958;17(4):277–83.
71. Todorov E. Optimality principles in sensorimotor control. *Nat Neurosci*. 2004;7(9):907–15.
72. Matsuoka K. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biol Cybern*. 1985;52(6):367–76.
73. Taga G, Yamaguchi Y, Shimizu H. Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biol Cybern*. 1991;65(3):147–59.
74. Ogihara N, Yamazaki N. Generation of human bipedal locomotion by a bio-mimetic neuro-musculo-skeletal model. *Biol Cybern*. 2001;84(1):1–11.
75. Hase K, Miyashita K, Ok S, Arakawa Y. Human gait simulation with a neuromusculoskeletal model and evolutionary computation. *J Visualiz Comput Anim*. 2003;14(2):73–92.
76. Jo S, Massaquoi SG. A model of cerebrotocerebellum-spinomuscular interaction in the sagittal control of human walking. *Biol Cybern*. 2007;96(3):279–307.

77. Aoi S, Ohashi T, Bamba R, Fujiki S, Tamura D, Funato T, Senda K, Ivanenko Y, Tsuchiya K. Neuromusculoskeletal model that walks and runs across a speed range with a few motor control parameter changes based on the muscle synergy hypothesis. *Sci Rep*. 2019;9(1):1–13.
78. Günther M, Ruder H. Synthesis of two-dimensional human walking: a test of the  $\lambda$ -model. *Biol Cybern*. 2003;89(2):89–106.
79. Dzeladini F, Van Den Kieboom J, Ijspeert A. The contribution of a central pattern generator in a reflex-based neuromuscular model. *Front Human Neurosci*. 2014;8:371.
80. Wang J, Qin W, Sun L. Terrain adaptive walking of biped neuromuscular virtual human using deep reinforcement learning. *IEEE Access*. 2019;7:92465–75.
81. Song S, Geyer H. Evaluation of a neuromechanical walking control model using disturbance experiments. *Front Comput Neurosci*. 2017;11:15.
82. Faloutsos P, Van de Panne M, Terzopoulos D. Composable controllers for physics-based character animation. In: *Proceedings of the 28th ACM SIGGRAPH*, 2001;251–60.
83. Fang AC, Pollard NS. Efficient synthesis of physically valid human motion. *ACM Trans Graph*. 2003;22(3):417–26.
84. Wampler K, Popović Z. Optimal gait and form for animal locomotion. *ACM Trans Graph*. 2009;28(3):1–8.
85. Coros S, Karpathy A, Jones B, Reveret L, Van De Panne M. Locomotion skills for simulated quadrupeds. *ACM Trans Graph*. 2011;30(4):1–12.
86. Levine S, Popović J. Physically plausible simulation for character animation. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2012;221–230. Eurographics Association.
87. Yin K, Loken K, Van de Panne M. Simbicon: Simple biped locomotion control. *ACM Trans Graph*. 2007;26(3):105.
88. Wu C-C, Zordan V. Goal-directed stepping with momentum control. In: *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2010;113–8.
89. De Lasa M, Mordatch I, Hertzmann A. Feature-based locomotion controllers. *ACM Trans Graph*. 2010;29(4):1–10.
90. Coros S, Beaudoin P, Van de Panne M. Generalized biped walking control. *ACM Trans Graph*. 2010;29(4):1–9.
91. Zordan VB, Hodgins JK. Motion capture-driven simulations that hit and react. In: *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2002;89–96.
92. Da Silva M, Abe Y, Popović J. Simulation of human motion data using short-horizon model-predictive control. In: *Computer Graphics Forum*, 2008;27, 371–380. Wiley Online Library.
93. Lee Y, Kim S, Lee J. Data-driven biped control. In: *ACM SIGGRAPH 2010 Papers*, 2010;1–8.
94. Hong S, Han D, Cho K, Shin JS, Noh J. Physics-based full-body soccer motion control for dribbling and shooting. *ACM Trans Graph*. 2019;38(4):1–12.
95. Wang JM, Hamner SR, Delp SL, Koltun V. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Trans Graph*. 2012;31(4):1–11.
96. Geijtenbeek T, Van De Panne M, Van Der Stappen AF. Flexible muscle-based locomotion for bipedal creatures. *ACM Trans Graph*. 2013;32(6):1–11.
97. Lee Y, Park MS, Kwon T, Lee J. Locomotion control for many-muscle humanoids. *ACM Trans Graph*. 2014;33(6):1–11.
98. Kuo AD. Stabilization of lateral motion in passive dynamic walking. *Int J Robot Res*. 1999;18(9):917–30.
99. Obinata G, Hase K, Nakayama A. Controller design of musculoskeletal model for simulating bipedal walking. In: *Annual Conference of the International FES Society*, 2004;2, p. 1.
100. Song S, Collins SH. Optimizing exoskeleton assistance for faster self-selected walking. *IEEE Trans Neural Syst Rehabil Eng*. 2021;29:786–95.
101. Choi JT, Bastian AJ. Adaptation reveals independent control networks for human walking. *Nat Neurosci*. 2007;10(8):1055–62.
102. Torres-Oviedo G, Bastian AJ. Natural error patterns enable transfer of motor learning to novel contexts. *J Neurophysiol*. 2012;107(1):346–56.
103. Emken JL, Benitez R, Sideris A, Bobrow JE, Reinkensmeyer DJ. Motor adaptation as a greedy optimization of error and effort. *J Neurophysiol*. 2007;97(6):3997–4006.
104. Cajigas I, Koenig A, Severini G, Smith M, Bonato P. Robot-induced perturbations of human walking reveal a selective generation of motor adaptation. *Sci Robot*. 2017;2(6):1–10.
105. Sutton RS, Barto AG. *Reinforcement Learning: An Introduction*. MIT press; 2018.
106. Levine S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.
107. Schulman J, Heess N, Weber T, Abbeel P. Gradient estimation using stochastic computation graphs. *arXiv preprint arXiv:1506.05254*, 2015.
108. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. Human-level control through deep reinforcement learning. *Nature*. 2015;518(7540):529–33.
109. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al. Mastering the game of go with deep neural networks and tree search. *Nature*. 2016;529(7587), 484.
110. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, et al. Mastering the game of go without human knowledge. *Nature*. 2017;550(7676):354–9.
111. Vinyals O, Babuschkin I, Czarnecki WM, Mathieu M, Dudzik A, Chung J, Choi DH, Powell R, Ewalds T, Georgiev P, et al. Grandmaster level in Starcraft II using multi-agent reinforcement learning. *Nature*. 2019;575(7782):350–4.
112. OpenAI Spinning Up Documentation. [https://spinningup.openai.com/en/latest/spinningup/rl\\_intro2.html#id20](https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#id20). Accessed 08 Aug 2020.
113. Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust region policy optimization. In: *International Conference on Machine Learning*, 2015;1889–97.
114. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
115. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
116. Fujimoto S, Van Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: *International Conference on Machine Learning*, 2018, p. 1587–96.
117. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: *International Conference on Machine Learning*, 2018, p. 1861–70.
118. Sutton RS, McAllester DA, Singh SP, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In: *Advances in Neural Information Processing Systems*, 2000;1057–63.
119. Kakade S, Langford J. Approximately optimal approximate reinforcement learning. In: *International Conference on Machine Learning*, vol. 2. 2002;267–74.
120. Deisenroth MP, Neumann G, Peters J, et al. A survey on policy search for robotics. *Foundations and Trends in Robotics* 2013;2(1–2), 1–142.
121. Hansen N. The cma evolution strategy: a comparing review. In: *Towards a New Evolutionary Computation*, pp. 75–102. Springer, 2006.
122. Wang JM, Fleet DJ, Hertzmann A. Optimizing walking controllers for uncertain inputs and environments. *ACM Trans Graph*. 2010;29(4):1–8.
123. Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evol Comput*. 2003;11(1):1–18.
124. Peng XB, Berseth G, Van de Panne M. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Trans Graph*. 2016;35(4):1–12.
125. Yu W, Turk G, Liu CK. Learning symmetric and low-energy locomotion. *ACM Trans Graph*. 2018;37(4):1–12.
126. Liu L, Hodgins J. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Trans Graph*. 2018;37(4):1–14.
127. Clegg A, Yu W, Tan J, Liu CK, Turk G. Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. *ACM Trans Graph*. 2018;37(6):1–10.
128. Peng XB, Berseth G, Yin K, Van De Panne M. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans Graph*. 2017;36(4):1–13.
129. Heess N, TB D, Sriram S, Lemmon J, Merel J, Wayne G, Tassa Y, Erez T, Wang Z, Eslami S, Riedmiller M, Silver David. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

130. Peng XB, van de Panne M. Learning locomotion skills using deeprl: Does the choice of action space matter? In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2017;1–13.
131. Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W. Openai gym. arXiv preprint [arXiv:1606.01540](https://arxiv.org/abs/1606.01540), 2016.
132. OpenAI Spinning Up. <https://anaconda.org/kidzik/opensim>. Accessed 08 Aug 2020.
133. Kidziński Ł, Mohanty SP, Ong CF, Huang Z, Zhou S, Pechenko A, Stelmazczyk A, Jarosik P, Pavlov M, Kolesnikov S, et al. Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments. In: The NIPS'17 Competition: Building Intelligent Systems, 2018;121–153. Springer.
134. Kidziński Ł, Ong C, Mohanty SP, Hicks J, Carroll S, Zhou B, Zeng H, Wang F, Lian R, Tian H, et al. Artificial intelligence for prosthetics: Challenge solutions. In: The NeurIPS'18 Competition, 2020;69–128. Springer.
135. osim-rl. <https://github.com/stanfordnmb/osim-rl/tree/v3.0.11>. Accessed 08 Aug 2020.
136. ALCROWD — Challenges - NeurIPS 2019: Learn to Move. <https://www.aicrowd.com/challenges/neurips-2019-learning-to-move-walk-around>. Accessed 08 Aug 2020.
137. osim-rl. <http://osim-rl.stanford.edu/>. Accessed 08 Aug 2020.
138. Schwartz MH, Rozumalski A, Trost JP. The effect of walking speed on the gait of typically developing children. *J Biomech*. 2008;41(8):1639–50.
139. Hamner SR, Delp SL. Muscle contributions to fore-aft and vertical body mass center accelerations over a range of running speeds. *J Biomech*. 2013;46(4):780–7.
140. John CT, Anderson FC, Higginson JS, Delp SL. Stabilisation of walking by intrinsic muscle properties revealed in a three-dimensional muscle-driven simulation. *Comput Methods Biomech Biomed Eng*. 2013;16(4):451–62.
141. Zhou B, Zeng H, Wang F, Li Y, Tian H. Efficient and robust reinforcement learning with uncertainty-based value expansion. arXiv preprint [arXiv:1912.05328](https://arxiv.org/abs/1912.05328), 2019.
142. Kolesnikov S, Hrinchuk O. Sample efficient ensemble learning with catalyst.rl. arXiv preprint [arXiv:2003.14210](https://arxiv.org/abs/2003.14210), 2020.
143. Akimov D. Distributed soft actor-critic with multivariate reward representation and knowledge distillation. arXiv preprint [arXiv:1911.13056](https://arxiv.org/abs/1911.13056), 2019.
144. Bengio Y, Louradour J, Collobert R, Weston J. Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, 2009;41–48.
145. Hase K, Stein R. Turning strategies during human walking. *J Neurophysiol*. 1999;81(6):2914–22.
146. Taylor MJD, Dabnichki P, Strike S. A three-dimensional biomechanical comparison between turning strategies during the stance phase of walking. *Human Movement Sci*. 2005;24(4):558–73.
147. Hallemaans A, De Clercq D, Aerts P. Changes in 3d joint dynamics during the first 5 months after the onset of independent walking: a longitudinal follow-up study. *Gait Posture*. 2006;24(3):270–9.
148. Adolph KE, Cole WG, Komati M, Garciaguirre JS, Badaly D, Lingeman JM, Chan GL, Sotsky RB. How do you learn to walk? thousands of steps and dozens of falls per day. *Psychol Sci*. 2012;23(11):1387–94.
149. Heess N, Wayne G, Tassa Y, Lillicrap T, Riedmiller M, Silver D. Learning and transfer of modulated locomotor controllers. arXiv preprint [arXiv:1610.05182](https://arxiv.org/abs/1610.05182), 2016.
150. Frans K, Ho J, Chen X, Abbeel P, Schulman J. Meta learning shared hierarchies. arXiv preprint [arXiv:1710.09767](https://arxiv.org/abs/1710.09767), 2017.
151. Merel J, Botvinick M, Wayne G. Hierarchical motor control in mammals and machines. *Nat Commun*. 2019;10(1):1–12.
152. Latombe J-C. *Robot Motion Planning*, vol. 124. Springer; 2012.
153. Paden B, Čáp M, Yong SZ, Yershov D, Frazzoli E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans Intell Vehicles*. 2016;1(1):33–55.
154. World Chase Tag™ - Home. <https://www.worldchasetag.com/>. Accessed 08 Aug 2020.
155. Afschrift M, De Groot F, Jonkers I. Similar sensorimotor transformations control balance during standing and walking. *PLoS Comput Biol*. 2021;17(6).
156. Müller R, Vielmeyer J, Häufle DF. Negotiating ground level perturbations in walking: Visual perception and expectation of curb height modulate muscle activity. *J Biomech*. 2020;113:110121.
157. Chambers V, Artemiadis P. A model-based analysis of supraspinal mechanisms of inter-leg coordination in human gait: Toward model-informed robot-assisted rehabilitation. *IEEE Trans Neural Syst Rehabil Eng*. 2021;29:740–9.
158. Sánchez N, Simha SN, Donelan JM, Finley JM. Using asymmetry to your advantage: learning to acquire and accept external assistance during prolonged split-belt walking. *J Neurophysiol*. 2021;125(2):344–57.
159. Smith MA, Ghazizadeh A, Shadmehr R. Interacting adaptive processes with different timescales underlie short-term motor learning. *PLoS Biol*. 2006;4:6.
160. Caporale N, Dan Y. Spike timing-dependent plasticity: a hebbian learning rule. *Annu Rev Neurosci*. 2008;31:25–46.
161. Shadmehr R, Smith MA, Krakauer JW. Error correction, sensory prediction, and adaptation in motor control. *Ann Rev Neurosci*. 2010;33:89–108.
162. Wolpert DM, Diedrichsen J, Flanagan JR. Principles of sensorimotor learning. *Nat Rev Neurosci*. 2011;12(12):739–51.
163. Haith AM, Krakauer JW. Theoretical models of motor control and motor learning. In: *Routledge Handbook of Motor Control and Motor Learning*, 2013;16–37. Routledge.
164. Tedrake R, Zhang TW, Seung HS. Stochastic policy gradient reinforcement learning on a simple 3d biped. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), 2004;3, 2849–54.
165. Manoonpong P, Geng T, Kulvicius T, Porr B, Wörgötter F. Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Comput Biol*. 2007;3(7):134.
166. Sar P, Geyer H. A model for the transfer of control from the brain to the spinal cord through synaptic learning. *J Comput Neurosci*. 2020;48(4):365–75.
167. Alexander RM. Simple models of human movement, 1995.
168. Seyfarth A, Geyer H, Günther M, Blickhan R. A movement criterion for running. *J Biomech*. 2002;35(5):649–55.
169. Geyer H, Seyfarth A, Blickhan R. Compliant leg behaviour explains basic dynamics of walking and running. *Proc R Soc Lond B*. 2006;273(1603):2861–7.
170. Srinivasan M, Ruina A. Computer optimization of a minimal biped model discovers walking and running. *Nature*. 2006;439(7072):72–5.
171. Bauby CE, Kuo AD. Active control of lateral balance in human walking. *J Biomech*. 2000;33(11):1433–40.
172. Bertram JE, Ruina A. Multiple walking speed-frequency relations are predicted by constrained optimization. *J Theor Biol*. 2001;209(4):445–53.
173. Arechavala G, Laumond J-P, Hicheur H, Berthoz A. An optimality principle governing human walking. *IEEE Trans Robot*. 2008;24(1):5–14.
174. Brown GL, Seethapathi N, Srinivasan M. Energy optimality predicts curvilinear locomotion. arXiv preprint [arXiv:2001.02287](https://arxiv.org/abs/2001.02287), 2020.
175. Srinivasan M. Fifteen observations on the structure of energy-minimizing gaits in many simple biped models. *J R Soc Interface*. 2011;8(54):74–98.
176. Faraji S, Wu AR, Ijspeert AJ. A simple model of mechanical effects to estimate metabolic cost of human walking. *Sci Rep*. 2018;8(1):1–12.
177. Darci O, Kuo AD. Humans optimally anticipate and compensate for an uneven step during walking. *bioRxiv*, 2021;2020–12.
178. Kuo AD. The relative roles of feedforward and feedback in the control of rhythmic movements. *Motor control*. 2002;6(2):129–45.
179. Geyer H, Seyfarth A, Blickhan R. Positive force feedback in bouncing gaits? *Proc R Soc London B*. 2003;270(1529):2173–83.
180. Sanchez N, Simha SN, Donelan JM, Finley JM. Taking advantage of external mechanical work to reduce metabolic cost: the mechanics and energetics of split-belt treadmill walking. *J Physiol*. 2019;597(15):4053–68.
181. Seethapathi N, Clark B, Srinivasan M. Exploration-based learning of a step to step controller predicts locomotor adaptation. *bioRxiv*, 2021.
182. Winter DA. *Biomechanics and Motor Control of Human Movement*. New York: Wiley; 2009.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.