

# MuscleVAE: Model-Based Controllers of Muscle-Actuated Characters

Yusen Feng

ysfeng@stu.pku.edu.cn

School of CS, Peking University  
Beijing, China

Xiyan Xu

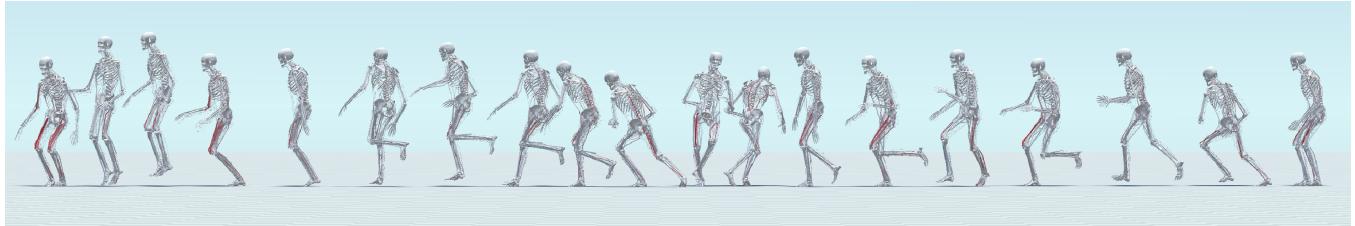
xiyan\_xu@pku.edu.cn

Peking University  
Beijing, China

Libin Liu\*

libin.liu@pku.edu.cn

Peking University  
Beijing, China  
National Key Lab of General AI  
Beijing, China



**Figure 1:** A simulated muscle-actuated character runs, walks, turns around, and performs single-leg and two-leg jumps. We present a model-based framework that uses variational autoencoders (VAE) for muscle control policy, thereby generating biologically plausible character motions. The color of the muscle curve represents the activation level; the redder the muscle, the greater its activation.

## ABSTRACT

In this paper, we present a simulation and control framework for generating biomechanically plausible motion for muscle-actuated characters. We incorporate a fatigue dynamics model, the 3CC-r model, into the widely-adopted Hill-type muscle model to simulate the development and recovery of fatigue in muscles, which creates a natural evolution of motion style caused by the accumulation of fatigue from prolonged activities. To address the challenging problem of controlling a musculoskeletal system with high degrees of freedom, we propose a novel muscle-space control strategy based on PD control. Our simulation and control framework facilitates the training of a generative model for muscle-based motion control, which we refer to as MuscleVAE. By leveraging the variational autoencoders (VAEs), MuscleVAE is capable of learning a rich and flexible latent representation of skills from a large unstructured motion dataset, encoding not only motion features but also muscle control and fatigue properties. We demonstrate that the MuscleVAE model can be efficiently trained using a model-based approach, resulting in the production of high-fidelity motions and enabling a variety of downstream tasks.

\*corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia  
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0315-7/23/12...\$15.00  
<https://doi.org/10.1145/3610548.3618137>

## CCS CONCEPTS

- Computing methodologies → Animation; Physical simulation; Reinforcement learning; Neural networks.

## KEYWORDS

muscle, fatigue simulation, motion control, generative models, VAE

## ACM Reference Format:

Yusen Feng, Xiyan Xu, and Libin Liu. 2023. MuscleVAE: Model-Based Controllers of Muscle-Actuated Characters. In *SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23)*, December 12–15, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3610548.3618137>

## 1 INTRODUCTION

Animating characters using detailed musculoskeletal models offers the potential for highly realistic and accurate character movements. However, progress in muscle-actuated character animation has been comparatively slow relative to advancements in rigid body character animation. This lag is primarily attributed to the challenges associated with the high-dimensional nature of muscle actuation space, which leads to substantial simulation times and poses significant training challenges. Additionally, muscle dynamics models used in recent computer animation research are often overly simplified. Some important factors, such as the effects of fatigue cumulation, remain inadequately explored.

In this paper, we present a comprehensive simulation and learning framework for muscle-actuated characters. Building upon the widely-adopted Hill-type muscle models [Hill 1938; Zajac 1989], we develop a new control scheme in the muscle space that efficiently determines muscle forces while taking into account the constraints

imposed by the musculotendon models. Additionally, we incorporate fatigue effects into our muscle simulator to generate motions that are more biologically accurate. Based on this simulation framework, we utilize a model based on Variational Autoencoder (VAE) to learn skill embeddings from unorganized motion data. This learned latent space of motion skills encompasses both muscle functionality and coordination, making it applicable to various downstream tasks.

To mitigate the difficulties caused by the high degrees of freedom of the muscle system, several recent successful animation systems [Lee et al. 2019; Park et al. 2022; Ryu et al. 2021] employ a two-level control framework proposed by [Lee et al. 2019]. This framework learns joint-level PD control using reinforcement learning and trains a specialized network to coordinate muscles to realize the computed joint torques. However, relying on joint-level control as the driving signal may not accurately capture the biomechanical patterns of muscle activation and may be susceptible to overfitting in specific torque regions. In this paper, we opt for controlling the character directly in the muscle space. We attach a PD servo to each muscle fiber and make the control policy compute the target length for these PD servos. The resulting muscle forces are then confined to the range determined by musculotendon models and fatigue dynamics. We find that such a straightforward strategy can effectively facilitate the learning of complex skills and can be easily incorporated into the training framework.

Fatigue is a common phenomenon in real muscle actuation systems. However, due to the challenges associated with acquiring fatigue data, quantitative studies on this issue are often limited to simple movements and specific fatigue parameters of certain body parts. Character animation studies examining the effects of fatigue on motion are also sparse [Cheema et al. 2020; Komura et al. 2000]. In this paper, we consider fatigue as an integral part of a complete musculoskeletal system. We incorporate the 3CC-r muscle fatigue model [Looft et al. 2018] into our system, expanding its scope to encompass full-body movements. Our control policy automatically shifts its strategy at different fatigue levels, generating natural change of motion patterns that are biomechanically plausible.

To learn a versatile skill representation from a large, unorganized motion dataset, many recent studies rely on model-free reinforcement techniques combined with adversarial networks [Peng et al. 2022] or autoencoders [Won et al. 2022]. However, model-free approaches can suffer from sample efficiency issues and can be difficult to converge on high-dimensional problems. Recently, model-based approaches have proven to be data efficient and stable in training complex motion controllers [Fussell et al. 2021; Hafner et al. 2023; Yao et al. 2022]. In these approaches, a world model is learned to capture the complex dynamics of character motion, which allows for differentiable training objectives. In this paper, we adopt a model-based method, ControlVAE [Yao et al. 2022], to learn generative control policies for our muscle-actuated characters. We integrate our differentiable muscle-space controller into the world model, utilizing gradient information to facilitate the learning of muscle activation coordination.

In summary, our work makes two principal contributions: (1) We propose a novel simulation and control framework for muscle-actuated characters. This framework incorporates fatigue effects

in simulation, and our muscle-space control mechanism facilitates biologically plausible control of complex human motions; (2) We develop a generative control policy for muscle-driven characters. Trained using a model-based approach, this policy not only provides a versatile skill representation for numerous downstream tasks but also automatically adjusts to fatigue levels. This adaptability ensures that characters exhibit natural variations in motion patterns during extended activities.

## 2 RELATED WORKS

### 2.1 Muscle Modeling and Simulation

Muscle modeling and simulation has been a long-standing topic in both biomechanics and computer graphics. The Hill-type model, proposed by [Hill 1938] and expanded by [Zajac 1989], numerically models musculotendon dynamics. OpenSim [Delp et al. 2007; Seth et al. 2018] leverages this model to simulate human body movement. Wang et al. [2022] accounted for muscle inertia, creating a compatible framework for the Hill-type muscle. For volumetric simulation, finite element methods (FEM) are used to model muscular soft tissue deformation [Fan et al. 2014; Lee et al. 2009; Zhu et al. 1998]. EMU [Modi et al. 2021] handles heterogeneously stiff meshes with better efficiency than FEM. Recently, various frameworks and suites for muscle simulation have been proposed, including open-source ones such as [Todorov et al. 2012; Vittorio et al. 2022], as well as commercial ones such as [Geijtenbeek 2021]. Various specific muscle models for body parts such as the face, neck, shoulder, and hand [Ichim et al. 2017; Lee and Terzopoulos 2006; Li et al. 2022; Maurel and Thalmann 2000; Srinivasan et al. 2021; Sueda et al. 2008; Van der Helm 1994; Yang et al. 2022], and animals like ostrich [Barbera et al. 2022] have also been developed.

Muscle fatigue, which is performance degradation resulting from intense muscle exercise, has been a research topic for years in biomechanics and related fields. Giat et al. [1993] analyzed fatigued quadriceps muscle, presenting a fatigue-recovery model [Giat et al. 1996, 1993]. With validations on calculated METs (Maximum Endurance Times), Ma et al. [2009] modeled fatigue patterns. Potvin and Fuglevand [2017] proposed a framework that considers the fatigue-related changes in motor unit force, but their model does not include recovery from fatigue. The works of [Komura et al. 2000; Liu et al. 2002] estimated tired poses and quantified muscle fatigue and recovery. The three-compartment controller fatigue model (3CC), proposed by Xia and Law [2008], predicts muscle fatigue in complex movements. The 3CC model was improved by adding a rest recovery parameter [Looft et al. 2018] and was integrated into reinforcement learning (RL) reward by Cheema et al. [2020] to design policies for mid-air interaction movements. We build our system on the Hill-type muscle model for its simplicity and efficiency. We augment this model with a modified 3CC-r model to simulate fatigue, and a PD control mechanism to allow efficient training of complex skills.

### 2.2 Motion Control and DRL

Reproducing realistic, interactive motions in physics-based simulation is a challenging problem. Early works relied on human insights and designed torque-actuation control strategies [Hodgins et al. 1995], while later works used learning algorithms, optimal control,

and abstract models [Muico et al. 2011; Sok et al. 2007; Yin et al. 2007]. Recently, deep reinforcement learning (DRL) has shown potential in various tasks [Lee et al. 2021; Liu and Hodgins 2017; Peng et al. 2018; Won et al. 2021; Yin et al. 2021]. Beyond merely tracking motion trajectory, a number of recent studies successfully learned generative control policies that allow efficient accomplishment of downstream tasks [Peng et al. 2022; Won et al. 2022; Yao et al. 2022].

Muscle-actuated control, built upon detailed musculoskeletal dynamics, has the potential to synthesize more realistic human postures compared to those produced by joint-actuated control [Komura et al. 2000]. Numerous efforts have been made to establish robust muscle-actuated control systems for tasks such as locomotion, swimming, and hand manipulation [Geijtenbeek et al. 2013; Geyer and Herr 2010; Lee et al. 2014; Si et al. 2014; Tsang et al. 2005; Wang et al. 2012]. Combined with deep reinforcement learning, several recent successful frameworks [Lee et al. 2019; Park et al. 2022; Ryu et al. 2021] achieve tracking control of various motion skills within a uniform framework. To mitigate the challenges posed by the high degrees of freedom of the muscle system, Lee et al. [2019] employ a two-level imitation learning algorithm, where a joint-level PD control policy is combined with a separate network to coordinate muscles and realize the computed joint torques. A more recent work, DEP-RL [Schumacher et al. 2023], also shows that this problem can be partially addressed by employing better exploration techniques in reinforcement learning. Our framework also utilizes deep reinforcement learning to train complex control policies. We directly compute muscle actuation using a novel muscle-space PD control mechanism, eliminating the need for guidance from a joint-level controller. We also employ a model-based reinforcement method, ControlVAE [Yao et al. 2022], to effectively train our control policies. Here, muscle states and fatigue information are taken into account, enabling the policy to adapt under different conditions.

### 3 MUSCLE SYSTEM

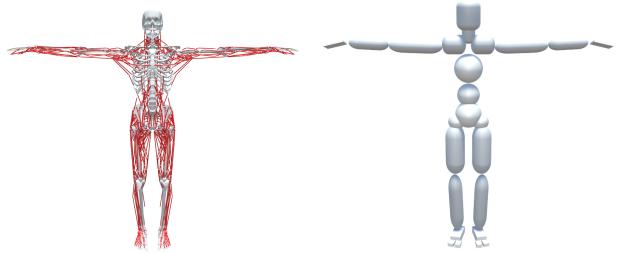
#### 3.1 Muscle Modeling

Our simulated character is adapted from the musculoskeletal model developed by Lee et al. [2019], with minor modifications made to enforce symmetry between the left and right sides of the character. As shown in Figure 2, the character model consists of eight revolute joints and fourteen ball-and-socket joints. It is actuated by 284 muscles. These muscles are the sole drivers of motion, while the joints provide the necessary physical constraints.

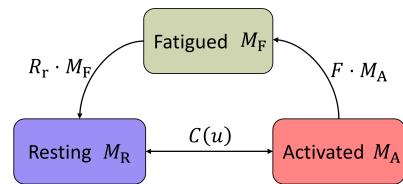
Muscles attach to bones via tendons at their ends, known as the origin and insertion. Following [Delp et al. 2007; Lee et al. 2019], we use a simplified muscle model, where each muscle is represented as polylines and may span across multiple joints. These polylines are defined by a set of anchor points, and the muscle force is considered to be transferred to the bones through these anchor points. When the character moves, the placement of these anchors is computed using Linear Blend Skinning (LBS).

#### 3.2 Muscle Dynamics

Muscles are often modeled using a simplified, three-element structure known as the Hill-type muscle model [Hill 1938; Zajac 1989]. This model comprises a contractile element (CE), a parallel elastic



**Figure 2: The muscle model and the simulated rigid bodies.** Muscles are colored in red to highlight the connections between muscles and bones.



**Figure 3: The 3CC-r model assumes that muscle actuators can be in one of three possible compartments. The differential quantities within these three compartments evolve according to their respective relationships.**

element (PE), and a tendon element. The CE represents the muscle fibers, which contract based on the muscle's activation state. It generates an active contractile force that is proportional to the level of activation  $\alpha \in [0, 1]$ . The PE represents the passive elastic material surrounding the muscle fibers and produces a passive, non-linear spring force. Following the common practice in previous works [Geijtenbeek et al. 2013; Jiang et al. 2019; Lee et al. 2019], we further simplify this model by neglecting changes in tendon length, assuming zero pennation angles, and calculating muscle length using the polylines. The muscle force generated by each muscle is then determined by the forces from the CE and PE as

$$f_m = \alpha f_{CE}^l(\bar{l}) f_{CE}^v(\dot{\bar{l}}) + f_{PE}(\bar{l}), \quad (1)$$

where  $\bar{l}$  and  $\dot{\bar{l}}$  represent the normalized muscle length and its rate of change, respectively. We compute  $\dot{\bar{l}}$  using the finite difference between consecutive frames.  $f_{CE}^l$ ,  $f_{CE}^v$ , and  $f_{PE}$  are the active force-length function, the force-velocity function, and the passive force-length function, respectively. The exact form of these functions is experimentally determined and can be found in the supplementary document.

#### 3.3 Fatigue Dynamics

Fatigue is the phenomenon in which a particular muscle cannot maintain the required force due to the accumulation of substances that cause fatigue. This loss of strength in specific muscles leads to a redistribution of muscle activation and a consequent alteration of movement patterns. In this paper, we adopt the 3CC-r model [Loof et al. 2018] to simulate fatigue effects. This model is an enhanced

version of the Three-Compartment Controller (3CC) model [Xia and Law 2008], incorporating additional factors for improved alignment with experimental data. The 3CC-r model assumes that each muscle consists of multiple hypothetical muscle-tendon actuators. Each of these actuators is presumed to be in one of three possible states: *Activated* ( $M_A$ ), *Resting* ( $M_R$ ), and *Fatigued* ( $M_F$ ). Each  $M_*$  here represents the percentage of actuators in a specific state. Actuators in the *Activated* state are considered to be ideally activated, producing maximal contractile forces. In contrast, actuators in both the *Resting* and *Fatigued* states generate no contractile forces. Based on these assumptions,  $M_A$  can be viewed as representing the activation level of the entire muscle, corresponding to  $\alpha$  in Equation (1).

In the 3CC-r model, the actuators in the *Resting* and *Activated* states can transition to the other state when needed. Once activated, the *Activated* actuators become *Fatigued* over time at a given rate, and the *Fatigued* actuators can recover gradually and revert to the *Resting* state. Figure 3 illustrates this process. The values of  $M_A$ ,  $M_R$ , and  $M_F$  are governed by a set of differential equations:

$$\dot{M}_A = C(u) - F \cdot M_A \quad (2)$$

$$\dot{M}_R = -C(u) + R_r \cdot M_F \quad \dot{M}_F = F \cdot M_A - R_r \cdot M_F, \quad (3)$$

where  $F$  and  $R_r$  denote the fatigue and recovery coefficients, respectively. The function  $C(u)$  denotes the transfer rate between  $M_R$  and  $M_A$ , determined by the difference between the target load,  $u$ , and the current activation level  $M_A$ . The target load,  $u \in [0, 100\%]$ , describes a desired level of muscle activation that the character's brain wishes to use to perform a motion. The effect of  $C(u)$  is akin to the activation dynamics [Delp et al. 2007], where  $u$  and  $M_A$  corresponds to the excitation and activation signals, respectively. Following [Loof et al. 2018; Xia and Law 2008], we use a piecewise function that increases monotonically with  $u$  to formulate  $C(u)$ . We refer readers the supplementary materials for its accurate form.

### 3.4 Muscle Space Control

The Hill-type model generates muscle forces based on the activation signals. However, in our early experiments, we found that using muscle activation levels as action space can lead to poor convergence. To remedy this problem, we use a PD control-like formulation at the muscle level to calculate the force applied by the muscle. The PD muscle force operates as

$$f_{pd} = \max(0, k_p \cdot (\tilde{l}_M - l_M) - k_d \cdot \dot{l}_M), \quad (4)$$

where  $\tilde{l}_M$  denotes a target muscle length,  $l_M$  and  $\dot{l}_M$  refer to the current length of the muscle and its rate of change, respectively.  $k_p$  and  $k_d$  are predefined PD gains. Since the force generated by the muscle can only be contractile, any negative component is eliminated.

We can compute the muscle activation  $\alpha_{pd}$  that leads to the PD muscle force  $f_{pd}$  using

$$\alpha_{pd} = \frac{f_{pd} - f_{PE}(\bar{l})}{f_{CE}^l(\bar{l})f_{CE}^v(\dot{\bar{l}})}. \quad (5)$$

However,  $\alpha_{pd}$  may not be achievable due to the muscle constraints and fatigue. In the Hill-type model, the muscle activation  $\alpha$  is confined to the range  $[0, 1]$  and evolves according to the fatigue dynamics of Equation (2). As a result, the PD muscle force computed

above is not always realizable. We argue that the feasible range of muscle force can be defined by a set of upper and lower bounds  $[f_{lb}, f_{ub}]$ . The final force applied to the character is then computed by

$$f = \text{clip}(f_{pd}, f_{lb}, f_{ub}). \quad (6)$$

To find  $f_{lb}$  and  $f_{ub}$ , considering that the muscle activation,  $\alpha$ , and percentage of activated actuators,  $M_A$ , are equivalent, the discrete form of the fatigue dynamics of Equation (2) can be written as

$$(\tilde{\alpha} - \alpha)/\Delta t = C(u) - F\alpha, \quad (7)$$

$$\text{or, } \tilde{\alpha} = \tilde{\alpha}(u) = (1 - \Delta t F)\alpha + \Delta t C(u), \quad (8)$$

where  $\Delta t$  represents the time interval. Equation (8) suggests that the next activation level,  $\tilde{\alpha}$ , is determined by the previous muscle activation level,  $\alpha$ , and the target load,  $u$ . Given that  $u$  can be freely selected within the range  $[0, 1]$  and considering that  $C(u)$  increases monotonically with  $u$ , it follows that  $\tilde{\alpha} \in [\tilde{\alpha}(u=0), \tilde{\alpha}(u=1)]$ . Furthermore, based on the Hill-type model presented in Equation (1), the muscle force also rises monotonically with muscle activation. Thus, the force bounds can be computed as

$$f_{lb} = \tilde{\alpha}(u=0) \cdot f_{CE}^l(\bar{l})f_{CE}^v(\dot{\bar{l}}) + f_{PE}(\bar{l}) \quad (9)$$

$$f_{ub} = \tilde{\alpha}(u=1) \cdot f_{CE}^l(\bar{l})f_{CE}^v(\dot{\bar{l}}) + f_{PE}(\bar{l}) \quad (10)$$

We employ a control policy  $\pi$  to calculate an action vector  $a$  that contains the action  $a$  for each muscle. The target muscle length,  $\tilde{l}_M$ , is then computed using

$$\tilde{l}_M = (a + 1.0) \cdot l_M^{\text{tpose}}, \quad (11)$$

where  $l_M^{\text{tpose}}$  denotes the reference muscle length, computed in the T-pose of the character. After computing  $f_{pd}$ , we apply the force bounds using Equation (6) and solve for the target load  $u$  that yields  $f$ . Finally,  $f$  is applied to actuate the character, while the corresponding  $u$  is used to simulate muscle fatigue using the 3CC-r model.

We find that this muscle PD control mechanism significantly facilitates training compared to using activation control. The improvement can be attributed to the introduction of local feedback loops by the PD control, which allows the system to self-adjust and self-correct. This finding aligns with the comparison made between joint-level PD control and torque control done by [Peng and van de Panne 2017]. In this context, muscle activations can be analogously related to joint torques, and our PD muscle force resembles the joint-level PD control.

The procedures described above are all differentiable. We can concisely write them as

$$\tilde{\alpha}, s_{\text{muscle}} = D(s_{\text{skeleton}}, s_{\text{fatigue}}, a, \alpha). \quad (12)$$

In this formulation, we use bold symbols to collectively represent the corresponding values for all the muscles.  $\tilde{\alpha}$  and  $\alpha$  represent the next and previous activation values, respectively.  $s_{\text{skeleton}}$  refers to the state of the skeleton, while  $s_{\text{fatigue}}$  denotes the fatigue state of the muscles. The muscle kinematic state,  $s_{\text{muscle}}$ , contains the  $l_M$  and  $\dot{l}_M$  values and is derived from  $s_{\text{skeleton}}$  using LBS.  $a$  represents the action computed by the policy  $\pi$ . And lastly,  $D$  denotes the entire procedure.

## 4 MUSCLE VAE

In this section, we introduce our model-based, muscle-actuated motion control framework, which we refer to as *MuscleVAE*. This framework is inspired by ControlVAE [Yao et al. 2022]. Formally, as depicted in Figure 4, our objective is to learn a policy,  $\pi$ , associated with a latent space, denoted as  $\mathcal{Z}$ . The latent space  $\mathcal{Z}$  encapsulates all the skills of a motion dataset. When a latent code  $z \in \mathcal{Z}$  is selected from this space, the policy  $\pi(a|s, z)$  converts it into an action vector  $a$  according to the current state of the character  $s$ . This action vector  $a$  then determines muscle forces, as outlined in Sec 3.4, that actuates the character to perform a particular skill.

The state  $s$  represents the complete state of the character, comprising both the skeleton state  $s_{\text{skeleton}}$  and the fatigue state  $s_{\text{fatigue}}$ . Note that we do not include the muscle state  $s_{\text{muscle}}$  in  $s$  as it can be directly computed from  $s_{\text{skeleton}}$ . We adopt the same skeleton state representation as described in [Yao et al. 2022], which includes the positions, orientations, and velocities of all rigid bones. When provided with motion data as input, our framework extracts skeleton states  $\{\tilde{s}_{\text{skeleton}}^t\}$  and muscle states  $\{\tilde{s}_{\text{muscle}}^t\}$  from it, using these as the reference states. Here,  $t$  represents the time index, the tilde symbols indicate quantities that are derived from the motion data, rather than from simulation, and the braces denote a sequence of such quantities. The input motions do not need fatigue data.

Following ControlVAE [Yao et al. 2022], we train a motion encoder, represented as a posterior distribution  $q(z^t|s^t, \tilde{s}_{\text{skeleton}}^{t+1})$ , to convert a motion into latent codes  $\{z^t\}$ , where  $\{\tilde{s}_{\text{skeleton}}^{t+1}\}$  are the reference states extracted from the motion and  $\{s^t\}$  are the simulation states of the character. These latent codes  $\{z^t\}$  can be decoded by the policy  $\pi$  into muscle forces to actuate the character, allowing it to reproduce the input motion in simulation. The VAE framework imposes additional regularization on the posterior distribution  $q(z^t|s^t, \tilde{s}_{\text{skeleton}}^{t+1})$ , encouraging it to stay close to a prior distribution  $p(z)$ . This scheme ensures that a random latent code sampled from the prior distribution can be decoded into a valid motion. A common choice for the prior distribution  $p(z)$  is the standard normal distribution  $\mathcal{N}(0, 1)$ . However, Yao et al. [2022] suggest that a state-dependent prior distribution  $p(z|s)$  can provide better performance. We thus adopt the same prior distribution in our framework. At runtime, the latent code can be either computed by the posterior encoder, sampled using the prior distribution, or provided by a high-level policy of a downstream task.

We formulate the components of the MuscleVAE, specifically the policy  $\pi(a|s, z)$ , the posterior distribution  $q(z^t|s^t, \tilde{s}_{\text{skeleton}}^{t+1})$ , and the state-dependent prior distribution  $p(z|s)$ , as normal distributions in the form of  $\mathcal{N}(\mu_*(\cdot; \theta_*), \sigma_*^2 I)$ . Here,  $\sigma_*$  is a predefined standard deviation, and the mean  $\mu_*(\cdot; \theta_*)$  is represented by a neural network with trainable parameters  $\theta_*$ . The structure of these networks can be found in the supplementary materials.

### 4.1 Training

We train MuscleVAE on a dataset containing multiple unstructured motion sequences, using an approach similar to ControlVAE [Yao et al. 2022]. During the training process, our framework iteratively extracts short motion clips from the dataset, encodes them using the posterior distribution  $q(z|s, \tilde{s}_{\text{skeleton}})$ , decodes the resulting latent codes with the policy  $\pi(a|s, z)$ , and reconstructs the motion clips via

simulation. We train all components of MuscleVAE simultaneously, aiming to minimize the reconstruction error while ensuring the posterior distribution and the state-dependent prior distribution  $p(z|s)$  stay close to each other. Formally, this objective can be written as

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \beta \mathcal{L}_{\text{kl}} + \mathcal{L}_{\text{act}}, \quad (13)$$

where  $\beta$  is a weight parameter suggested by Higgins et al. [2017]. The reconstruction loss  $\mathcal{L}_{\text{rec}}$  measures the discrepancy between the simulated states and the reference states. In MuscleVAE, we consider both the skeletal motion and muscle length, so

$$\mathcal{L}_{\text{rec}} = \sum_{t=0} \gamma^t \left[ \|\tilde{s}_{\text{skeleton}}^t - s_{\text{skeleton}}^t\|_W^2 + \|\tilde{s}_{\text{muscle}}^t - s_{\text{muscle}}^t\|_{W'}^2 \right], \quad (14)$$

This function is evaluated over the motion sequence. Here,  $W$  and  $W'$  represent weight matrices, and  $\gamma$  is a discount factor. The KL-divergence loss

$$\mathcal{L}_{\text{kl}} = \sum_{t=0} \gamma^t \mathcal{D}_{\text{KL}}(q(z^t|s^t, \tilde{s}_{\text{skeleton}}^{t+1}) \| p(z^t|s^t)) \quad (15)$$

penalizes the difference between the prior and posterior distributions. Finally, to mitigate excessive control and satisfy the biological requirements of minimal bioenergy and activation thresholds, we use a combination of  $L_1$  and  $L_2$  losses on the activation level, hence the regularization loss  $\mathcal{L}_{\text{act}}$  is defined as

$$\mathcal{L}_{\text{act}} = \sum_{t=0} \gamma^t \left( w_{a_1} \|\alpha_{\text{pd}}^t\|_1 + w_{a_2} \|\alpha_{\text{pd}}^t\|_2 \right), \quad (16)$$

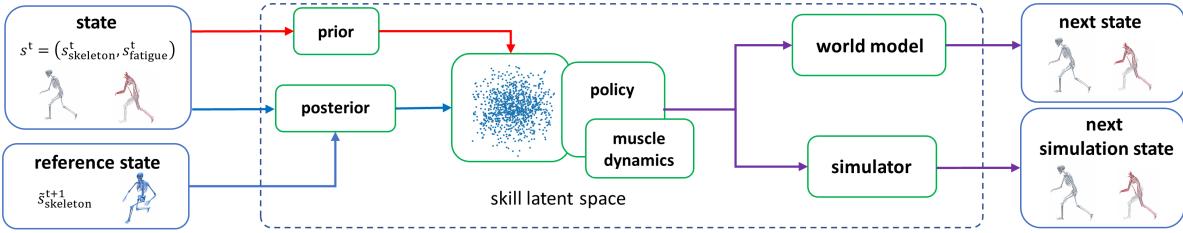
where  $w_{a_1}$  and  $w_{a_2}$  are weights of the regularization terms.

**4.1.1 Model-based Learning.** The objective function, as shown in Equation (13), cannot be directly optimized since evaluating it requires going through a complex rigid body simulation. Our system treats the simulation procedure as a black box. This approach potentially allows our system to accommodate various simulation backends, but it also makes the simulation procedure non-differentiable. Instead, as suggested by Yao et al. [2022], we adopt a model-based training procedure for our MuscleVAE, given its proven efficiency.

Briefly speaking, we train a world model,  $\omega$ , to approximate the dynamics of the musculoskeletal system,  $s^{t+1} = \omega(s^t, a^t)$ . Then,  $\omega$  is used as a substitute for the real simulation when evaluating Equation (13). We formulate  $\omega$  as a neural network, making it differentiable and enabling the optimization of Equation (13) through gradient-based methods. To train  $\omega$ , our system first generates a simulation sequence  $\{s^0, a^0, s^1, a^1, \dots\}$  by repeatedly executing the current MuscleVAE to a track random motion sequence in the real simulation. Then, starting from  $s^0$ , our system creates a synthetic sequence by executing the same series of actions  $\{a^t\}$  in the world model. This results in a sequence of synthetic states  $\{\tilde{s}^t\}$ , where  $\tilde{s}^{t+1} = \omega(\tilde{s}^t, a^t)$ . At last,  $\omega$  is trained by optimizing the objective

$$\mathcal{L}_w = \sum_{t=0} \|\tilde{s}^{t+1} - s^{t+1}\|_{\tilde{W}}^2 \quad (17)$$

against these simulation samples.  $\tilde{W}$  represents a weight matrix. The training of the world model and that of MuscleVAE's components are interleaved. When one is being trained, the other is frozen.



**Figure 4: Overview of our MuscleVAE System.**

We employ a world model similar to that used in ControlVAE [Yao et al. 2022], which is formulated in maximal coordinates. In practice, we find that our musculoskeletal system is more sensitive to the accumulative error of the world model than the rigid body system used by Yao et al. [2022], especially in the early stage of the training. This is because a small change in the length of certain muscles can lead to excessive muscle forces. We thus employ an additional differentiable forward kinematics pass to mitigate such errors.

During training, we randomly switch between reference motion clips to expose the model to different motion patterns. This abrupt switching also helps the model adaptively learn from mismatched poses and velocities and recover from such disparities. Additionally, we randomly set fatigue states for the character at the beginning of each training rollout to allow the trained MuscleVAE to be robust in different fatigue configurations. For the formulation of the world model and the training details, we refer readers to [Yao et al. 2022] and the supplementary materials.

**4.1.2 High-Level Policies.** With a trained MuscleVAE, our system further enables a downstream task to be accomplished through a goal-conditioned task policy  $\pi(z|s, g)$ . This task policy  $\pi(z|s, g)$  operates in the latent space  $\mathcal{Z}$  and outputs latent codes based on the current state  $s$  and the goal  $g$  of the task. Following [Yao et al. 2022], we train  $\pi(z|s, g)$  in a model-based manner with the trained MuscleVAE and the world model kept fixed during this phase. When given the objective function of the task, denoted as  $\mathcal{L}_g$ , our system repeatedly executes the task policy  $\pi(z|s, g)$  and then decodes the resulting latent codes into a motion sequence using both the policy  $\pi$  and the world model  $\omega$ . The performance of these motion sequences is evaluated against  $\mathcal{L}_g$ , and the policy  $\pi(z|s, g)$  is subsequently updated to minimize  $\mathcal{L}_g$ .

## 5 EXPERIMENTS

### 5.1 System Setup

The muscle-actuated character depicted in Figure 2 is used in all our experiments. It has a height 1.68 m, weighs 61.4 kg, consists of 23 rigid bodies connected by 22 joints, and is actuated by 284 muscles. The muscle model, including both the muscle dynamics and fatigue dynamics, operates at a frequency of 120 Hz. The simulation framework is implemented based on the rigid body simulator, Open Dynamics Engine (ODE). To ensure numerical stability with this relatively large time step, we additionally incorporate implicit joint damping into ODE, as suggested by Liu et al. [2013]. The MuscleVAE is implemented and trained using PyTorch [Paszke et al. 2019] and runs at a lower frequency of 20 Hz. When the MuscleVAE policy

computes an action, this action is reused for the subsequent six simulation steps. The entire system achieves real-time performance, enabling interactive control of the simulated character.

We train our MuscleVAE on a dataset containing approximately 25 minutes of motion sequences. These motion sequences are selected from the open-source LaFAN dataset [Harvey et al. 2020]. They include a variety of locomotion skills, including walking, running, turning, hopping, and jumping. The model is trained for 20,000 iterations. Our unoptimized implementation takes about 1 week to train using six parallel threads on a workstation equipped with Intel Xeon Gold 6133 CPU and a single NVIDIA GTX 3090 graphics card.

### 5.2 Evaluation

We evaluate the effectiveness of our learned MuscleVAE using three types of tasks: tracking, generation, and fatigue simulation.

**Tracking.** The MuscleVAE trained on the large locomotion dataset can be used to track other similar locomotion clips, even if they were not used for training. In this task, the posterior distribution is used to compute the latent codes of the input motion. The character can perform the input motion accurately over an extended time frame. When given another clip, it can figure out a smooth transition and then tracks the new target motion. If the two clips differ too significantly, such a switch can lead to abrupt movement. However, the character remains balanced thanks to the robustness of the control policy. While the MuscleVAE model struggles more with motions substantially different from its training dataset, like tracking a dance using the locomotion MuscleVAE, it still strives to reproduce the motion as accurately as possible. Figure 5 shows screenshots of several results. Note that the input motions do not include muscle-related data. The MuscleVAE automatically recovers such information by reproducing the motion in simulation. Figure 7 shows the muscle activation level curves when tracking four different motions.

To demonstrate the capability of our MuscleVAE to handle more dynamic and challenging short motions, we train a separate MuscleVAE on a *Jump Spin Kick* motion from the SFU Motion Capture Database [Ying and Yin 2023]. The policy allows the character to perform the skill indefinitely, as sketched in Figure 5h. We encourage readers to view the supplemental video for a better visual evaluation of the results.

**Generation.** We experiment with two generation tasks, both using the trained locomotion MuscleVAE. In the *random sampling* task, we draw random latent codes from the state-dependent prior

distribution and decode these codes into motion using the policy and simulation. As shown in Figure 6, our MuscleVAE generates a diverse range of high-quality motions in this setting. The fluctuating latent codes can cause the character to frequently change its skills, leading to occasional jittering. However, this issue can be mitigated by sampling with smaller noise.

In the *high-level control* task, we consider a downstream task of controlling the speed and direction of the character. We train a task policy to compute the latent codes with the objective of minimizing the discrepancy between speed and heading direction of the character and input from user. Once trained, this policy allows the character to adjust its direction and speed smoothly in response to interactive user commands. Please refer to the supplementary video for a visualization of such behaviors.

**Fatigue.** In both the tracking and generation tasks, the character adapts its movement based on the evolution of the fatigue stage. Fatigue accumulates naturally based on the muscle activation levels, leading to varied patterns over extended exercises. Figure 8 depicts the changes of the fatigue parameter  $M_F$  when the character is instructed to track the same motion indefinitely. It can be observed that more dynamic motions can result in faster fatigue accumulation.

To better demonstrate the fatigue effect, we instruct the character to hold its arms horizontally using the muscle-space PD control. The character's body is fixed to prevent it from falling. Figure 10 illustrates the changes in activation level and the fatigue parameter over time. As depicted in Figure 9 and also shown in the supplementary video, the character's arm gradually descends due to fatigue accumulation. When we let the character rest its arm for a certain amount of time (120°180s in Figure 10), the fatigued muscle recovers, enabling the character to raise its arm back to its initial height. This experiment can be extended to more complex motions. Specifically, we instruct the character to run for an extended period, causing it to become fatigued and subsequently run in a less powerful manner. After this, the character is directed to walk or stand for a few seconds. During this time, the character's fatigue recovers, allowing it to return to its original running style. Figure 11 shows the fatigue curves for two major muscles in both the run-walk and run-stand settings. Notably, muscle fatigue recovers faster when the character is standing than walking, and the character exhibits a better recovery of its motion style in the subsequent running. In this case, we increased fatigue rate by 50 times for faster fatigue manifestation.

We train our MuscleVAE using a predefined set of parameters for the fatigue dynamics. However, the trained model can resist changes to such parameters. To demonstrate this, we increase the values of  $F$  and  $R_r$  in the 3CC-r model to 5, 10 and 25 times that used in training, respectively. The character can still perform locomotion under the control of the trained MuscleVAE. However, with the latter settings, it becomes fatigued much faster, causing a more rapid change in motion style.

### 5.3 Ablation Study

To show the importance of our proposed muscle-space control framework, we compare its performance with a vanilla muscle control strategy that directly controls the muscles using activation

signals. Given that muscle activations are confined to the range of  $[0, 1]$ , as suggested by Lee et al. [2019], we introduce an additional activation function in the form of  $\text{ReLU}(\text{Tanh}(x))$  after the last layer of the MuscleVAE policy. This modification ensures compliance with the aforementioned range constraint. All other network configurations, including the prior distribution, the posterior distribution, the policy, and the world model, remain unchanged. The physical parameters of the character also remain the same. Figure 12 shows typical learning curves of MuscleVAEs with the proposed muscle-space control and the vanilla muscle activation control. The results indicate that the system struggles to find a feasible MuscleVAE in the MuscleVAE + vanilla activation control setting. The character is unable to maintain balance, leading to an early plateau in the reward without any growth.

## 6 CONCLUSION

In this paper, we present a comprehensive simulation and control framework for muscle-actuated characters. We augment the widely used Hill-type muscle mechanism with the 3CC-r fatigue dynamics model, effectively simulating activation dynamics and fatigue effects. We further propose a muscle-space control mechanism that combines PD control with a simple strategy that equivalently realizes the fatigue dynamics using more efficient clip operations. This control framework allows for the learning of a VAE-based generative control model, the MuscleVAE, which can accommodate a diverse range of movements in an unstructured motion dataset. The MuscleVAE model enables the encoding of not only motion features but also muscle control and fatigue properties within a rich and flexible latent space. With the aid of the MuscleVAE, we can easily recover muscle dynamics information from a motion by tracking it in simulation. Moreover, The muscle-actuated character can generate a variety of motion skills by sampling from the latent space and can accomplish downstream tasks by learning high-level control policies that operate in that latent space. The MuscleVAE model incorporates fatigue states into all its components, leading to a natural evolution of motion styles during extended exercises. We believe that our findings extend beyond the realm of graphics and have potential implications in other domains such as biomechanics and human-computer interaction.

Currently, there are several limitations in our framework. First, our simulation model and control strategy encompass numerous parameters. Many of these parameters are borrowed from existing literature [Lee et al. 2019; Loof et al. 2018; Xia and Law 2008] but were originally measured for or designed around human bodies with distinct properties, possibly making them inaccurate for the character we have utilized. A future research avenue could involve automatically determining these parameters from motion data or body measurements. Second, our control model does not account for muscle coordination. All the muscles are currently controlled individually. Despite our results indicating some coordination, there are times when the character activates both agonist and antagonist muscles simultaneously, which is not biomechanically accurate. Incorporating more biomechanical aspects, like relationships among agonists, antagonists, and synergists during a movement, could increase the control's biomechanical precision. Third, MuscleVAE aims to replicate reference motions faithfully, but if these motions

are not physically viable, the resultant motion can produce unsatisfactory artifacts. For instance, because of retargeting errors, the character's legs sometimes intersect in our training motions. If we track these motions with self-collision activated, the character may trip and fall. While fine-tuning the policy with self-collision helps the character maintain balance post-trip, it does not rectify the issue present in the reference motion. Another example is our training of MuscleVAE on a manually crafted *Horse Stance* motion. Its imbalance causes a conflict between pose tracking and balance maintenance, making the character oscillate and result in a wagging motion. Lastly, this training objective encourages the character to reproduce training motions without considering its fatigue level. This often leads the character to sustain its motion until it loses balance. Fatigue should not only be perceived as a mechanical effect but also as a stylistic element to motion. Exploring the transition from mechanical changes due to fatigue to the consequent changes in style is a noteworthy research challenge.

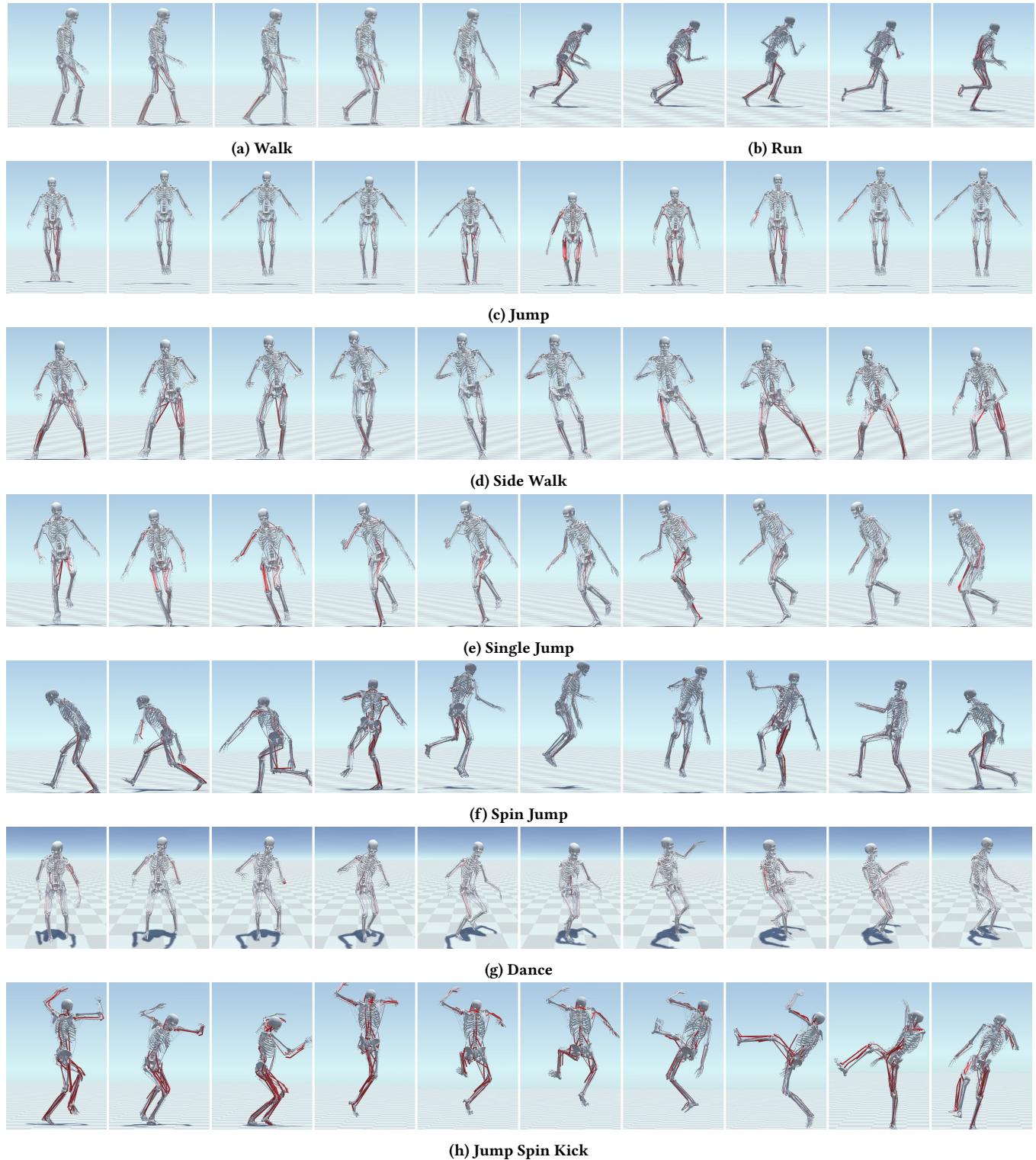
## ACKNOWLEDGMENTS

We would like to thank Baoquan Chen for his insightful discussions and assistance. We are also thankful to Zhenhua Song for his invaluable suggestions on the rendering system and rigid body simulation, and to Heyuan Yao for his inspiring conversations on generative models and his organization of open-source code. We also appreciate the anonymous reviewers for their constructive feedback. This work was supported in part by The Fundamental Research Funds for the Central Universities, Peking University.

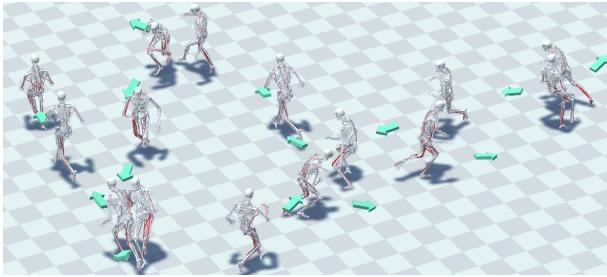
## REFERENCES

- Vittorio La Barbera, Fabio Pardo, Yuval Tassa, Monica Daley, Christopher Richards, Petar Kormushev, and John Hutchinson. 2022. OstrichRL: A Musculoskeletal Ostrich Simulation to Study Bio-mechanical Locomotion. arXiv:2112.06061 [cs.RO]
- Noshaba Cheema, Laura A Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämäläinen. 2020. Predicting mid-air interaction movements and fatigue using deep reinforcement learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
- Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering* 54, 11 (2007), 1940–1950.
- Ye Fan, Joshua Litven, and Dinesh K Pai. 2014. Active volumetric musculoskeletal systems. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–9.
- Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. SuperTrack: Motion Tracking for Physically Simulated Characters Using Supervised Learning. *ACM Trans. Graph.* 40, 6, Article 197 (dec 2021), 13 pages. <https://doi.org/10.1145/3478513.3480527>
- Thomas Geijtenbeek. 2021. The Hyfydy Simulation Software. <https://hyfydy.com>
- Thomas Geijtenbeek, Michiel Van De Panne, and A Frank Van Der Stappen. 2013. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–11.
- Hartmut Geyer and Hugh Herr. 2010. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering* 18, 3 (2010), 263–273.
- Yohanan Giat, Joseph Mizrahi, and Mark Levy. 1996. A model of fatigue and recovery in paraplegic's quadriceps muscle subjected to intermittent FES. (1996).
- Yohanan Giat, Joseph Mizrahi, Mark Levy, et al. 1993. A musculotendon model of the fatigue profiles of paralyzed quadriceps muscle under FES. *IEEE transactions on biomedical engineering* 40, 7 (1993), 664–674.
- Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2023. Mastering Diverse Domains through World Models. *arXiv preprint arXiv:2301.04104* (2023).
- Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 60–1.
- Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew W. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*.
- Archibald Vivian Hill. 1938. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B-Biological Sciences* 126, 843 (1938), 136–195.
- Jessica K Hodgins, Wayne L Wooten, David C Brogan, and James F O'Brien. 1995. Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 71–78.
- Alexandru-Eugen Ichim, Petr Kadlecák, Ladislav Kavan, and Mark Pauly. 2017. Phace: Physics-based face modeling and animation. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.
- Yifeng Jiang, Tom Van Wouwe, Friedl De Groote, and C Karen Liu. 2019. Synthesis of biologically realistic human motion using joint torque actuation. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.
- Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L Kunii. 2000. Creating and retargeting motion by the musculoskeletal human body model. *The visual computer* 16, 5 (2000), 254–270.
- Seyoung Lee, Sunmin Lee, Yongwoo Lee, and Jehee Lee. 2021. Learning a family of motor skills from a single motion clip. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.
- Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. 2019. Scalable Muscle-Actuated Human Simulation and Control. *ACM Trans. Graph.* 38, 4, Article 73 (jul 2019), 13 pages. <https://doi.org/10.1145/3306346.3322972>
- Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. 2009. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Transactions on Graphics (TOG)* 28, 4 (2009), 1–17.
- Sung-Hee Lee and Demetri Terzopoulos. 2006. Heads up! Biomechanical modeling and neuromuscular control of the neck. In *ACM SIGGRAPH 2006 Papers*. 1188–1198.
- Yoonsang Lee, Moon Seok Park, Taesoo Kwon, and Jehee Lee. 2014. Locomotion control for many-muscle humanoids. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 1–11.
- Yuwei Li, Longwen Zhang, Zesong Qiu, Yingwenqi Jiang, Nianyi Li, Yuexin Ma, Yuyao Zhang, Lan Xu, and Jingyi Yu. 2022. NIMBLE: a non-rigid hand model with bones and muscles. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–16.
- Jing Z Liu, Robert W Brown, and Guang H Yue. 2002. A dynamical model of muscle activation, fatigue, and recovery. *Biophysical journal* 82, 5 (2002), 2344–2359.
- Libin Liu and Jessica Hodgins. 2017. Learning to Schedule Control Fragments for Physics-Based Characters Using Deep Q-Learning. *ACM Transactions on Graphics* 36, 4 (June 2017), 42a:1. <https://doi.org/10.1145/3072959.3083723>
- Libin Liu, KangKang Yin, Bin Wang, and Baining Guo. 2013. Simulation and Control of Skeleton-Driven Soft Body Characters. *ACM Transactions on Graphics* 32, 6 (Nov. 2013), 1–8.
- John M Loft, Nicole Herkert, and Laura Frey-Law. 2018. Modification of a three-compartment muscle fatigue model to predict peak torque decline during intermittent tasks. *Journal of biomechanics* 77 (2018), 16–25.
- Liang Ma, Damien Chablat, Fouad Bennis, and Wei Zhang. 2009. A new simple dynamic muscle fatigue model and its validation. *International journal of industrial ergonomics* 39, 1 (2009), 211–220.
- Walter Maurel and Daniel Thalmann. 2000. Human shoulder modeling including scapulo-thoracic constraint and joint sinus cones. *Computers & Graphics* 24, 2 (2000), 203–218.
- Vismay Modi, Lawson Fulton, Alec Jacobson, Shinjiro Sueda, and David IW Levin. 2021. Emu: Efficient muscle simulation in deformation space. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 234–248.
- Uldarico Muico, Jovan Popović, and Zoran Popović. 2011. Composite control of physically simulated characters. *ACM Transactions on Graphics (TOG)* 30, 3 (2011), 1–11.
- Jungnam Park, Sehee Min, Phil Sik Chang, Jaedong Lee, Moon Seok Park, and Jehee Lee. 2022. Generative gaitnet. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems* 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035.
- Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. 2018. Deep-mimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)* 37, 4 (2018), 1–14.
- Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *arXiv preprint arXiv:2205.01906* (2022).
- Xue Bin Peng and Michiel van de Panne. 2017. Learning Locomotion Skills Using DeepRL: Does the Choice of Action Space Matter?. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) (SCA '17). ACM, New York, NY, USA, Article 12, 13 pages. <https://doi.org/10.1145/3099564.3099567>
- Jim R. Potvin and Andrew J. Fuglevand. 2017. A motor unit-based model of muscle fatigue. *PLOS Computational Biology* 13, 6 (06 2017), 1–30. <https://doi.org/10.1371/journal.pcbi.10371>

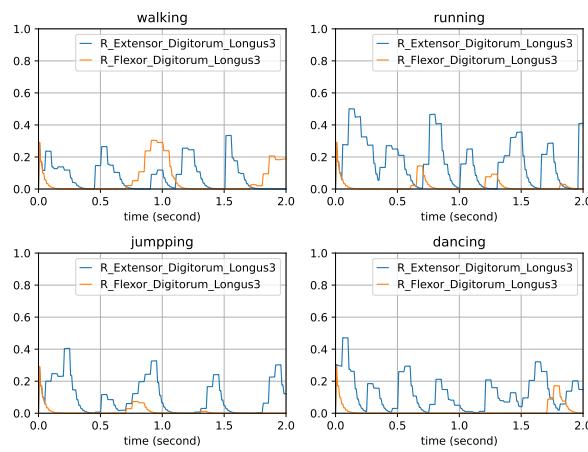
- journal.pcbi.1005581  
Hoseok Ryu, Minseok Kim, Seungwhan Lee, Moon Seok Park, Kyoungmin Lee, and Je-hee Lee. 2021. Functionality-Driven Musculature Retargeting. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 341–356.
- Pierre Schumacher, Daniel F.B. Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. 2023. DEP-RL: Embodied Exploration for Reinforcement Learning in Overactuated and Musculoskeletal Systems. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*. [https://openreview.net/forum?id=C-xa\\_D3oTj6](https://openreview.net/forum?id=C-xa_D3oTj6)
- Ajay Seth, Jennifer L Hicks, Thomas K Uchida, Ayman Habib, Christopher L Dembia, James J Dunne, Carmichael F Ong, Matthew S DeMers, Apoorva Rajagopal, Matthew Millard, et al. 2018. OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology* 14, 7 (2018), e1006223.
- Weiguang Si, Sung-Hee Lee, Eftychios Sifakis, and Demetri Terzopoulos. 2014. Realistic biomechanical simulation and control of human swimming. *ACM Transactions on Graphics (TOG)* 34, 1 (2014), 1–15.
- Kwang Won Sok, Manmyung Kim, and Jehee Lee. 2007. Simulating biped behaviors from human motion data. In *ACM SIGGRAPH 2007 papers*, 107–es.
- Sangeetha Gramma Srinivasan, Qisi Wang, Junior Rojas, Gergely Klár, Ladislav Kavan, and Eftychios Sifakis. 2021. Learning active quasistatic physics-based models from data. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.
- Shinjiro Sueda, Andrew Kaufman, and Dinesh K Pai. 2008. Musculotendon simulation for hand animation. In *ACM SIGGRAPH 2008 papers*, 1–8.
- Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033. <https://doi.org/10.1109/IROS.2012.6386109>
- Winnie Tsang, Karan Singh, and Eugene Fiume. 2005. Helping hand: an anatomically accurate inverse dynamics solution for unconstrained hand motion. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 319–328.
- Frans CT Van der Helm. 1994. A finite element musculoskeletal model of the shoulder mechanism. *Journal of biomechanics* 27, 5 (1994), 551–569.
- Caggiano Vittorio, Wang Huawei, Durandau Guillaume, Sartori Massimo, and Kumar Vikash. 2022. MyoSuite – A contact-rich simulation suite for musculoskeletal motor control. <https://github.com/facebookresearch/myosuite>. <https://arxiv.org/abs/2205.13600>
- Jack M Wang, Samuel R Hamner, Scott L Delp, and Vladlen Koltun. 2012. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–11.
- Ying Wang, Jasper Verheul, Sang-Hoon Yeo, Nima Khademi Kalantari, and Shinjiro Sueda. 2022. Differentiable simulation of inertial musculotendons. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–11.
- Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2021. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–11.
- Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2022. Physics-based character controllers using conditional VAEs. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–12.
- Ting Xia and Laura A Frey Law. 2008. A theoretical approach for modeling peripheral muscle fatigue and recovery. *Journal of biomechanics* 41, 14 (2008), 3046–3052.
- Lingchen Yang, Byungssoo Kim, Gaspard Zoss, Baran Gözcü, Markus Gross, and Barbara Solenthaler. 2022. Implicit neural representation for physics-driven actuated soft bodies. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–10.
- Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. 2022. ControlVAE. *ACM Transactions on Graphics* 41, 6 (nov 2022), 1–16. <https://doi.org/10.1145/3550454.3555434>
- KangKang Yin, Kevin Loken, and Michiel Van de Panne. 2007. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 105–es.
- Zhiqi Yin, Zeshi Yang, Michiel Van De Panne, and KangKang Yin. 2021. Discovering diverse athletic jumping strategies. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–17.
- Goh Jing Ying and KangKang Yin. 2023. SFU Motion Capture Database. <https://mocap.cs.sfu.ca/>. Accessed: 2023/09/01.
- Felix E Zajac. 1989. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical reviews in biomedical engineering* 17, 4 (1989), 359–411.
- Qing-hong Zhu, Yan Chen, and Arie Kaufman. 1998. Real-time biomechanically-based muscle volume deformation using FEM. In *Computer Graphics Forum*, Vol. 17. Wiley Online Library, 275–284.



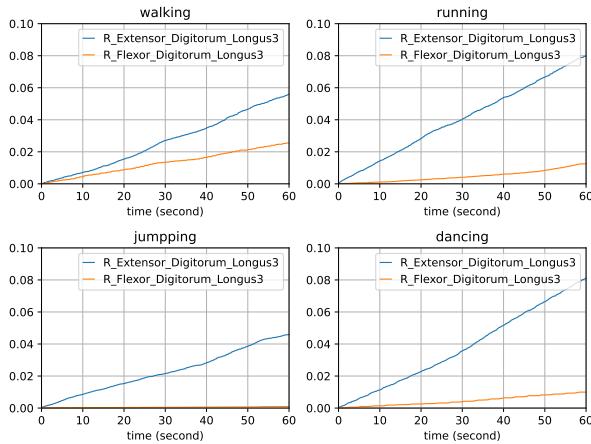
**Figure 5:** The muscle-actuated character can perform a diverse range of motions.



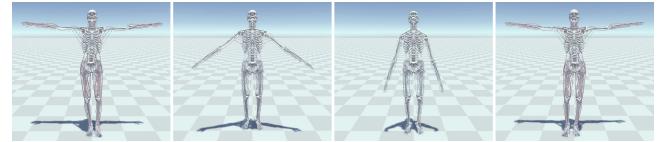
**Figure 6:** Trajectories generated by the muscle-actuated character in the random sampling experiment. The arrows indicate moving directions.



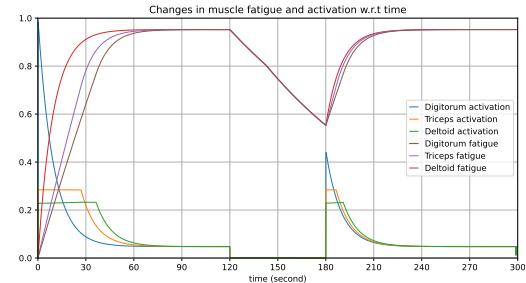
**Figure 7:** Muscle activation curves of different motions.



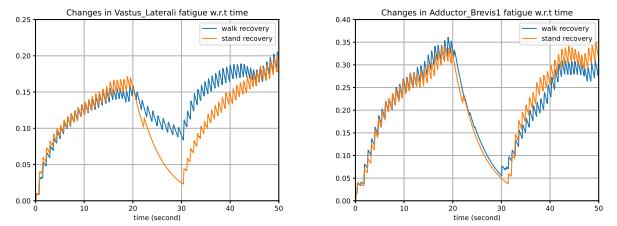
**Figure 8:** Muscle fatigue curves of different motions.



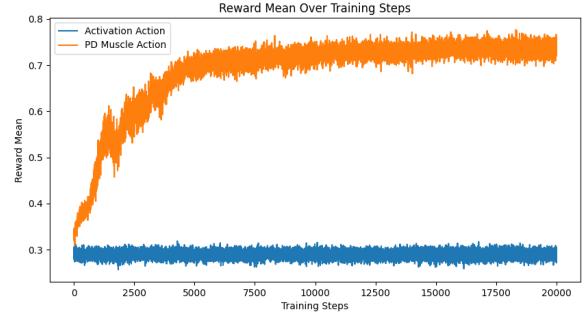
**Figure 9:** The character holds its arms horizontally but fatigues after 2 minutes. After a 1-minute rest, it resumes the task, recovering from fatigue.



**Figure 10:** Activation and fatigue curves of the forearm, upper arm, and shoulder muscles in the experiments shown in Figure 9.



**Figure 11:** Fatigue curve of the fatigue and recovery in the run-walk/idle test.



**Figure 12:** Typical learning curves of MuscleVAE using the proposed muscle-space control and the vanilla muscle activation control.