Low Dimensional Motor Skill Learning Using Coactivation

Avinash Ranganath
Clemson University
arangan@clemson.edu

Pei Xu
Clemson University
peix@clemson.edu

Ioannis Karamouzas
Clemson University
ioannis@clemson.edu

Victor Zordan
Clemson University
vbz@clemson.edu

ABSTRACT

We propose an approach for motor skill learning of highly articulated characters based on the systematic exploration of low-dimensional joint coactivation spaces. Through analyzing human motion, we first show that the dimensionality of many motion tasks is much smaller than the full degrees of freedom (DOFs) of the character. Indeed, joint motion appears organized across DOFs, with multiple joints moving together and working in synchrony. We exploit such redundancy for character control by extracting task-specific joint coactivations from human recorded motion, capturing synchronized patterns of simultaneous joint movements that effectively reduce the control space across DOFs. By learning how to excite such coactivations using deep reinforcement learning, we are able to train humanlike controllers using only a small number of dimensions. We demonstrate our approach on a range of motor tasks and show its flexibility against a variety of reward functions, from minimalistic rewards that simply follow the center-of-mass of a reference trajectory to carefully shaped ones that fully track reference characters. In all cases, by learning a 10-dimensional controller on a full 28 DOF character, we reproduce high-fidelity locomotion even in the presence of sparse reward functions.

CCS CONCEPTS

• **Computing methodologies** → **Animation**; *Physical simulation*; *Reinforcement learning*.

KEYWORDS

Motor skill learning, coordination, activation space, reinforcement learning

ACM Reference Format:
Avinash Ranganath, Pei Xu, Ioannis Karamouzas, and Victor Zordan. 2019. Low Dimensional Motor Skill Learning Using Coactivation. In *Motion, Interaction and Games (MIG '19), October 28–30, 2019, Newcastle upon Tyne, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3359566.3360071

1 INTRODUCTION

The problem of learning physics-based skills to control the movement of highly articulated agents has many applications in computer graphics, robotics, and human simulation, in general. The aspiration within video games is that robust, autonomous, physics-based characters will enrich gameplay and lead to greater generalization of characters' interactions in game worlds. In the last few years, impressive results have been obtained for character (and robot) control of 3D bipedal walkers by formulating the problem as Markov decision process and solve it using deep reinforcement learning (DRL) techniques [Peng et al. 2018a; Schulman et al. 2015a,b; Yu et al. 2018]. Nevertheless, state-of-the-art DRL-based articulated systems still lack the intelligence and sophistication seen in real life. The main issue is that controlling many degrees of freedom (DOFs) is inherently ambiguous with respect to most behaviors. While a motor task may be uniquely specified within the action's domain, there is typically an ample space of controllers to accomplish such a task. It is this redundancy in controls that leads to flexibility and adaptation in natural systems. However, it is the same redundancy that creates challenges in learning skills because the control problem is under specified and highly dimensional.

The redundancy in articulated systems has been well studied in different fields including biomechanics, character animation, and robotics (see, e.g., [Chai and Hodgins 2005; d'Avella and Bizzi 2005; Levine et al. 2012; Safonova et al. 2004; Shin and Lee 2006; Shum et al. 2010]), with the literature suggesting that the dimensionality of many motions is much lower than the DOFs of the full action set. In the context of DRL, the full set of DOFs can be detrimental because they allow for poor selection and ambiguity in the behavior controller. Typically, when sophisticated creatures such as humans and animals perform complex tasks, joint motion appears organized across DOFs, e.g. where groupings of joints move together and work in synchrony. Current DRL algorithms do not account for such *coordinated* behavior that stems from the coarticulation of structures, but instead focus on learning a mapping from states to all DOFs – while often they have to deal with competing reward functions. This leads to unnatural, disjoint control strategies, with agents often lacking grace and resourcefulness unless special care is given to shaping the reward function used in training.

In this paper, we propose an alternative DRL approach for motor skill learning that focuses on the systematic exploration of low dimensional activation spaces. To do so, we introduce the concept of *coactivations*, i.e., latent representations that utilize multiple DOFs simultaneously to effectively reduce the control space across the DOFs. By learning how to activate such coordinates, we show that a limited number of dimensions is sufficient to describe motor tasks and this number can vary between different tasks. The learned control policies can reproduce high-fidelity motion that mimics reference data, on par with state-of-the-art DRL techniques such as in [Peng et al. 2018a,b]. Importantly, by exploiting the inherent dimensionality of different behaviors, agents are also capable of learning locomotion skills with simpler reward functions without the need of carefully tuning the character model or closely tracking motion capture examples.

MIG '19, October 28–30, 2019, Newcastle upon Tyne, United Kingdom

ACM ISBN 978-1-4503-6994-7/19/10…$15.00
https://doi.org/10.1145/3359566.3360071

2 RELATED WORK

2.1 Motion Control

The problem of motion control for articulated physical systems has been extensively studied in computer animation and robotics. Typical approaches for synthesizing character motion include manual and data-driven approaches for developing control [Coros et al. 2010; Geijtenbeek et al. 2012; Yin et al. 2007], manifold-learning techniques [Safonova et al. 2004; Shum et al. 2010], optimization-based formulations [Abe et al. 2007; De Lasa et al. 2010; Lee et al. 2010; Liu et al. 2012; Macchietto et al. 2009; Wang et al. 2009], and (deep) neural network-based control [Allen and Faloutsos 2009; Grzeszczuk et al. 1998; Liu and Hodgins 2017, 2018; Peng et al. 2018a, 2016; Yu et al. 2018], among others. In robotics, researchers have also been focusing on reinforcement learning techniques that enable agents to learn motor skills through interactions with the environment [Collins et al. 2005; Kober et al. 2013; Kohl and Stone 2004]. With the recent rise of deep learning, such approaches have gained a lot popularity as agents can learn control policies from raw sensory data. State-of-the-art deep reinforcement learning techniques (DRL) for motor skill development rely on variants of policy gradient methods, such as actor-critic methods that learn a policy network and value function network in tandem [Lillicrap et al. 2015; Mnih et al. 2016; Schulman et al. 2015b], and policy optimization approaches that maximize a surrogate objective function [Schulman et al. 2015a, 2017].

We can classify current approaches for DRL-enabled control of high-dimensional humanoids based on the type of actuation model and the reward function used. Regarding the actuation model, on one hand, there is torque-actuated approaches where the policy network directly outputs torques [Lillicrap et al. 2015; Schulman et al. 2015a; Yu et al. 2018] or maps states to proportional derivative (PD) control targets [Peng et al. 2018a,b], with the latter having shown to be faster to train [Peng and van de Panne 2017]. On the other hand, approaches have also been proposed that focus on the activation level of musculotendon units [Driess et al. 2018; Kidziński et al. 2018; Nakada et al. 2018]. Such approaches lead to more accurate torque patterns at the expense of being more demanding to compute, with recent results enabling control of highly articulated characters and improving performance by transforming the control problem from the muscle-actuation space to the joint-actuation space [Jiang et al. 2019; Lee et al. 2019]. Regarding the cumulative reward that the motion policy seeks to optimize, approaches have been proposed that focus on accurate tracking of motion captured data either by carefully shaping the reward function or by employing adversarial learning [Merel et al. 2017; Peng et al. 2018a]. Others learn policies without the use of reference data or morphology-specific knowledge, e.g., by exploiting low-energetic actions and curriculum learning in a range of environments [Yu et al. 2018] or by using generic reward functions [Duan et al. 2016; Heess et al. 2017] .

2.2 Redundancy in Motor Tasks

Redundant control across many degrees of freedom is inherently ambiguous with respect to most behaviors. In biomechanics, such redundancy is studied as "muscle synergies", defined as low-dimensional modules formed by muscle components coactivated in space and/or time [d'Avella and Bizzi 2005; Torres-Oviedo and Ting 2010]. These synergies can be used by the nervous system as building blocks for constructing motor programs during locomotion and running [Cappellini et al. 2006]. Prior work also identifies consistent spatiotemporal activation patterns in animal trials for tasks such as frog swimming and jumping [Cheung et al. 2005; Tresch et al. 1999], and during human postural tasks [Ting and Macpherson 2005; Torres-Oviedo and Ting 2007]. Overall, these results provide evidence that motor tasks are highly coordinated, and typically only a small number of control signals are needed to accomplish tasks, expressed through the combination of activation inputs.

Researchers have also long been exploring low-dimensional control spaces to develop models of human motion. For example, there has been a lot of work for creating reduced search spaces for manipulation tasks in robotics control and animation (see, e.g., [Andrews and Kry 2013; Ciocarlie et al. 2007]). Safonova et al. [2004] employed principal component analysis on motion capture reference examples and used a reduced basis to synthesize physically plausible motion. Similarly, Chai and Hodgins [2005] reconstructed full-body human motion from local, low-dimensional control signals obtained from a set of motion capture examples, while Shum et al. [2010] used Landmark Isomap to reduce the dimensionality of stable controllers. Other manifold-learning approaches have been investigated to create a low-dimensional latent space of controls from reference data that can be used for motion synthesis, including Gaussian Process Latent Variable Models [Grochow et al. 2004; Levine et al. 2012; Wang et al. 2007], PCA [Shin and Lee 2006], deep autoencoders [Holden et al. 2015], and multiplicative composition primitives [Peng et al. 2019]. Researchers have also applied modal analysis for control reduction, generating behaviors that excite and/or align with the resonant modes of a physical system [Jain and Liu 2011; Kry et al. 2009; Nunes et al. 2012]. Redundancies have also been exploited to create a reduced actuation space for throwing motions [Cruz Ruiz et al. 2017], generating postular responses under perturbations [Ye and Liu 2008], and simulating reliable anatomical features [Lee et al. 2019].

Our work is related to the aforementioned approaches, as we also seek to exploit redundancy in motion control. In particular, similar to approaches that combine low dimensional signals extracted from motion capture data (e.g. [Levine et al. 2012; Safonova et al. 2004]), we also rely on reference data to extract a motion manifold. However, we use this manifold to create a low dimensional control space that an agent can learn to activate in a DRL setting. In that sense, we are complementary to recent DRL techniques that focus on motor skill development, such as [Peng et al. 2018a; Yu et al. 2018]. As opposed to these techniques, though, we plan in an intermediate, low dimensional space, and can support a range of reward objectives (from fully imitating reference data to simply tracking a reference trajectory) while still producing humanlike movement.

3 LEARNING FOR LOW-DIMENSIONAL CONTROL

We formulate motor skill learning as a discounted Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, P, \rho_0, \gamma\}$, where

$\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the action space available to the agent, and $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function which can vary between fully imitating reference motion capture data to simply following a reference trajectory (see Section 6). $P : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the state transition function, ρ_0 is the probability distribution over initial states, and $\gamma \in (0, 1]$ is the discount factor. At a given time step t, the agent interacts with the environment by observing the current state $\mathbf{s}_t \in \mathcal{S}$ and sampling an action $\mathbf{a}_t \in \mathcal{A}$ from a stochastic policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$. This leads to a new state $\mathbf{s}_{t+1}$ according to the dynamics function P that rewards the agent with r_t. The goal of the agent is to maximize the return $R_t = \Sigma_{k=0}^{T-t} \gamma^k r_{t+k}$, which is the total discounted reward starting from time t until a given time horizon T is met or some termination condition is satisfied. We consider parameterized policies $\pi_\theta(\mathbf{a}|\mathbf{s})$, represented as neural networks, and hence the objective of the learning process is to find the optimal set of weights θ^*:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \, \mathbb{E}_{\mathcal{M}, \pi_\theta}[R_{t=0} | \pi_\theta] \tag{1}$$

3.1 Policy Optimization

Policy gradient methods offer a preferred family of algorithms for finding θ^* by directly searching in the policy space [Sutton et al. 2000]. Here, each learning iteration generates samples from the current policy, uses these samples to estimate the gradient of the objective function, and updates the policy parameters with a local search method such as gradient ascent. Unfortunately, classic policy gradient methods suffer from high variance, and are sample inefficient as they are on-policy methods. To address this issue, in this paper, we use the Proximal Policy Optimization (PPO) method by Schulman et al. [2017] which has shown to produce state-of-the-art results in several high-dimensional continuous-action domains. PPO uses a modified objective to the MDP problem that relies on off-policy samples from an older policy $\pi_{\theta_{old}}$ to estimate the expectation of the current policy π_θ:

$$L(\theta) = \mathbb{E}_{\mathbf{s}_t, \mathbf{a}_t \sim \pi_{\theta_{old}}}[\min(g_t(\theta)A_t, clip(g_t(\theta), 1-\epsilon, 1+\epsilon)A_t)], \tag{2}$$

where $g_t = \frac{\pi_\theta(\mathbf{a}|\mathbf{s})}{\pi_{\theta_{old}}(\mathbf{a}|\mathbf{s})}$ is the importance resampling term, ϵ is a tunable hyperparameter that determines how far the new policy can deviate from the old, and A_t is the advantage at time t. In its simplest form, the advantage can be defined as $A_t = R_t - V(\mathbf{s}_t)$, where $V(\mathbf{s}_t) = \mathbb{E}[R_t \mid \pi_{\theta_{old}}; \mathbf{s}_t]$ is the value function approximated as a separate neural network that estimates the expected return starting from state s_t and following the policy $\pi_{\theta_{old}}$. The PPO formulation can handle better the non-stationarity of observation and reward distributions upon collecting samples, and it leads to faster learning performance than other policy gradient methods.

Overall, our implementation largely follows Peng et al. [2018a] with the same network architecture for the policy network and value function network, the same hyperparameters, and PPO approach where the advantage is estimated using a λ-return as in the generalized advantage estimation [Schulman et al. 2015b] (GAE) and the value function is trained with TD(λ) [Sutton and Barto 1998]. See Section 5 for details. The main difference is the output of our policy network, as outlined in the next subsection and further detailed in Section 4, and the type of reward functions considered.

3.2 Coactivation Policies

Previous learning approaches primarily focus on mappings from states to actions based on individual joints, where each action typically represents a set of target angles given as input to a PD-servo or directly specifies torques for each joint [Peng and van de Panne 2017]. Recent work also learns excitations for controlling individual musculotendon units such as in [Driess et al. 2018; Lee et al. 2019]. In contrast, we are interested in learning a mapping from states to excitations of *coactivations*, where each excitation contributes to the control of multiple (indeed all) joints simultaneously.

To do so, we propose to use a set of joint coactivations captured in the form of a *coactivation matrix*, $\mathbf{C} \in \mathbb{R}^{k \times n}$, which describes the set of synchronous joint movements. Here, k denotes the number of coactivations and n is the number of DOFs. Given $\mathbf{C}$, we can specify the control inputs, $\mathbf{u}$, for each individual DOF through a latent representation obtained from the transformation of learnt actions $\mathbf{a}$ (coactivation excitations) with the coactivation matrix $\mathbf{C}$ as: $\mathbf{u} = \mathbf{a}\,\mathbf{C}$. Importantly, because the coactivations in $\mathbf{C}$ control multiple (all) joints, we can reduce the count of coactivations vectors ($k \ll n$) and still control the full DOFs of the character. And so, the goal of our work is to learn policies that output $\mathbf{a}$ in a low dimensional coactivation space. In our framework, the control input is then mapped to target orientations for each joint angle that are then passed to an SPD servo [Tan et al. 2011] to generate torques.

We discuss next how to construct coactivation matrices from motion capture examples. Our training procedure is detailed in Section 5.

4 LOW-DIMENSIONAL JOINT COACTIVATION

Motivated by dimensionality reduction methods in human motion analysis (e.g., [Chai and Hodgins 2005; Safonova et al. 2004; Ye and Liu 2008]) and the muscle synergies seen in biomechanics literature (e.g., [d'Avella et al. 2006; Torres-Oviedo and Ting 2007]), we aim to produce a coactivation matrix that is both lower dimensional than the full DOFs of the character, but also helpful in producing coordination like that seen in human motion. With the goal of learning in the space of coactivation signals, we propose to precompute a set of joint coactivations, assembled in matrix $\mathbf{C}$, from reference motion data of the desired behavior. Further, by choosing a number of coactivations, k, to represent n DOFs such that $k < n$, we produce low-dimensional embedding $\mathbf{C} \in \mathbb{R}^{k \times n}$.

To motivate this choice, let $\mathbf{X} \in \mathbb{R}^{m \times n}$ denote a time series of human poses for a specific motion task, with m being the number of frames and n the number of DOFs. Although the data in $\mathbf{X}$ span the n-dimensional space, research shows that human motion is highly coordinated and so the data lies on a much lower-dimensional sub-space ($\mathbb{R}^{m \times k}$, $k \leq m$), embedded within the n-dimensional full-space. By extracting the structure of this lower-dimensional subspace, we find a set of latent dimensions where joints move together, dependent on the particular motion example. We propose to represent our joint coactivation as the latent variables of this low-dimensional subspace. Additional motivation for this choice appears in Section 4.2 following our process for extracting $\mathbf{C}$ from reference motion.

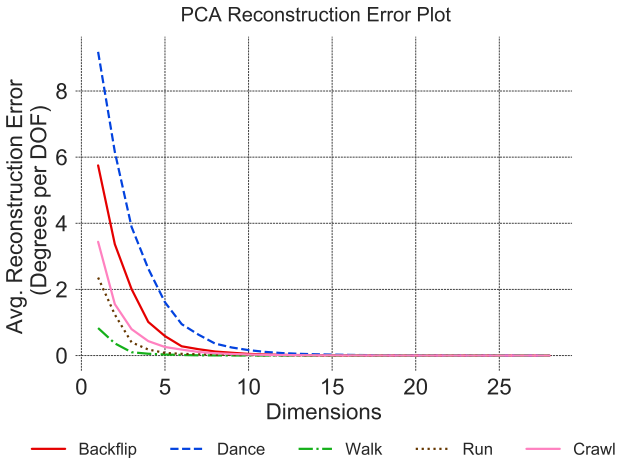

Figure 1: PCA reconstruction error for different behaviors. Using only the first 3 principal dimensions, the average reconstruction error is less than 1° for walking, running, and crawling, highlighting the control redundancy in locomotion tasks. Similar conclusions can be drawn for backflip and dance behaviors when using the top 6 principal dimensions.

4.1 Coactivation Extractions

Our coactivation matrix C, representing the structure of the underlying lower-dimensional sub-space, can be extracted from any reference motion using dimensionality reduction techniques. We looked at two simple methods, principle component analysis (PCA) and independent component analysis (ICA) to accomplish this process. In PCA, using singular value decomposition, a human pose data matrix $\mathbf{X}$ can be decomposed as $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$. Within this decomposition, $\Sigma\mathbf{V}^\top$, the right-singular matrix scaled by the singular values, would represent the *coactivation* matrix $\mathbf{C}$, and $\mathbf{U}$, the left singular matrix, would represent the "excitation" of the coactivations for each frame. Similarly, using ICA, the human pose data matrix $\mathbf{X}$ can be decomposed as, $\mathbf{X} = \mathbf{SW}$, where the mixing matrix, $\mathbf{W}$, would represent the *coactivation* matrix, and $\mathbf{S}$, the set of source signals, represent the excitation values of the coactivations for each frame.

After empirical testing of both PCA and ICA, we opted to select PCA as our primary extraction process for coactivation. We note both processes worked effectively for our task, likely indicating that the technique is not very sensitive to the precise choice of coactivations. Further detail on this topic is added in the discussion in Section 7.

4.2 Dimensionality Reduction

The motion capture data used in this work comes from two publicly available datasets [CMU 2019; SFU 2019]. For each motor behavior, the data consists of a single cycle of the behavior, and the number of data frames ranges between 25 and 177. 3D joints such as shoulder joints, hip joints and ankle joints are modeled as quaternions, with hinge joints for the elbows and knees. Because PCA assumes that the data values are independent, which is broken when joints are modeled as quaternions, we convert all 3D joints to Euler angles when we perform PCA. After the conversion, each reference clip has 28 DOFs, which we normalize to have zero mean.

Let $\mathbf{X} \in \mathbb{R}^{m\times n}$ denote one of the processed $n-$dimension motion clips consisting of m frames, so that $\mathbf{x}^{(i)} \in \mathbb{R}^n$ for each frame i. By applying PCA on $\mathbf{X}$, the n-dimension set of motion can be projected on to a k-dimension coactivation subspace by choosing the top-k principle components. Then, given the set of coactivation excitation values, $\mathbf{U} \in \mathbb{R}^{m\times k}$, we can reconstruct the original motion as: $\widetilde{X} = \mathbf{UC}$. Figure 1 shows the reconstruction error of different motion clips while varying the number of principal dimensions. As can be seen from the plots, the reconstruction error drops below 2° per joint within the top 5 dimensions for all the behaviors, and within the top 3 dimensions for the behaviors of running, walking, and crawling. In all behaviors, using just 10 dimensions rather than the full set of 28 is sufficient for almost perfect reconstruction.

Along with minimizing reconstruction error, we can observe that the biggest contributors to the movements (largest components) reveal *coordinated actions* in the movement of the joints. Namely, the top components move the body in synchronized ways that reveal symmetry and coordinated displacements of the end effectors (among other notable characteristics). See Figure 2 and the accompanying video for further details. This set of observations motivated our choice of using the latent variables obtained with PCA as coactivations for joint control.

5 GENERATING CONTROLS

As our implementation draws from the work of [Peng et al. 2018a], we refer readers to their paper for additional detail. The core difference is that the outputs of our policy network are coactivation excitation values rather than target values (for PD control). We also make explicit augmentations to the reward function for training controllers, as outlined in Section 6.

5.1 State, Action and Architecture

The state, $\mathbf{s}$, consists of the relative position of each joint to that of the root of the character, along with the joints' rotations, velocities, and angular velocities. The angles of 3D joints are represented as quaternions, while hinge (1D) joints are scalar rotation angles. Included in the state descriptor is also a normalized phase parameter, ϕ, based on the phase of the reference motion.

The control architecture implemented in our framework is illustrated in Figure 3. The policy network consists of two hidden layers of 1024 and 512 neurons, respectively. Rectifier Linear Unit (ReLU) activation functions are used in both hidden layer neurons, and the output neurons are obtained using a linear activation function. The policy network outputs the mean, $\mu(\mathbf{s})$, of a multivariate Gaussian distribution which is subsequently used to sample excitation actions $a \in \mathbb{R}^k$, where k is the number of coactivation dimensions. The covariance of the Gaussian is represented by a fixed diagonal matrix $\Sigma = diag(\sigma_1, \ldots, \sigma_k)$, where σ_i denote the variance of the i^{th} action. The sampled excitations are then transformed into a control input as $u = a\mathbf{C}$. Torque, F, is generated by a PD-servo controller for each joint, and applied to the simulated character. Note that our coactivation matrix $\mathbf{C}$, for a specified dimension k, is

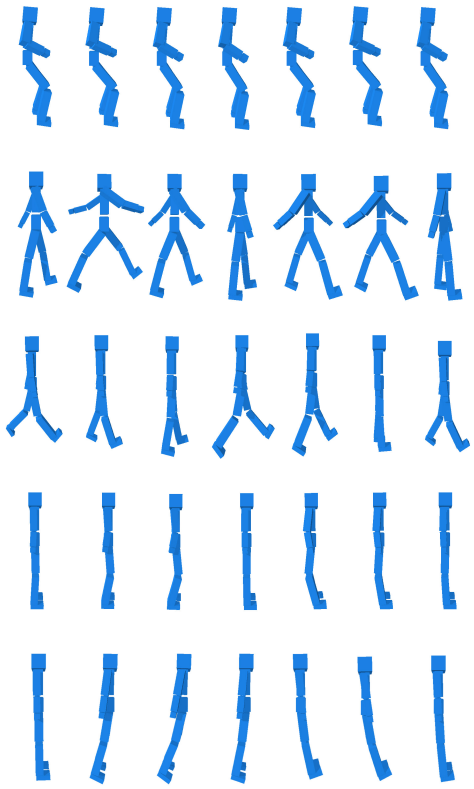

Figure 2: Excitations of the top-5 components extracted from running motion clip. Each row above shows time-lapses of exciting each of the top-5 components individually, shedding light on the underlying low-level coordinations in bipedal walking.

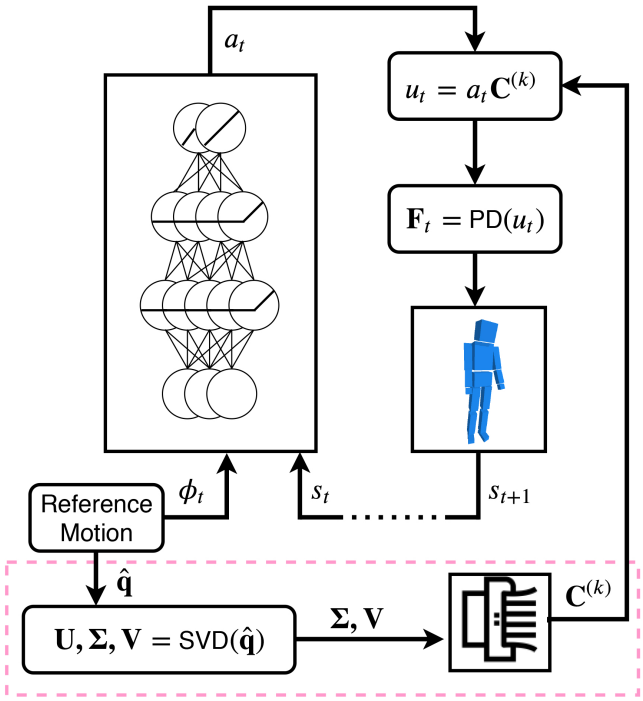

Figure 3: Overview of our learning framework. A reference motion clip is used to construct offline a coactivation matrix **C**. During training, a policy is learned that maps states to actions, **a**, that excite the coactivations. The resulting control input, **u**, is mapped to target angles that are given as input to a stable PD controller to generate joint torques, **F**.

pre-computed offline prior to policy training. During training, **C** is held constant.

5.2 Training

We use PPO [Schulman et al. 2017] for training the control policies and learn a policy network $\pi_\theta(\mathbf{a}|\mathbf{s})$ and a value network $V_\psi(\mathbf{s})$, parameterized by θ and ψ respectively. The training is performed episodically, by sampling an initial state from the reference motion, and then sampling actions from the policy network there after, generating rollouts for updating the policy and value networks. An episode terminates either after a fixed period of 20 seconds, or when certain links of the character make contact with the ground. Minibatches of size 256 are sampled from the collected data to update the networks, where the value function is trained using TD (λ) and the policy gradient is updated using the clipped surrogate objective (Equation 2) with a GAE(λ) advantage estimate. Common learning parameters values, consisting of a discount factor of $\gamma = 0.95$, $\lambda = 0.95$ for TD(λ) and GAE(λ), a likelihood ratio clipping threshold of $\epsilon = 0.2$, a fixed diagonal covariance matrix $\Sigma = I \times 5^{-2}$, policy step size of $\alpha_\pi = 5^{-4}$ and value function step size of $\alpha_v = 10^{-2}$, are used across all trained behaviors.

6 REWARD FUNCTIONS

The reward function, r, that guides the learning process of the agent during training is one of the most important components for solving the underlying MDP. In its simplest form, r, could simply be an indicator function for task completion. In practice, though, carefully shaped reward functions are needed to learn robust and humanlike motor skills, typically formed as a weighted sum of different objective terms. In our framework, we start with the reward function proposed in [Peng et al. 2018a] that encourages the agent to be as close as possible to the reference motion. In addition, to test the sensitivity of learning under different reward functions, we truncate this reward in two successive steps. We designated these three rewards as: *imitation* reward (IR), *end-effector* reward (ER), and *center-of-mass* reward (CR). We define the specifics of each next.

6.1 Imitation Reward (IR)

For learning in the low-dimensional space, the agent is encouraged to follow the reference motion by embedding imitation components in the IR function, which is defined as:

$$r_t^I = w_I^p r_t^p + w_I^v r_t^v + w_I^e r_t^e + w_I^c r_t^c + w_I^r r_t^r. \quad (3)$$

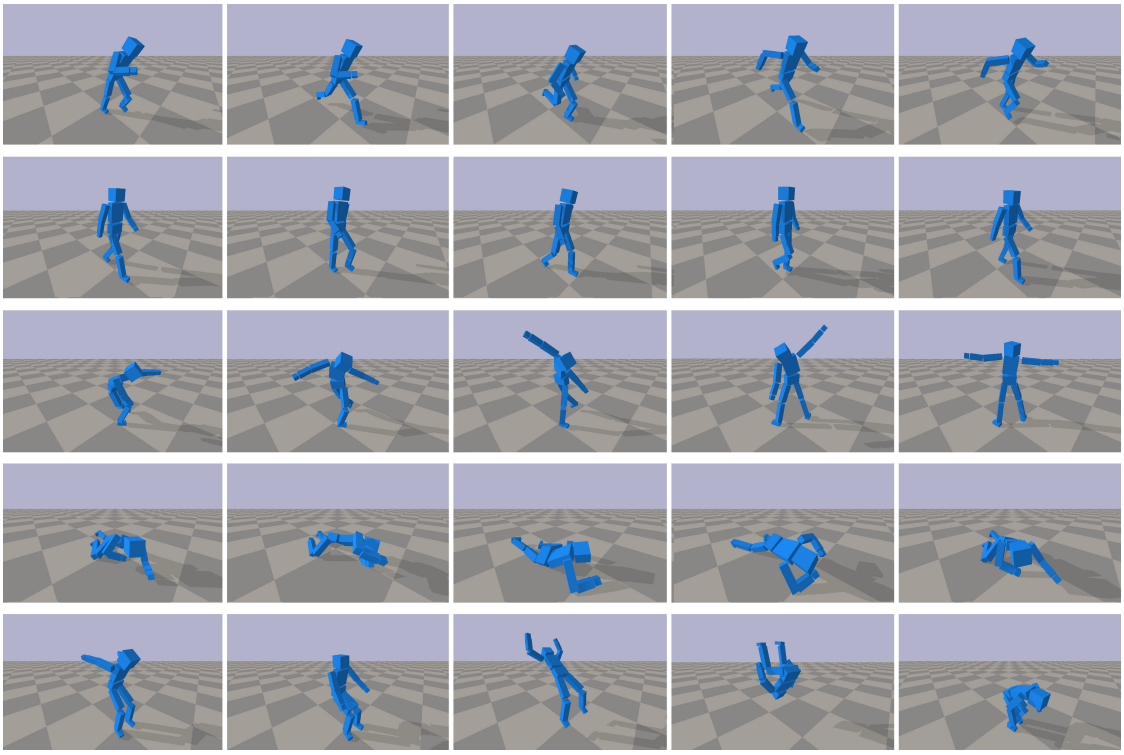

Figure 4: Filmstrips of motor skills controlled by 10-dimensional controllers. Snapshots of a single cycle of trained behaviors of (top to bottom) run, walk, dance, crawl and backflip motor skills.

The pose tracking term, $r_t^p = exp(-\omega\Sigma_j\|\hat{q}_t \ominus q_t\|^2)$, encourages the agent to match the joint orientation in the reference data, minimizing the difference between the joint angles present in the reference motion $\hat{q}_t$, and that of the simulated character q_t by using the quaternion difference operator $\ominus$. The velocity tracking term, $r_t^v = exp(-\omega\Sigma_j\|\hat{\dot{q}}_t - \dot{q}_t\|^2)$, encourages the agent to match the angular velocities of the joints with the reference motion. Next, $r_t^e = exp(-\omega\Sigma_e\|\hat{p}_t - p_t\|^2)$ encourages the agent to follow the trajectory of the end effectors, $e \in$ { left hand, right hand, left foot, right foot }. Similarly, $r_t^c = exp(-\omega\|\hat{\dot{p}}_t - \dot{p}_t\|^2)$ encourages the agent to follow the velocity of the center of mass (COM) from the reference motion. Finally, the r_t^r term promotes tracking of the position, velocity, orientation, and angular velocity of root body from the reference motion, as follows:

$$r_t^r = exp(-\omega(\Delta p_t^r + 10^{-1}\Delta\dot{p}_t^r + 10^{-2}\Delta q_t^r + 10^{-3}\Delta\dot{q}_t^r)), \quad (4)$$

where Δ is difference of the value to the analogous reference data for each term. Note this reward function exactly duplicates that which is present in the code implementation of Peng et al.'s [2018a] work. We purposefully do not change this reward to enable comparisons, however we note that this includes a total of eight terms and more than a dozen weight values, which represents a substantial amount of reward shaping to achieve the spectacular results we find in [Peng et al. 2018a].

6.2 End-effector Reward (ER)

Since we expect that the coordination needed for the target behavior comes from reference motion in the form of *coactivations*, we hypothesize that a behavior could be trained without having to imitate the reference motion explicitly. To test this, we created two more reward functions, ER and CR, as a subset of the original reward function in Equation 3. The reward function for ER removes the explicit joint tracking terms, but maintains the information related to the end effectors in the reference motion, as

$$r_t^E = w_E^e r_t^e + w_E^c r_t^c + w_E^r r_t^r. \quad (5)$$

6.3 COM Reward (CR)

The CR reward encourages the agent to follow the COM and root body trajectories derived from the reference motion, but removes all other behavior specific terms from the reward. As such,

$$r_t^C = w_C^c r_t^c + w_C^r r_t^r. \quad (6)$$

Note this final reward has similarity, in nature and content, to the reward proposed by [Yu et al. 2018] with the exception that the terms for the COM and root body still contain (limited) data extracted from the reference motion and, in our case, there is no explicit term to promote minimal actuation (the symmetric action term added to the PPO objective by the authors is implicitly upheld in our coactivation matrix). We uphold the structure presented to maintain consistency between all three of own reward functions, allowing us

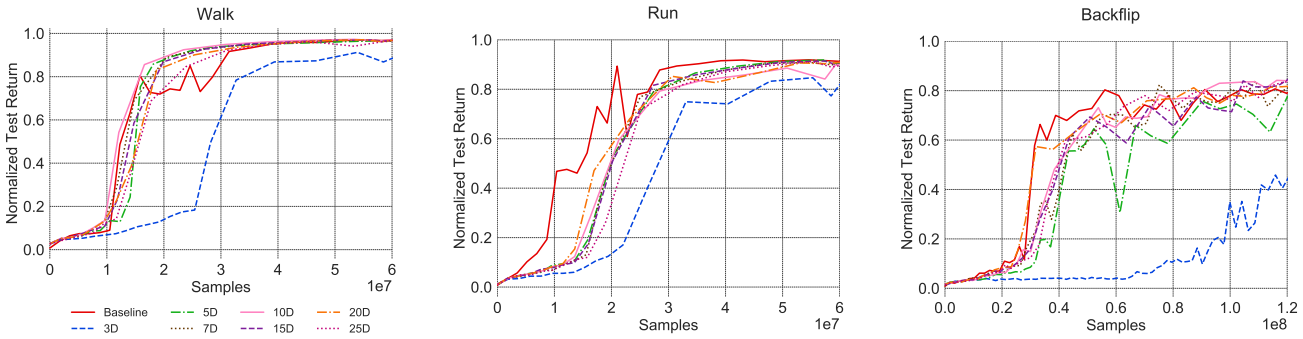

Figure 5: Learning curves obtained with various coactivation dimensions and the baseline for different tasks using the imitation reward. In all cases, motor skills can be learned at lower dimensions, with high fidelity locomotion obtained with as low as 5 dimensions, and backflip with 7 dimensions.

to make more clear inference pertaining to the differences observed between them.

All weight values (w's) for Equations 3 and 4 as well as their included terms' weightings (ω's) are taken from [Peng et al. 2018a]. The weights used for ER and CR reward functions are $w_E^e = w_E^c = 0.4$ and $w_E^r = 0.2$, and $w_C^c = 0.8$ and $w_C^r = 0.2$, respectively.

7 RESULTS AND ANALYSIS

We use the physical 3D humanoid character shown in Figure 3 to train all of our policies. We focused on five behaviors in total: run, walk, backflip, crawl and dance. Filmstrips of each appear in Figure 4. For evaluating the effects of training in the reduced dimension space, we also trained the five behaviors in the original independent joint action space, as [Peng et al. 2018a], and use the resulting policies as *baseline* for each behavior. Learned behaviors are best seen in the companion video.

7.1 Task Dimension

We trained a set of motor skills (walk, run, backflip) over a wide range of reduced coactivation dimensions using the IR reward function that aims to fully track the reference motion (Equation 3). From this result we can distinguish the impact of dimension on the learning performance. The learning curves for each are shown in Figure 5. Comparisons with baseline motions appear in the video.

It could be observed from Figure 5 that: (i) motor skills can be learned at much lower dimensions, and still match the performance of the baseline; and (ii) that there is a lower bound on the minimum number of dimensions needed to learn any skill. The plots show that for running and walking, performance close to that of the baseline can be achieved at around 5 dimensions, and 7 dimensions for backflip. Adding further dimensions contribute very little to the performance. These results are consistent with findings from Figure 1 indicating the dimension required for faithful reconstruction following PCA decomposition.

Figure 6 compares performance of low dimensional learning using coactivations derived from PCA and ICA. Notably, the learner did not favor one technique over the other in overall performance measures, although some qualitative differences appear in the motion. Comparisons of the two along with a comparison of the two

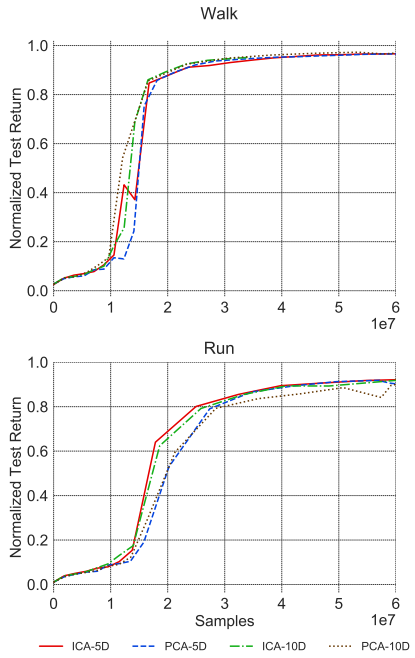

Figure 6: Learning curves comparison between PCA and ICA. Walking and running tasks trained with 5D and 10D coactivations show similar performance between the two methods, though ICA converges faster for low dimensions.

with respect to a randomly generated coactivation matrix appear in the video. The highlight related to the latter is that while the learner did not show preference over ICA and PCA, it fails to learn a successful policy with a random coactivation.

7.2 Reward Sensitivity

We hypothesize that training a policy in the coactivation space is more robust to different reward functions as compared to the baseline that uses independent joint activation, as the coactivation injects task-specific knowledge into the control routine. To test

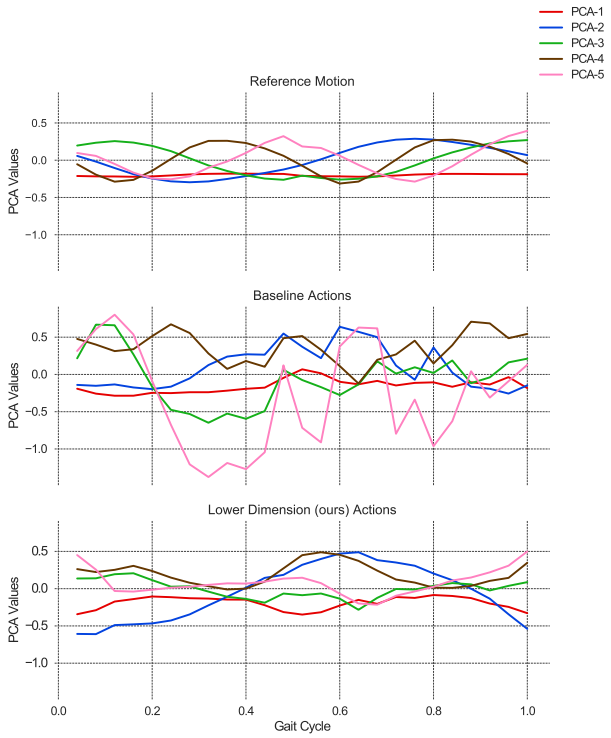

Figure 7: Action plots of running task from reference motion clip, baseline, and coactivation. (Top) Excitations, **U**, extracted from running motion clip. (Middle) Output of the baseline policy network projected to the coactivation space by multiplying it by the inverse of the coactivation matrix, $\mathbf{C}^{-1}$. (Bottom) Policy network output of a low dimension (10D) controller. Both the baseline and the 10D versions were trained with the sparse reward function CR, but only the low-D controller generates actions that closely match the ones performed by the human subject.

this hypothesis, we trained on two successively minimal reward functions, ER and CR respectively, and compare them qualitatively (in the video) and quantitatively in Table 1 to the original (full) reward, IR. To compare, we test the learned policies for 20 episodes with a fixed length of 20s each, and evaluate the motion against the best possible IR value that can be obtained. Since the reward function IR, strongly focuses on imitating, one assessment of the "humanlike-ness" of the output is achieved by comparing the learnt policy of ER and CR against the IR metric. In general, if the policy produces a behavior that does well with the trained reward function, but does not imitate the behaviors (e.g. a walking behavior that performs flipping to locomote), the output policy performs poorly in this comparison.

We can make several observations from this experiment. Notably, for the baseline, all three reward function work for running, but the resulting motion is less humanlike for ER and particularly CR as indicated by their low scores. For baseline walking, the character fails to learn a humanlike gait and instead just follows the COM by

Table 1: Evaluation of different models on walking and running motion using the imitation (IR), end-effector (ER), and COM (CR) reward functions. The baseline character fails to walk as a human when trained with rewards that lack joint tracking (ER, CR). Similar behavior is observed for running under CR. In contrast, using between just 5 and 10 coactivation dimensions, humanlike locomotion controllers are obtained even when only the COM of reference data is tracked. Reported numbers denote the maximum return over 20 testing episodes normalized based on the best possible reward that the character would have gotten if it perfectly follows the reference data. * denotes failure to learn the intended behavior.

Model	Walk			Run		
	IR	ER	CR	IR	ER	CR
Baseline	0.98	*(0.41)	*(0.32)	0.92	0.81	0.55
Coactivation 3D	0.88	*(0.30)	0.67	0.79	0.78	*(0.09)
Coactivation 5D	**0.96**	**0.96**	**0.72**	**0.89**	**0.85**	**0.74**
Coactivation 10D	**0.96**	**0.97**	**0.74**	**0.94**	**0.89**	**0.75**
Coactivation 15D	0.98	0.97	0.75	0.94	0.87	0.54
Coactivation 20D	0.98	0.97	*(0.13)	0.89	0.88	0.59
Coactivation 25D	0.97	0.97	*(0.33)	0.94	0.85	0.72

performing forward flips (see video). This is due to the fact that with lack of pose tracking, the control space is highly under-constrained, leading to many locally optimal solutions - but no guarantee of a humanlike gait. For coactivation, we observe learnt walking and running is more successful at imitation using a controller between 5D and 10D than other dimensions. As the number of dimensions increases past 10, the controls start approximating the baseline, resulting in running motions that lack grace and walking that fails to learn the intended motion. As the dimension of the controller is reduced (above the critical threshold as described in Section 7.1), the constraints placed by the coactivations help to create motions that better imitate (under IR) human motion, in comparison to the baseline.

The suitability of the coactivation space for motor control training is further highlighted in Figure 7. Here, we compare the excitation values of joint coactivations of a human runner to the ones obtained by a 10D coactivation controller and a baseline controller trained using the CR function. For the 10D controller, the reported values are simply the output of the policy network, while for the baseline controller the output of the policy network is first projected to the coactivation space. As can be observed by the figure, training policies in the latent space rather than directly in the full control space leads to actions that more closely follow joint excitation values of real humans. Note that when agents are trained with the full imitation reward, IR, both coactivation-based controllers and baseline ones were able to closely match the human excitations. This is because, to perfectly mimic running motion, the agent eventually will have to learn how to properly coordinate its joint movements similar to a real runner.

8 CONCLUSION

In this paper, we present a DRL approach for motor skill learning of highly articulated characters using the concept of coactivations. We exploit redundancy in character control by extracting a task-specific embedding from motion data, where the latent encoding specifies sychronized patterns of simultaneous joint movements. By learning excitations for such coactivations, we show that the required control dimensionality of many motions is much smaller that the full DOFs of the character. Further, for each task, there is also a dimension below which the character fails to learn, and above which a policy can always be trained. We test our approach on a range of motor skills and show its flexibility against a set of reward functions, from minimalistic rewards that simply follow the center of mass of a reference motion, to more carefully shaped ones that track end effector and joint angle reference data precisely. We show that the resulting policies are significantly influenced by both the type of reward used and the number of latent dimensions as well as the interplay of both. An important finding is that maximizing a carefully shaped reward with too many dimensions can capture unimportant variations of the motion, while a minimalistic reward with too many dimensions results in many local minima and solutions that deviate significantly from humanlike behavior. In total, our findings point to there being an inherent (low) dimensionality to different motions, and further we show how this can be exploited for motor skill learning in a DRL framework.

Our approach has a number of limitations that we would like to address in the future. Even though the agent learns to activate a lower dimensional space than the full action space, this does not always translate to sample efficient training and faster learning times. We hypothesize that the main bottleneck may be that the agent still has to learn the dynamics of the environment. As well, the PD-servo regulating the output of the policy network and the final torques applied to the agent adds another layer of training complexity. As such, we would like to investigate latent representations directly in torque space. Currently, we extract coactivations from a single reference clip per motion task. Even though we have shown the robustness of such a latent space, in the future, we want to generate coactivations from multiple motion capture trials and compare the resulting spaces to the ones obtained here. We also want to look into different control reduction approaches such as modal decomposition [Kry et al. 2009; Nunes et al. 2012] as an alternative to extract joint coordination basis. Another avenue for future research is to construct coactivations by combining different, but related, motor tasks such as walking and running. We believe that this will lead to policies that can generalize more easily, allowing composition of skills that can be transferred in other tasks. In this setting, we may benefit from learning the latent encoding online rather than build a static one offline as we currently do. The recent works of [Hausman et al. 2018; Peng et al. 2019] provide interesting insights towards this direction. Finally, in this paper, we showed applicability of coactivations in mainly locomotion tasks. However, most motor tasks performed by humans involve some sort of spatiotemporal coherence and coordination, such as, e.g., dancing or swinging a golf club [Aristidou et al. 2018]. As such, we plan to test our underlying system on a wider range of tasks and investigate simple reward objectives that can reproduce humanlike skills.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their effort and input. This work was partly funded by the Clemson University Fellows Program.

REFERENCES

Yeuhi Abe, Marco Da Silva, and Jovan Popović. 2007. Multiobjective control with frictional contacts. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 249–258.

Brian F Allen and Petros Faloutsos. 2009. Evolved controllers for simulated locomotion. In *International Workshop on Motion in Games*. Springer, 219–230.

Sheldon Andrews and Paul G Kry. 2013. Goal directed multi-finger manipulation: Control policies and analysis. *Computers & Graphics* 37, 7 (2013), 830–839.

Andreas Aristidou, Daniel Cohen-Or, Jessica K. Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. 2018. Deep Motifs and Motion Signatures. *ACM Transactions on Graphics* 37, 6 (Dec. 2018), 187:1–187:13.

Germana Cappellini, Yuri P Ivanenko, Richard E Poppele, and Francesco Lacquaniti. 2006. Motor patterns in human walking and running. *Journal of neurophysiology* 95, 6 (2006), 3426–3437.

Jinxiang Chai and Jessica K Hodgins. 2005. Performance animation from low-dimensional control signals. *ACM Transactions on Graphics* 24, 3 (2005), 686–696.

Vincent CK Cheung, Andrea d'Avella, Matthew C Tresch, and Emilio Bizzi. 2005. Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors. *Journal of Neuroscience* 25, 27 (2005), 6419–6434.

Matei Ciocarlie, Corey Goldfeder, and Peter K Allen. 2007. Dimensionality reduction for hand-independent dexterous robotic grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 3270–3275.

CMU. 2019. Carnegie Mellon University MoCap Database. (2019). http://mocap.cs.cmu.edu/

Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. 2005. Efficient bipedal robots based on passive-dynamic walkers. *Science* 307, 5712 (2005), 1082–1085.

Stelian Coros, Philippe Beaudoin, and Michiel Van de Panne. 2010. Generalized biped walking control. *ACM Transactions on Graphics* 29, 4 (2010), 130.

Ana Lucia Cruz Ruiz, Charles Pontonnier, Jonathan Levy, and Georges Dumont. 2017. A synergy-based control solution for overactuated characters: Application to throwing. *Computer Animation and Virtual Worlds* 28, 6 (2017), e1743.

Andrea d'Avella and Emilio Bizzi. 2005. Shared and specific muscle synergies in natural motor behaviors. *Proceedings of the National Academy of Sciences* 102, 8 (2005), 3076–3081.

Andrea d'Avella, Alessandro Portone, Laure Fernandez, and Francesco Lacquaniti. 2006. Control of fast-reaching movements by muscle synergy combinations. *Journal of Neuroscience* 26, 30 (2006), 7791–7810.

Martin De Lasa, Igor Mordatch, and Aaron Hertzmann. 2010. Feature-based locomotion controllers. In *ACM Transactions on Graphics*, Vol. 29. ACM, 131.

Danny Driess, Heiko Zimmermann, Simon Wolfen, Dan Suissa, Daniel Haeufle, Daniel Hennes, Marc Toussaint, and Syn Schmitt. 2018. Learning to Control Redundant Musculoskeletal Systems with Neural Networks and SQP: Exploiting Muscle Properties. In *IEEE International Conference on Robotics and Automation*. IEEE, 6461–6468.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. 2016. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*. 1329–1338.

Thomas Geijtenbeek, Nicolas Pronost, and A Frank van der Stappen. 2012. Simple data-driven control for simulated bipeds. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, Goslar Germany, Germany, 211–219.

Keith Grochow, Steven L Martin, Aaron Hertzmann, and Zoran Popović. 2004. Style-based inverse kinematics. *ACM transactions on graphics* 23, 3 (2004), 522–531.

Radek Grzeszczuk, Demetri Terzopoulos, and Geoffrey Hinton. 1998. NeuroAnimator: fast neural network emulation and control of physics-based models.. In *Proceedings of SIGGRAPH 98*. ACM, New York, NY, USA, 9–âĂŞ20.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. 2018. Learning an Embedding Space for Transferable Robot Skills. In *International Conference on Learning Representations*.

Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, Martin Riedmiller, et al. 2017. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286* (2017).

Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning Motion Manifolds with Convolutional Autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*. ACM, New York, NY, USA, Article 18, 4 pages.

Sumit Jain and C Karen Liu. 2011. Modal-space control for articulated characters. *ACM Transactions on Graphics* 30, 5 (2011), 118.

Yifeng Jiang, Tom Van Wouwe, Friedl De Groote, and C Karen Liu. 2019. Synthesis of Biologically Realistic Human Motion Using Joint Torque Actuation. *arXiv preprint arXiv:1904.13041* (2019).

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

Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274.

Nate Kohl and Peter Stone. 2004. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation*. IEEE, 2619–2624.

Paul G Kry, Lionel Revéret, François Faure, and M-P Cani. 2009. Modal locomotion: Animating virtual characters with natural vibrations. *Computer Graphics Forum* 28, 2 (2009), 289–298.

Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. 2019. Scalable Muscle-Actuated Human Simulation and Control. *ACM Transactions on Graphics* (2019).

Yoonsang Lee, Sungeun Kim, and Jehee Lee. 2010. Data-driven biped control. *ACM Transactions on Graphics* 29, 4 (2010), 129.

Sergey Levine, Jack M Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun. 2012. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics* 31, 4 (2012), 28.

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

Libin Liu and Jessica Hodgins. 2017. Learning to Schedule Control Fragments for Physics-Based Characters Using Deep Q-Learning. *ACM Transactions on Graphics* 36, 3 (2017).

Libin Liu and Jessica Hodgins. 2018. Learning Basketball Dribbling Skills Using Trajectory Optimization and Deep Reinforcement Learning. *ACM Transactions on Graphics* 37, 4 (2018).

Libin Liu, KangKang Yin, Michiel van de Panne, and Baining Guo. 2012. Terrain runner: control, parameterization, composition, and planning for highly dynamic motions. *ACM Transactions on Graphics* 31, 6 (2012), 154–1.

Adriano Macchietto, Victor Zordan, and Christian R. Shelton. 2009. Momentum control for balance. *ACM Transactions on Graphics* 28, 3 (2009), Article 80.

Josh Merel, Yuval Tassa, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess. 2017. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201* (2017).

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.), Vol. 48. PMLR, New York, New York, USA, 1928–1937.

Masaki Nakada, Tao Zhou, Honglin Chen, Tomer Weiss, and Demetri Terzopoulos. 2018. Deep learning of biomimetic sensorimotor control for biomechanical human animation. *ACM Transactions on Graphics* 37, 4 (2018), 56.

Rubens F. Nunes, Joaquim B. Cavalcante-Neto, Creto A. Vidal, Paul G. Kry, and Victor B. Zordan. 2012. Using natural vibrations to guide control for locomotion. In *ACM Symposium on Interactive 3D Graphics and Games*. 87–94.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018a. DeepMimic: Example-guided Deep Reinforcement Learning of Physics-based Character Skills. *ACM Transactions on Graphics* 37, 4 (2018), 143:1–143:14.

Xue Bin Peng, Glen Berseth, and Michiel Van de Panne. 2016. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics* 35, 4 (2016), 81.

Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. 2019. MCP: Learning Composable Hierarchical Control with Multiplicative Compositional Policies. *arXiv preprint arXiv:1905.09808* (2019).

Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. 2018b. SFV: Reinforcement Learning of Physical Skills from Videos. *ACM Transactions on Graphics* 37, 6 (2018), 178:1–178:14.

Xue Bin Peng and Michiel van de Panne. 2017. Learning locomotion skills using deeprl: does the choice of action space matter?. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, New York, NY, USA, 12:1–12:13.

Alla Safonova, Jessica K Hodgins, and Nancy S Pollard. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics* 23, 3 (2004), 514–521.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015a. Trust region policy optimization. In *International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 1889–1897.

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

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

SFU. 2019. Simon Fraser University Motion Capture Database. (2019). http://mocap.cs.sfu.ca/

Hyun Joon Shin and Jehee Lee. 2006. Motion synthesis and editing in low-dimensional spaces. *Computer Animation and Virtual Worlds* 17, 3-4 (2006), 219–227.

Hubert PH Shum, Taku Komura, Takaaki Shiratori, and Shu Takagi. 2010. Physically-based character control in low dimensional space. In *International Conference on Motion in Games*. Springer, 23–34.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction* (1st ed.). MIT press, Cambridge, MA, USA.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*. 1057–1063.

Jie Tan, Karen Liu, and Greg Turk. 2011. Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications* 31, 4 (2011), 34–44.

Lena H Ting and Jane M Macpherson. 2005. A limited set of muscle synergies for force control during a postural task. *Journal of neurophysiology* 93, 1 (2005), 609–613.

Gelsy Torres-Oviedo and Lena H Ting. 2007. Muscle synergies characterizing human postural responses. *Journal of neurophysiology* 98, 4 (2007), 2144–2156.

Gelsy Torres-Oviedo and Lena H Ting. 2010. Subject-specific muscle synergies in human balance control are consistent across different biomechanical contexts. *Journal of neurophysiology* 103, 6 (2010), 3084–3098.

Matthew C Tresch, Philippe Saltiel, and Emilio Bizzi. 1999. The construction of movement by the spinal cord. *Nature neuroscience* 2, 2 (1999), 162.

Jack M Wang, David J Fleet, and Aaron Hertzmann. 2007. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence* 30, 2 (2007), 283–298.

Jack M Wang, David J Fleet, and Aaron Hertzmann. 2009. Optimizing walking controllers. *ACM Transactions on Graphics* 28, 5 (2009), 168.

Yuting Ye and C Karen Liu. 2008. Animating responsive characters with dynamic constraints in near-unactuated coordinates. *ACM Transactions on Graphics* 27, 5 (2008), 112.

KangKang Yin, Kevin Loken, and Michiel Van de Panne. 2007. Simbicon: Simple biped locomotion control. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 105.

Wenhao Yu, Greg Turk, and C Karen Liu. 2018. Learning symmetric and low-energy locomotion. *ACM Transactions on Graphics* 37, 4 (2018), 144.