Motor Babble: Morphology-Driven Coordinated Control of Articulated Characters

Avinash Ranganath
Clemson University
Clemson, SC, USA
arangan@clemson.edu

Avishek Biswas
Clemson University
Clemson, SC, USA
avisheb@clemson.edu

Ioannis Karamouzas
Clemson University
Clemson, SC, USA
ioannis@clemson.edu

Victor B. Zordan
Clemson University
Clemson, SC, USA
vbz@clemson.edu

ABSTRACT

Locomotion in humans and animals is highly coordinated, with many joints moving together. Learning similar coordinated locomotion in articulated virtual characters, in the absence of reference motion data, is a challenging task due to the high number of degrees of freedom and the redundancy that comes with it. In this paper, we present a method for learning locomotion for virtual characters in a low dimensional latent space which defines how different joints move together. We introduce a technique called motor babble, wherein a character interacts with its environment by actuating its joints through uncoordinated, low-level (motor) excitations, resulting in a corpus of motion data from which a manifold latent space is extracted. Dimensions of the extracted manifold define a wide variety of synergies pertaining to the character and, through reinforcement learning, we train the character to learn locomotion in the latent space by selecting a small set of appropriate latent dimensions, along with learning the corresponding policy.

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**; *Learning latent representations*; *Reinforcement learning*.

KEYWORDS

character animation, physics-based control, reinforcement learning, animal locomotion

ACM Reference Format:
Avinash Ranganath, Avishek Biswas, Ioannis Karamouzas, and Victor B. Zordan. 2021. Motor Babble: Morphology-Driven Coordinated Control of Articulated Characters. In *Motion, Interaction and Games (MIG '21), November 10–12, 2021, Virtual Event, Switzerland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3487983.3488291

MIG '21, November 10–12, 2021, Virtual Event, Switzerland

ACM ISBN 978-1-4503-9131-3/21/11…$15.00
https://doi.org/10.1145/3487983.3488291

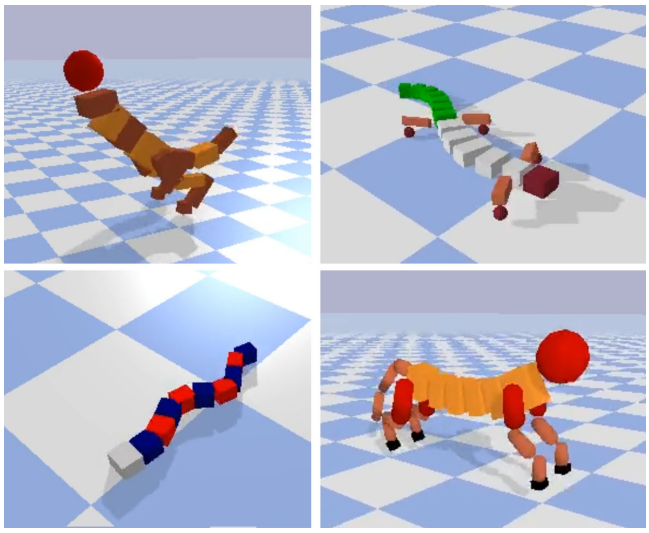

Figure 1: Locomotion learned from morphologically specific motor babble.

1 INTRODUCTION

Despite recent advances in trajectory optimization and reinforcement learning, it remains challenging to learn motor skills for physics-based articulated characters. While human motion data has been used to bootstrap control for humanoid characters, animating complex non-human characters like those seen in Figure 1 presents a challenging control problem which can be under specified and prohibitively high dimensional. While there is typically an ample space of control policies to accomplish motor tasks, not all results lead to *natural* and *coordinated* motion. This paper introduces an approach that attempts to mitigate this problem by extracting coordinated motor activations which are drawn from the character's own dynamics directly using a technique we call "motor babble" after its inspiration taken from robotics.

State-of-the art deep reinforcement learning (DRL) approaches excel at generating natural control policies for physically simulated humanoids, and, more recently, for simple quadrupeds by imitating motion capture clips of expert behaviors [Park et al. 2019; Peng

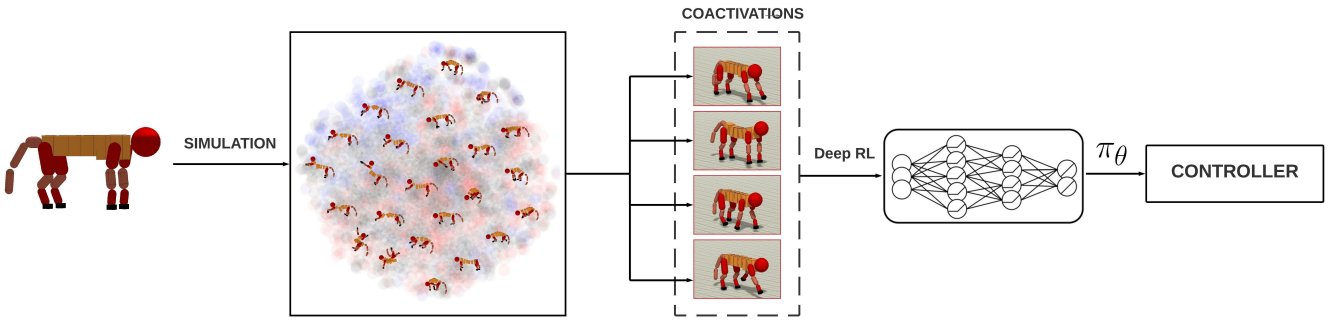

Figure 2: Overview of the proposed morphology-driven framework for learning coordinated control for articulated characters.

et al. 2018a, 2020, 2018b; Won et al. 2020]. Unfortunately, imitation learning cannot extend to characters of arbitrary morphology where expert data is not readily available. While some impressive results have been obtained for articulated animals, such solutions are typically character-specific, relying on careful tuning of the character model along with shaping the reward function or performance index in order to generate lifelike controls [Heess et al. 2017; Wampler and Popović 2009; Yu et al. 2018].

In this paper, we present a general framework for learning locomotion skills for characters with diverse morphologies and high articulation without requiring any access to pre-existing expert data. We draw inspiration from the line of work in extracting coordination for control through modal decomposition [Kry et al. 2009; Nunes et al. 2012]. The inspiration for this body of work is that complex beings such as animals exhibit natural coordination that stems from their physical structure and its co-articulation. Similarly, the core idea of our work is to exploit the morphological characteristics of the underlying physical system and use this to "co-activate" joint control.

To do so, we build a representative *coactivation* control space through a novel approach that deliberately actuates (exercises) low-level controls through low-level joint torques, while maintaining appropriate joint and torque limits. We call this technique motor babble, borrowing terminology from robotics in which exercising a physical robots leads to a predictable model of internal dynamics [Saegusa et al. 2009]. Upon proper activation, the motor babble corpus reveals joint synergies particular to the physical system. We use these as a joint coordination basis for co-activated control.

The main contributions of our work is a framework for generating control that can support a wide range of articulated characters. We particularly focus on characters for which expert reference motion is not available, and address the problem of natural control by proposing the following. **1)** *Motor babble*, a novel approach to build a representative corpus of motion that directly unfolds the synergies inherent in the dynamics of a character's morphological structure; **2)** A babble-inferred *joint-coactivation* space, i.e., a latent control action space extracted from the babble data that utilizes multiple joints simultaneously to effectively reduce the control space across the character's DOFs. **3)** *Coactivation-based training* of control policies using DRL. Our policy automatically selects coactivations that can be used to learn coordinated control for different articulated characters using the same reward function, reducing the need for character-specific tuning.

1.1 Overview

We refer to Figure 2 for an overview of our system. Motor babble (Section 3.1) is executed over a new character. Co-activations are constructed to become the viable basis for coordinated control (Section 3.2). And a motor policy is then trained using DRL (Section 4) that learns how to activate the synergies employing a minimalistic reward function. Namely, this general framework can handle a variety of articulated characters while relying on the same simple reward function, that focuses on the forward speed of the character in locomotion (Section 5). We demonstrate the applicability of our work on a number of physically simulated animals (Section 6). We further elaborate on the different choices for designing the motor babble and use a range of experiments to show the effect that different coordination bases have on the final control policy before concluding.

2 RELATED WORK

Study of animal locomotion has had a long history in robotics [Raibert 1986], with snake like locomotion controllers [Owen 1994] dating back to the early 1970's. Bi-modal gaits on amphibious robots like sea snake [Crespi et al. 2005; Crespi and Ijspeert 2006, 2008] and salamander [Ijspeert et al. 2007] are achieved by modeling biological neural circuitry called central pattern generators [Ijspeert 2008], found in spinal chord of vertebrates, as locomotion controller. In [Iida and Pfeifer 2004], simple oscillators are used as locomotion controllers on a quadruped robot, which result in a bounding gait as a result of the robot's interaction with its environment. The work of [Kohl and Stone 2004] is one of the early examples of learning fast walking gait on a quadruped as a control policy, using a policy gradient method. More recent state-of-the-art results in learning-based controllers on a quadruped robot, by imitating motor skills collected from a real animal, is shown in [Peng et al. 2020], where the authors first train a control policy in simulation, and then apply domain adaptation techniques to fine tune the learned policy for real world deployment. A related solution has been to rely on reference examples of expert motion [Lee et al. 2010; Sok et al. 2007; Wampler et al. 2014], with recent approaches following a reinforcement learning paradigm where the simulated character seeks to

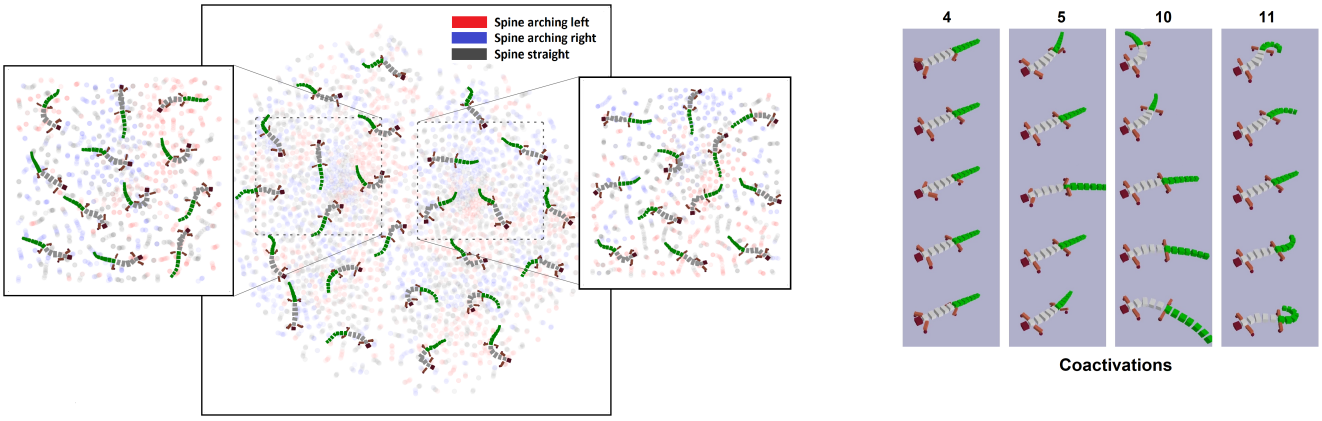

Figure 3: Motor Babble and Related Coactivations for the Salamander. (Left) Two-dimensional embedding using t-SNE of exemplar poses generated with our motor babble approach. (Right) Examples of babble-inferred coactivations. The coactivations are organized by column and are ranked based on their eigenvalues. The excitation of each coactivation results in simultaneous actuation of all of the character's joints. We refer to the video for the corresponding animations and additional results.

learn a policy that minimizes the tracking error between simulated poses and reference motion capture clips.

Research on animal character control in physics-based simulation environments has been a long standing focus of the animation community [Geijtenbeek and Pronost 2012]. [Miller 1988] is an early example where the authors demonstrate the use of periodic functions as locomotion controllers for snake and worm like character, in physics-based simulation. In [Van De Panne 1996; Van de Panne et al. 1994], the authors present locomotion controllers for articulated 2D cheetah and 3D legged characters, respectively. Here the authors use *pose control graph*, which is an open-loop control mechanism that associates, through optimization, a desired character pose for each state, which are used by the underlying PD controllers to drive the character's joints. A repertoire of quadruped gaits and motor skill is presented in [Coros et al. 2011], where a collection of controllers, including inverted pendulum model, PD controllers, and virtual forces are used as building blocks for producing high fidelity motions in a dog-like character. Other contributions from the animation community in this space includes hopping gait controllers for biped, quadruped and a kangaroo model [Raibert and Hodgins 1991], muscle-actuated locomotion controllers for snake and fish characters [Grzeszczuk and Terzopoulos 1995], swimming gaits in aquatic characters [Min et al. 2019; Tan et al. 2011], and aerial locomotion involving flapping-wings [Ju et al. 2013; Wu and Popović 2003].

More recent advances in DRL for learning in continuous action domain [Haarnoja et al. 2018; Peng et al. 2016; Schulman et al. 2017a] have produced start-of-the-art results in learning motor controllers for highly articulated characters in dynamic environments [Duan et al. 2016; Heess et al. 2017; Peng et al. 2018a, 2017; Won et al. 2017, 2018]. In [Yu et al. 2018], locomotion controller for simulated humanoid and other animal characters are learned using a curriculum learning approach, and by optimizing for gait symmetry and low-energy in the objective and reward functions respectively. An approach that shares some similarities with our proposed control framework is Multiplicative Compositional Policies (MCP) [Peng et al. 2019], where a control policy is modeled as an ensemble of behavior specific policy primitives, along with a learned gating function. The authors in [Luo et al. 2020] use the MCP model, along with a Generative Adversarial Networks (GAN) model to modulate high-level controls to learn very smooth gait transitions, and for recovering from unforeseen scenarios in a quadruped character.

Similar to this paper, other researchers have proposed techniques similar to our own for character animation where controls are computed from modal decomposition [Jain and Liu 2011; Kry et al. 2009; Nunes et al. 2012]. The benefit of our motor babble approach over modal analysis is twofold, we incorporate more variability in ground contact and pose (as we describe in the next section) which is prohibited in the previously described approaches for modes. Further, we deviate from these works by synthesizing complex full-body coordinated control policies automatically with DRL while previous work was applied to much simpler control problems [Kry et al. 2009], did not solve control explicitly [Jain and Liu 2011], or relied on hand-picked synergy selection [Kry et al. 2009; Nunes et al. 2012]. Other similar work [Ranganath et al. 2019] shows coactivation control spaces can be generated by decomposing human reference motion data for physics-based simulation control – which compels our interest in this area. In contrast, this work relied on task-specific reference motion data with a manual selection process to build its latent control space.

3 MOTOR BABBLE

Non-humanoid reference motion data is very rare. We propose to create artificial motion data, from which a latent control space and locomotion can be constructed for a specific character's morphology. To build a representative corpus of reference motion, our motor-babble method actuates low-level controls to exercise the dynamics of the character. We take care to craft a wide database by varying start and end poses in short (0.5sec) *micro-behaviors* and varying contact across the episodes. Our approach draws from the

modal decomposition work mentioned in [Kry et al. 2009] that also derives coordinated joint movements from the physical dynamics of a character, but it differs both in concept and in practice. While modal analysis stems from simple oscillations centered about a single known rest pose, our motor babble can include any pose within specified joint limits. Modes do not have a notion of joint limits, but rather staying close to the given rest pose. Further, contact changes (e.g., between ground and air) need to be built explicitly [Nunes et al. 2012] while babble includes multiple combinations of contacts seamlessly. This is important in characters such as quadrupeds where the combinations of different foot contacts make controllers derived from modal analysis difficult.

3.1 Episodic micro-behaviors

To exercise an unknown morphology with motor babble, we build a dynamic version of the character, with known masses, inertial properties, and limb lengths and connections. This simulation is reused in synthesis, and so the cost is negligible. Next, we specify nominal joint limits, as simple ranges over individual joint degrees of freedom (DOFs) as well as joint torque maximums. We also identify the contact bodies. With this information in place, the motor babble procedure produces micro-behaviors by creating random point-to-point activations under random contact conditions. Specifically, samples are performed episodically by initializing the character to a random (valid) state and the desired target pose is set as input to a stable proportional derivative (SPD) controller. The simulation is then integrated to produce the outcome. We refer to the accompanying video for examples.

Episodes are terminated and discarded if a termination condition is met: 1) there is an explosion in the joint velocity, 2) some predefined segment of the character touches the ground surface. Each valid micro-behavior is recorded at 30 Hz, and stored as motion data. As a post processing step, the motion data is mirrored in the XY-plane (i.e, around the forward and lateral axis), to maintain the left-right symmetry of the character in the motion data. For example, the motion database for the feline quadruped shown in Figure 2 contains over 20K frames after mirroring. Similar databases are built for the rest of the other characters, with the process taking up to an hour for a complex character such as the feline quadruped or kangaroo. Specifications about character morphology and parameters used for motor babble are available in the supplemental data.

3.2 Babble-informed Coactivation

The data generated through the process of motor babble constitutes a corpus of character poses which capture the synergies between the DOFs of a particular character. See, for example, Figure 3. Given such data, we use the notion of control *coactivations* [Ranganath et al. 2019], i.e., a set of motion primitives expressed as joint-coordination vectors where each vector defines how *all* the joints move together. Performing singular value decomposition on the collected motion data results in a set of coactivations which describe the fundamental motion primitives of the respective character. Formally, let matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ denote the motor babble generated data, where m corresponds to the number of poses and n to the number of DOFs of the character. Using singular value decomposition, the data matrix $\mathbf{X}$ can be decomposed as $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$, where $\Sigma\mathbf{V}^\top \in \mathbb{R}^{n \times n}$, a full rank basis matrix, represent n joint-coordination vectors that can be ordered based on their eigenvalues (see Figure 3 for some examples). Subsets of these motion primitives constitute representative control spaces for the character.

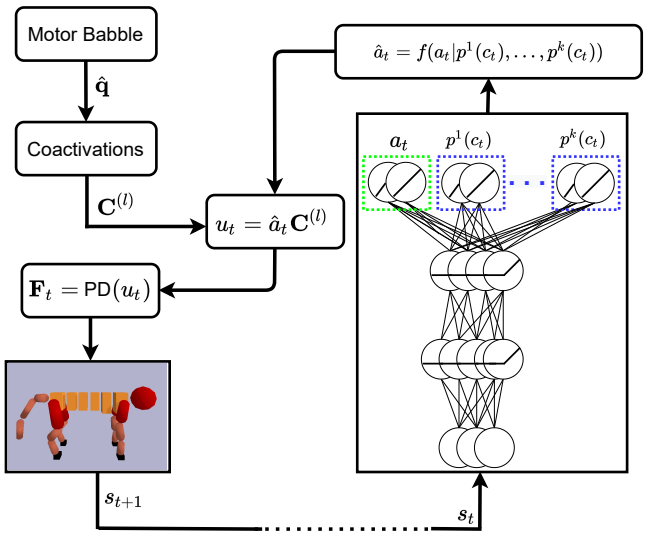

Figure 4: An overview of our general framework, with a multi-headed neural network as control policy, for sampling up to k out of l coactivations ($p^1(c_t), \ldots, p^k(c_t)$), and the corresponding *excitation* (a_t).

In total, motor babble can be conceptualized as a general process through which the dynamics of a character is exercised to discover a generic set of joint coordinations. However, like in previous modes-driven approaches, for a given specific motor task, typically only a small subset of joint coordinations are needed, constituting a task-specific *latent control space* for the character. In Section 4, we introduce a DRL approach for automatically selecting this latent space for locomotion, along with learning the coactivation excitations in the form of control policies.

4 LEARNING FROM COACTIVATIONS

In [Peng et al. 2018a], controllers for high fidelity motor skills are trained on a humanoid character by carefully crafting a reward function, including closely following joint trajectories from expert reference motion data as well as limiting motion through a number of reward terms. In contrast, in [Ranganath et al. 2019], a low dimensional controller is learned in the latent coactivation space derived from reference motion, significantly reducing the dimensionality of the control and lessening the need for a carefully shaped reward function. Building on this work, we learn to *select* a representative set of coactivations for a particular motor task from motor-babble coactivations, along with learning the excitations for these selected coactivations.

Specifically, we formulate the learning process to learn to automatically select a discrete set of k coactivations, from a given set

of top-l coactivations (where $l > k$), along with learning excitations for the respective coactivations. To do so, we formulate the control problem as a discounted Markov Decision Process (MDP) and solve it using reinforcement learning. The MDP is 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, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $P : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the state transition function, ρ_0 is the probability distribution over initial states, and $\gamma \in (0, 1]$ is the discount factor. As the character interacts with its environment, at every time step t, makes an observation $s_t \in \mathcal{S}$, takes an action $a_t \in \mathcal{A}$, receives a scalar reward r_t, and transitions to a new state s_{t+1} based on the underlying transition model P, while following a stochastic policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The goal of the character 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 the end of the episode, or until some termination condition is satisfied. For a policy $\pi_\theta(a|s)$, parameterized by θ, the objective of the learning process is to find the optimal set of parameters θ^*, which can be formulated as

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

The control architecture for the RL is presented in Figure 4. To learn using the motor babble-driven coactivations, the control policy π maps from states to actions ($\pi : \mathcal{S} \rightarrow \mathcal{A}$), where an action $\mathbf{a} \in \mathbb{R}^l$ denotes the *excitation* vector for the coactivations. Unlike learning in independent joint action space, here each excitation value $a_i \in \mathbb{R}, \forall i \in \{1, 2, \ldots, l\}$ contributes to the motion of all the joints (where l is the count of coactivations). In this way with even a small number of excitations (e.g. 2-3) we can control the full character. Following along with Figure 4, the policy network is tasked to learn k independent discrete probability distributions as a function of the state: $p^i(c|s), \forall i \in \{1, 2, \ldots, k\}$, where $c \in \{1, 2, \ldots, l\}$ represent the coactivation indices. Here, each $p^i(\cdot)$ is a distribution over all top-l coactivations, from which a coactivation is sampled at each time step t. The sampled coactivation from each distribution is encoded as a one-hot vector $\mathbb{1}^i$, where $\mathbb{1}^i_c = 1$ corresponds to the c^{th} coactivation being sampled from the i^{th} distribution $p^i(\cdot)$.

Since all k distributions are modelled as independent distributions, it is possible for the same coactivation to be sampled by multiple distributions. So, post sampling, only $\hat{k}$ unique coactivations, where $1 \leq \hat{k} \leq k$, are considered and represented as an indicator variable $\mathbb{1}^E$, where $\mathbb{1}^E_c = 1$, indicates that coactivation c was sampled by at least one of the k discrete distributions. Then, a sparse vector of sampled excitations of the corresponding coactivations are represented as $\hat{\mathbf{a}} \in \mathbb{R}^l$, where

$$\hat{a}_c = \begin{cases} a_c & \text{if } \mathbb{1}^E_c = 1 \\ 0 & \text{otherwise.} \end{cases}$$

By learning in the coactivation space, the target joint angles for the character are obtained by a simple transformation: $\mathbf{u} = \hat{\mathbf{a}}\mathbf{C}^{(l)}$, where $\mathbf{u} \in \mathbb{R}^n$ are the target joint angles for the character, $\hat{\mathbf{a}} \in \mathbb{R}^l$ is a sparse vector of coactivation excitations, and $\mathbf{C}^{(l)} \in \mathbb{R}^{l \times n}$ is the coactivation matrix.

The policy network is designed as a multi-headed neural network with $k+1$ heads at the output layer. Each head has l output neurons, with one head (O_e) representing the means $\mu(s)$ of a multivariate Gaussian distribution over excitations, while the remaining k heads (O_{c^i}) representing discrete distributions over coactivations. The policy network consists of two hidden layers of 1024 and 512 neurons, respectively. Rectified Linear Units (ReLU) activation functions are used for the hidden layers, and a linear activation function for the output layer, while a soft-max operation is used to obtain probabilities at each O_{c^i}. The excitations head (O_e) outputs the mean, $\mu(s)$, for an independent multivariate Gaussian distribution, from which excitations as actions of the control policy can be sampled as $a \sim \mathcal{N}(\mu(s), \Sigma)$. The covariance of the Gaussian is represented by a fixed diagonal matrix $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_l)$, where σ_i is the variance of the i^{th} excitation. The sampled excitations are transformed into target joint angles $\mathbf{u}$, which are fed as input to a SPD controller. The generated joint torques, $\mathbf{F}$, are then applied to the simulated character.

5 TRAINING LOCOMOTION

To employ the architecture to our problem of interest, learning locomotion motor skills, we propose a simple reward function and introduce a handful of behavior specific termination conditions that make training possible.

5.1 Reward Function

An important impact of the motor babble approach is that we do not need a complicated reward function because the computed control space constrains the controller inherently. Our simple reward function for locomotion consists of a single term that pertains to the velocity of the root of the character which encourages the character to move at the target speed along the forward direction while penalizing high lateral speeds. In particular

$$r_t = exp(-\|\Delta \mathbf{v}\|^2), \tag{2}$$

where $\Delta \mathbf{v}_x = \max(0, \min(\hat{\mathbf{v}}_x, \dot{\mathbf{v}}_x t) - \mathbf{v}_x)$ is the difference between the character's forward velocity, $\mathbf{v}_x$, and the target forward velocity, $\hat{\mathbf{v}}_x$. The user-defined initial acceleration term $\dot{\mathbf{v}}_x$, allows the character to reach the target forward velocity in $\hat{\mathbf{v}}_x / \dot{\mathbf{v}}_x$ seconds from the start of the episode, and t is the elapsed time of the episode in seconds. Setting the initial acceleration values aids in the character learning to transition from zero velocity, at the beginning of the episode, to the target velocity smoothly, and in the absence of which the learning algorithm over optimizes and results in the character lunging forward at the beginning, which in turn leads to gait instability. The velocity difference along the vertical axis is not considered in the reward. The velocity difference along the lateral axis set as a hyperparameter as:

$$\Delta \mathbf{v}_z = \begin{cases} \mathbf{v_z} & \text{if } \mathbb{1}^z = 1 \\ 0 & \text{otherwise,} \end{cases}$$

5.2 Termination Conditions

The training is performed episodically, with a fixed episode length of $T = 20s$. Along with the reward function, several termination conditions are used to shape the behavior of the learning agent by guiding it in the task space. If the agent encounter's a termination

condition, then the episode is terminated early, and resulting in a zero reward value for that time step.

COM. All the characters have a COM (Center Of Mass) position limit in the vertical axis, going beyond which the episode is terminated. This condition ensure that there isn't a torque or velocity explosion.

Fall detection. This termination condition checks at every time step, if any of a subset of the character's links makes contact with the ground surface. This condition speeds up learning, especially during the early stages of the training, by eliminating those sections of the control space which result in the character falling to the ground. Also, this termination condition can help shape the learned behavior; without fall detection, certain characters could end up learning to crawl instead of walking or running (in case of a quadruped for example), as the reward function is minimalistic, relying mainly on the velocity of the character's root.

Self-collisions. If segments of a character collide with each other, unless the colliding segments share a joint between them, then the episode is terminated as well. This is a standard termination condition used across all our characters and training.

Flight phase. A flight phase is defined as a time step where none of the character's segments are in contact with the ground surface. We use this as an optional termination condition, terminating an episode when a flight phase is detected, while ground contact should be maintained, such as for walking and slithering behaviors.

5.3 Training

We use the Proximal Policy Optimization (PPO) [Schulman et al. 2017b] for training control policies. PPO is an off-policy method that relies on samples collected from an older policy $\pi_{\theta_{old}}$ to estimate the expected return of the current policy π_θ and uses a clipped surrogate objective to constrain how far the new policy can deviate from the old one. In our implementation, the value function is trained with multi-step TD(λ) return and the advantage is estimated using λ-return as in GAE [Schulman et al. 2015].

The clipped surrogate objective function used in PPO is:

$$L(\theta) = \mathbb{E}_{\mathbf{s}_t, \mathbf{a}_t \sim \pi_{\theta_{old}}}[\min(g_t(\theta)A_t, clip(g_t(\theta), 1-\epsilon, 1+\epsilon)A_t)], \tag{3}$$

where $g_t = \frac{\pi_\theta(\mathbf{a}|\mathbf{s})}{\pi_{\theta_{old}}(\mathbf{a}|\mathbf{s})}$ is the importance sampling term, ϵ is a tunable hyper parameter that determines how far the new policy can deviate from the old, and A_t is the advantage at time t.

Regarding importance sampling, in our formulation, the joint probability of action $\mathbf{a}$ over the state for a given policy is computed as

$$\pi_\theta(\mathbf{a}|\mathbf{s}) = \prod_c^l p(a_c|\mathbf{s},\theta)^{\mathbb{1}_c^E} \cdot \prod_i^k \prod_j^l p^i(c_j|\mathbf{s},\theta)^{\mathbb{1}_j^i}, \tag{4}$$

where, l is the number of coactivations provided to the learning framework at the beginning of the training, k is the number of discrete distributions modelled to learn to select at most k unique coactivations, $p(\cdot)$ is the continuous action probability distribution, modelled as a multivariate Gaussian distribution, $p^i(\cdot)$ is the discrete coactivation sampling distribution, and $\mathbb{1}^E$ and $\mathbb{1}^i$ are the indicator variable and one-hot vectors, respectively, over the sampled coactivations. With the clipped objective function, at each time step t, action probabilities are considered exclusively for actions whose corresponding coactivation is sampled. Accordingly, the gradients are back propagated only for these selected actions.

From the action vector $\mathbf{a}$ of the policy network, representing excitations, target joint angles for the character are obtained by the transformation $\mathbf{u} = \hat{\mathbf{a}}\mathbf{C}^{(l)}$, where $\mathbf{u} \in \mathbb{R}^n$ are the target joint angles for the character, $\hat{\mathbf{a}} \in \mathbb{R}^l$ is a sparse vector of excitations corresponding to the sampled coactivations, and $\mathbf{C}^{(l)} \in \mathbb{R}^{l \times n}$ is the coactivation matrix. The target joint angles are fed as input to the SPD controllers which generate joint torques applied to the respective joints. State $\mathbf{s}$ consists of the position, orientation, linear, and angular velocity of each joint of the character, with orientations of 3D joints being represented as quaternions, while 1D joints are represented as scalar rotation angles in radians. All values are expressed in a local coordinate system centered at the root of the character and oriented along the root's facing direction.

6 RESULTS

To showcase the power of the approach, we develop examples for locomotion for various characters with unique morphologies and gait/movement affordances. Figure 1 and the supplemental data highlight these characters. The inputs for each character includes the anticipated joint ranges and relative torque limits as well as the basic skeletal data and masses. This information was then used to build motor babble and a representative set of coactivations from which to learn controllers. Training (as described in the last two sections) requires a small set of specific choices related to each character. Namely, we must identify the contact mode and bodies, including whether we want the locomotion to include a flight phase. Also, we must select the desired speed, e.g. the feline quadruped was to run at 6m/s while the salamander character was to target 3m/s. And finally, we also indicate a basic dimensionality for the controller; the snake, being rather simple, we allow to use only three coactivations, while the feline and salamander quadrupeds we set to allow to use up to five.

With this information, the gaits of each character "emerge". The distinction in the locomotion styles is driven by the babble and control coactivations, with no special efforts made beyond the inputs to the babble procedure. To exercise the system, we conducted a small set of experiments, using the automatic selection and comparing this technique to others approaches.

6.1 Automatic Selection of Coactivations

Each animal was tested to produce a basic gait unique to its own morphology. For example, the salamander character was trained to select at most 5 unique coactivations, from the top-12 coactivations, for achieving a target velocity of $3ms^{-1}$. The *flight phase* termination condition was applied, which resulted in a fast walking gait, akin to a salamander (see video). Notably, the RL achieved this gait by learning to select only two unique coactivations. Similarly, the feline character was tasked to learn using up to 5 unique coactivations, out of the top-32 coactivations. The velocity was set to $6ms^{-1}$. In this case, the agent learns a common quadruped bounding gait with only two unique coactivations. Both characters can be seen in

the filmstrips in Figure 5 along with a hopping gait for the kangaroo character. Two versions of the snake character were trained with coactivations generated from motor babble performed by actuating the character's joints around the vertical axis, and lateral axis. The latter appears to be more caterpillar-like. In the vertical (snake) case, an anisotropic friction was added of 0.125 in the forward axis, and a lateral friction of 2.0 to enable a slithering gait. Both cases are trained with top-8 coactivations, and are tasked to learn to select up to 3 unique coactivations. The resulting gaits are presented in the video. Training hyperparameters used in each case are available in the supplemental data.

We see promise in the approach based on the diversity of character gaits and morphologies that are derived from the same framework, with simple and intuitive changes made from case to case. Figure 7 shows that the training time and success naturally follows with the complexity of the character, with the feline character being the most complicated of our examples. Minimal tuning was necessary, as an example, the kangaroo was hopping from babble to full gait from only a few edits from an expert user. To further showcase the benefits of this approach, we contrast the automatic DRL described with control derived without babble and with an approach in which an expert selects the specific coactivations to employ. To make these comparisons, we modify the DRL architecture by removing the selection of top k coactivations and instead create a fixed control matrix $\mathbf{C^k} \in \mathbb{R}^{k \times n}$, built as described below. Finally, we highlight our exploration of babble in contrast to modal analysis as a means of control.

6.2 Independent Joint Action Space

To compare the proposed approach to more common control schemes, we train all our characters with the same reward formulation, termination conditions and training hyperparameters, but in the Independent Joint Action Space (IJAS). Learning a control policy in the IJAS corresponds to learning to control each DOF of each joint of the character independently, which differs starkly from our approach where we learn to control the character in the coactivation space of joint *coordinations*. Noting the IJAS for our characters is an order of magnitude more complex in general (e.g. the feline has 59 DOFs), while the total dimensionality of the control for our character is less than five in all of our automatic selection cases.

The characters are trained in IJAS with the actions as the desired values for each DOF. That is, the output of the policy network are the desired joint positions, rather than coactivation excitations. As the video demonstrates, the resulting gaits are successful, as defined by the reward function but are far from natural. For example, the resulting gait of the salamander exhibits an awkward *crooked-leg* walking gait, which appears unnatural. For the two versions trained at different speeds of the feline character, the approach was able to find a policy at the slow target speed, the resulting gait is an asymmetric two-legged hop. The second increased target speed creates a locomotion which is almost unrecognizable, lumbering forward in a highly unrealistic manner. These and other animations of the gaits appear in the video (labelled IJAS).

Learning to control in the coactivation space constrains the agent to the space of joint synergies that are found from exercising the character's morphology. But in the case of controlling in the IJAS, the agent is unconstrained and fails to learn natural looking gaits due to the sheer amount of redundant solutions that satisfy the minimal reward function.

In context, the previous work in character control gets around the issues of IJAS by either adding expert data (motion capture), or by shaping the reward function or power limits carefully. We argue that while specific action spaces can be hand shaped in characters with simple morphology like the snake, allowing natural-looking training to results from the IJAS, such approach does not scale for more complex characters and tasks as in the feline. Our approach sidesteps these issues, in lieu of a more streamlined, easier to tune DRL approach which we can further exploit through the addition of manual selection, described next.

6.3 Manual Selection of Coactivations

The next investigation we perform is that of manual selection, harkening back to the modes research previously mentioned [Kry et al. 2009; Nunes et al. 2012]. We note that except for very simple cases, all results obtained with such previous work required manual selection of the desired modes. Our approach of automatic selection of coactivations sidesteps this need. However, we explore here the utility of manual selection as a high-level means for directing locomotion.

With minimal changes to the existing framework, we can construct the control space by manually selecting the coactivations, thereby targeting a specific gait/style. To accomplish this, the control space is constructed from select coactivations picked by visualizing them, and choosing the desired behavior's "style". Similar ideas appear in the previous work and it is not a surprise that by shaping the control space in this manner, we can also direct the style of the produced locomotion. In the video, we present gaits learned for the salamander, feline, and kangaroo characters through this pre-selection of coactivations (labeled Manual).

We highlight our exploration and modifications for a manually crafted walk gait for the feline character. Motor babble data of the character results in a rich and diverse set of coactivations, as representation of the character's overall dynamics. With a natural-looking quadrupedal walk as the target gait, we can empirically select a small subset of coactivations that include the signature swing of the legs in alternation trivially from quick inspection. Next, we train the character with forward velocity of $1ms^{-1}$. Exclusively for walking, we found it best to add an additional energy penalty to the reward to balance torque expenditure here formulated as: $J^{-1} \sum_j^J \max(1, |\omega_t^j \tau_t^j|)^{-1}$, where, ω_t^j and τ_t^j are the angular velocity and torque of the j^{th} joint at time t, respectively. This term minimizes the energy cost as measured by the total instantaneous power averaged over all J joints. A filmstrip of the resulting gait is in Figure 5 and the video showcases it as well. In total, we produced three unique gaits form the same set of motor babble, shown in Figure 6.

An important distinction from the previous papers on modes is the more powerful DRL framework being employed here. While the closest previous work [Nunes et al. 2012] show the production of plausible gaits using CMA-ES [Hansen 2006], they do not produce balancing physically simulated locomotion. In contrast, (all of) our characters learn policies that maintain balance in 3D for

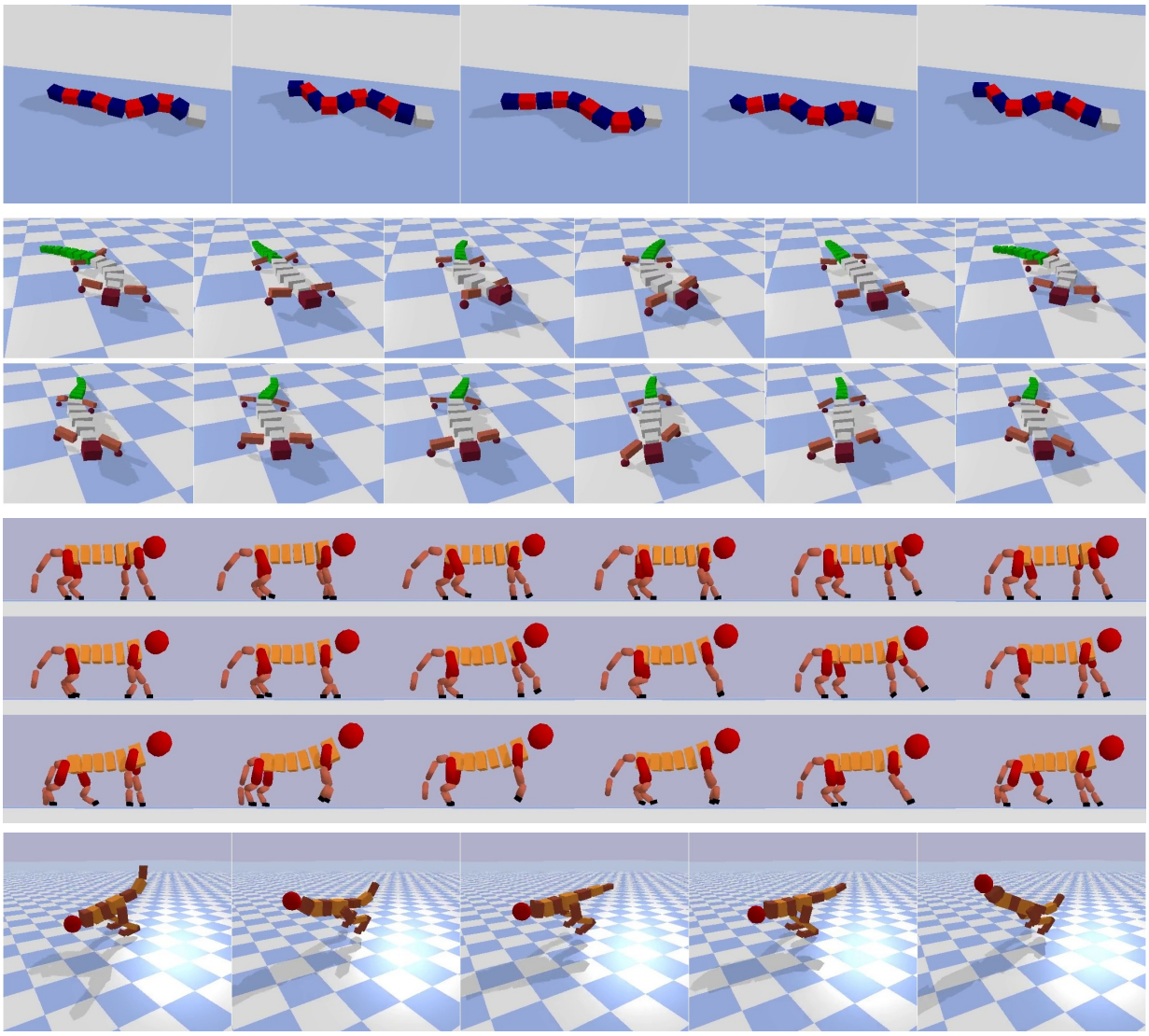

Figure 5: Film strips showcasing several of the gaits learned by our characters. A video containing all these gaits are available in the supplemental data.

long episodes. This alone is not a significant advance, but should be included in consideration of the technique overall as other DRL approaches are notoriously hard to control style without any reference motion [Peng et al. 2018a].

6.4 Modes vs. Babble

In comparison to modal decomposition in [Kry et al. 2009; Nunes et al. 2012], motor babble is a general process for extracting synergies in articulated characters. For example, to produce joint coactivations in line with the aforementioned work, we perform motor babble with two changes to the process which are: (i) using a fixed root of the character such that the character does not interact with the ground surface, and (ii) starting each episode with the character in the rest pose. Performing modes like motor babble on the salamander and feline quadrupeds result in coactivations which are visually very similar to the original coactivations, and training with these coactivations as inputs result in similar gaits.

While these experiments to assess the differences caused by the motor babble over modal decomposition did not reveal large differences, this suggests that the motor babble approach may be *more general* than the procedure used in the proposed approach and further investigation of babble will likely further reveal the advantage of this generalizability. But it can be noted that the motor babble approach by its nature provides an easy mechanism for including and excluding more aspects of the dynamics, such as joint limits, torque limits, and varying contact configurations. The modal analysis, in contrast is limited, as the outcome is specific to oscillations

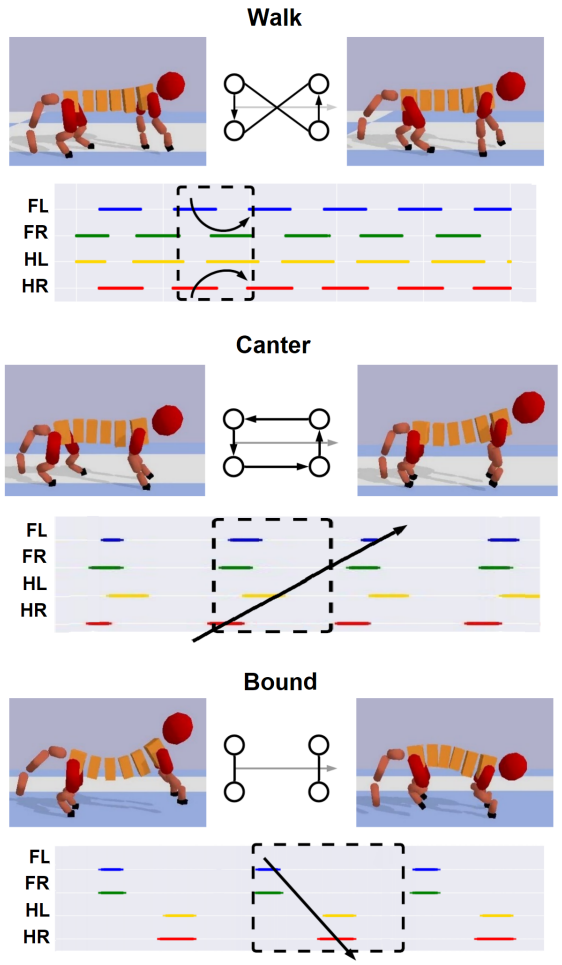

Figure 6: Feet contact plots and visualization of cheetah gaits. Our approach is able to automatically discover a range of gaits by varying the target speed of the character. The grey arrows represent the head of the character and the open circles the feet. The black lines connecting the feet represent simultaneous footfalls and the black arrows the order of succession of the footfalls. FL: front left foot; FR: front right foot; HL: hind left foot; HR: hind right foot.

about a single rest pose, under a single contact configuration, and with no notion of joint limits.

7 DISCUSSION AND CONCLUSION

We present a general framework for controlling highly articulated characters in the absence of expert reference data. Below we provide some empirical analysis of the results shown in the previous section, as well as discuss limitations of our work along with avenues for future work. We propose different ways to generate control through our framework that includes automatically learning to select a small set of coactivations as part of the DRL training process as well as manual selection. In all cases, different types of gait styles emerge

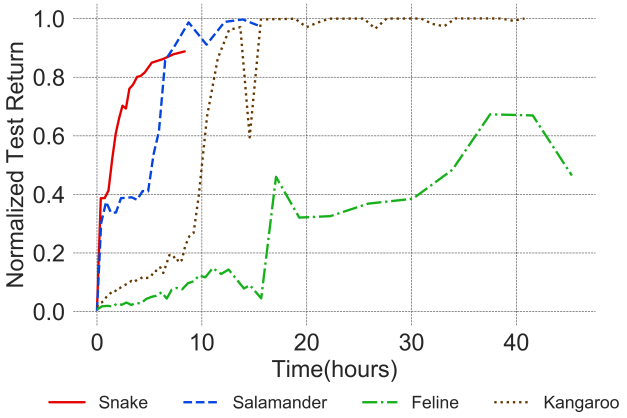

Figure 7: Plot of test returns normalized over the length of the episode for the four characters trained using automatic selection procedure.

with a simple reward function. By adjusting the target speed and using the different coactivation mechanisms, for example, the feline quadruped exhibits a range of gaits such as walking, trotting, and cantering, as shown see Figure 6.

It is interesting to note that while exploring the two axes (target speed vs coactivation selection mechanism), we were able to gain more insights about how to generate coordinated control that stems from the co-articulation of structures. As the target speed is increasing, asking for the system to automatically find and excite a small number of coactivations is typically successful. Increasing the target speed, makes the control problem in the coactivation space more constrained, and subsequently the task of finding a few coactivations is more defined as compared to the many ways that a character can walk slowly in a coordinated fashion (e.g. see salamander walks in video). In contrast, when targeting high speeds, it is difficult for a user to manually decide which coactivations need to be combined for achieving such a task while it is convenient (to control style) by manually picking coactivations that lead to desired locomotion behaviors when the character targets a low moving speed. While one can perceive this as similar to reward shaping, in practice it is hard to create reward functions that are applicable to a wide range of characters. Manually choosing a small number of ranked joint synergies is a less tedious task, and can generalize across characters.

In our current work, coactivations are extracted by applying eigenvalue decomposition to the babble data. While non-linear manifolds obtained with techniques such as autoencoders [Holden et al. 2017] have the potential to find a richer latent space, the eigenvalue decomposition provides a natural ranking to the extracted coactivations. Under the light of the above discussion, such ranking is particularly important as it can be used to reduce the space exposed to the user (for manual selection) or to the policy network (for automatic selection). In the future, we would like to compare methods for learning a latent action space from the babble data and the effect that it has on the generated controls.

Another interesting direction is exploring the applicability of the motor babble to generating a wide range of controllers rather than just focusing on individual motor tasks, such as locomotion. Currently, we use a state-independent coactivation matrix during training that is extracted offline from the babble data. We speculate that such a matrix can be refined during training of different individual policies for separate primitives/behaviors, while still being state-independent. The connections between various expert policies can be added in a subsequent phase following the recent works of [Luo et al. 2020; Peng et al. 2019; Won et al. 2020].

REFERENCES

Stelian Coros, Andrej Karpathy, Ben Jones, Lionel Reveret, and Michiel Van De Panne. 2011. Locomotion skills for simulated quadrupeds. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–12.

Alessandro Crespi, André Badertscher, André Guignard, and Auke Jan Ijspeert. 2005. AmphiBot I: an amphibious snake-like robot. *Robotics and Autonomous Systems* 50, 4 (2005), 163–175.

Alessandro Crespi and Auke Jan Ijspeert. 2006. AmphiBot II: An amphibious snake robot that crawls and swims using a central pattern generator. In *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)*. 19–27.

Alessandro Crespi and Auke Jan Ijspeert. 2008. Online optimization of swimming and crawling in an amphibious snake robot. *IEEE Transactions on Robotics* 24, 1 (2008), 75–87.

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*. PMLR, 1329–1338.

Thomas Geijtenbeek and Nicolas Pronost. 2012. Interactive character animation using simulated physics: A state-of-the-art review. In *Computer graphics forum*, Vol. 31. Wiley Online Library, 2492–2515.

Radek Grzeszczuk and Demetri Terzopoulos. 1995. Automated learning of muscle-actuated locomotion through control abstraction. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 63–70.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*. PMLR, 1861–1870.

Nikolaus Hansen. 2006. The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation* (2006), 75–102.

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, Taku Komura, and Jun Saito. 2017. Phase-functioned Neural Networks for Character Control. *ACM Transactions on Graphics* 36, 4, Article 42 (2017), 13 pages.

Fumiya Iida and Rolf Pfeifer. 2004. Cheap rapid locomotion of a quadruped robot: Self-stabilization of bounding gait. In *Intelligent autonomous systems*, Vol. 8. IOS Press Amsterdam, The Netherlands, 642–649.

Auke Jan Ijspeert. 2008. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks* 21, 4 (2008), 642–653.

Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. 2007. From swimming to walking with a salamander robot driven by a spinal cord model. *science* 315, 5817 (2007), 1416–1420.

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

Eunjung Ju, Jungdam Won, Jehee Lee, Byungkuk Choi, Junyong Noh, and Min Gyu Choi. 2013. Data-driven control of flapping flight. *ACM Transactions on Graphics (TOG)* 32, 5 (2013), 1–12.

Nate Kohl and Peter Stone. 2004. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, Vol. 3. IEEE, 2619–2624.

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

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

Ying-Sheng Luo, Jonathan Hans Soeseno, Trista Pei-Chun Chen, and Wei-Chao Chen. 2020. CARL: Controllable Agent with Reinforcement Learning for Quadruped Locomotion. *ACM Transactions on Graphics* 39, 4 (2020), 10.

Gavin SP Miller. 1988. The motion dynamics of snakes and worms. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*. 169–173.

Sehee Min, Jungdam Won, Seunghwan Lee, Jungnam Park, and Jehee Lee. 2019. Softcon: Simulation and control of soft-bodied animals with biomimetic actuators. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–12.

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 *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. 87–94.

Tony Owen. 1994. Biologically Inspired Robots: Snake-Like Locomotors and Manipulators by Shigeo Hirose Oxford University Press, Oxford, 1993, 220pages, incl. index (£ 40). *Robotica* 12, 3 (1994), 282–282.

Soohwan Park, Hoseok Ryu, Seyoung Lee, Sunmin Lee, and Jehee Lee. 2019. Learning Predict-and-Simulate Policies From Unorganized Human Motion Data. *ACM Transactions on Graphics* 38, 6, Article 205 (2019).

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* (2018).

Xue Bin Peng, Glen Berseth, and Michiel van de Panne. 2016. Terrain-Adaptive Locomotion Skills Using Deep Reinforcement Learning. *ACM Transactions on Graphics (Proc. SIGGRAPH 2016)* 35, 4 (2016).

Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. 2017. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics* 36, 4 (2017), 41.

Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. 2019. MCP: Learning Composable Hierarchical Control with Multiplicative Compositional Policies. In *Annual Conference on Neural Information Processing Systems*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 3681–3692.

Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Edward Lee, Jie Tan, and Sergey Levine. 2020. Learning Agile Robotic Locomotion Skills by Imitating Animals. In *Robotics: Science and Systems*. https://doi.org/10.15607/RSS.2020.XVI.064

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* (2018).

Marc H Raibert. 1986. Legged robots. *Commun. ACM* 29, 6 (1986), 499–514.

Marc H Raibert and Jessica K Hodgins. 1991. Animation of dynamic legged locomotion. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. 349–358.

Avinash Ranganath, Pei Xu, Ioannis Karamouzas, and Victor Zordan. 2019. Low Dimensional Motor Skill Learning Using Coactivation. In *Motion, Interaction and Games*. 1–10.

Ryo Saegusa, Giorgio Metta, and Giulio Sandini. 2009. Active learning for multiple sensorimotor coordination based on state confidence. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2598–2603.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. 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. 2017a. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

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

Kwang Won Sok, Manmyung Kim, and Jehee Lee. 2007. Simulating Biped Behaviors from Human Motion Data. *ACM Transactions on Graphics* 26, 3 (2007), 107âĂŞes. https://doi.org/10.1145/1276377.1276511

Jie Tan, Yuting Gu, Greg Turk, and C Karen Liu. 2011. Articulated swimming creatures. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–12.

Michiel Van De Panne. 1996. Parameterized gait synthesis. *IEEE Computer Graphics and Applications* 16, 2 (1996), 40–49.

Michiel Van de Panne, Ryan Kim, and Eugene Fiume. 1994. Virtual wind-up toys for animation. In *Graphics Interface*. Citeseer, 208–208.

Kevin Wampler and Zoran Popović. 2009. Optimal gait and form for animal locomotion. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 1–8.

Kevin Wampler, Zoran Popović, and Jovan Popović. 2014. Generalizing locomotion style to new animals with inverse optimal regression. *ACM Transactions on Graphics* 33, 4 (2014), 1–11.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics* 39, 4 (2020), 33–1.

Jungdam Won, Jongho Park, Kwanyu Kim, and Jehee Lee. 2017. How to train your dragon: example-guided control of flapping flight. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

Jungdam Won, Jungnam Park, and Jehee Lee. 2018. Aerobatics control of flying creatures via self-regulated learning. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.

Jia-chi Wu and Zoran Popović. 2003. Realistic modeling of bird flight animations. *ACM Transactions on Graphics (TOG)* 22, 3 (2003), 888–895.

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