SUPPLEMENTARY MATERIAL

Bounded Action Space

Properties such as torque and neural activation limits result in bounds on the range of values that can be assumed by actions for a particular parameterization. Improper enforcement of these bounds can lead to unstable learning as the gradient information outside the bounds may not be reliable (Hausknecht and Stone 2015). To ensure that all actions respect their bounds, we adopt a method similar to the inverting gradients approach proposed by Hausknecht and Stone (2015). Let $\nabla a = (a - \mu(s))\mathbb{A}(s,a)$ be the empirical action gradient from the policy gradient estimate of a Gaussian policy. Given the lower and upper bounds $[l^i, u^i]$ of the ith action parameter, the bounded gradient of the ith action parameter $\nabla \tilde{a}^i$ is determined according to

$$\nabla \tilde{a}^i = \begin{cases} l^i - \mu^i(s), & \mu^i(s) < l^i \text{ and } \nabla a^i < 0 \\ u^i - \mu^i(s), & \mu^i(s) > u^i \text{ and } \nabla a^i > 0 \\ \nabla a^i, & \text{otherwise} \end{cases}$$

Unlike the inverting gradients approach, which scales all gradients depending on proximity to the bounds, this method preserves the empirical gradients when bounds are respected, and alters the gradients only when bounds are violated.

Reward

The terms of the reward function are defined as follows:

$$r_{pose} = \exp\left(-||q^* - q||^2_W\right)$$

$$r_{vel} = \exp\left(-||\dot{q}^* - \dot{q}||^2_W\right)$$

$$r_{end} = \exp\left(-40 \sum_e ||x_e^* - x_e||^2\right)$$

$$r_{root} = \exp\left(-10(h^*_{root} - h_{root})^2\right)$$

$$r_{com} = \exp\left(-10||\dot{x}^*_{com} - \dot{x}_{com}||^2\right)$$

q and q^* denotes the character pose and reference pose represented in reduced-coordinates, while $\dot{q}$ and $\dot{q}^*$ are the respective joints velocities. W is a manually-specified per joint diagonal weighting matrix. h_{root} is the height of the root from the ground, and $\dot{x}_{com}$ is the center of mass velocity.

Sensitivity Analysis

We further analyze the sensitivity of the results to different initializations and design decisions. Figure 13 compares the learning curves from multiple policies trained using different random initializations of the networks. Four policies are trained for each actuation model. The results for a particular actuation model are similar across different runs, and the trends between the various actuation models also appear to be consistent. To evaluate the sensitivity to the amount of exploration noise applied during training, we trained policies where the standard deviation of the action distribution is twice and half of the default values. Figure 14 illustrates the learning curves for each policy. Overall, the performance of the policies do not appear to change significantly for the particular range of values. Finally, Figure 15 compares the results using different network architectures. The network variations include doubling the number of units in both hidden layers, halving the number of hidden units, and inserting an additional layer with 512 units between the two existing hidden layers. The choice of network structure does not appear to have a noticeable impact on the results, and the differences between the actuation models appear to be consistent across the different networks.

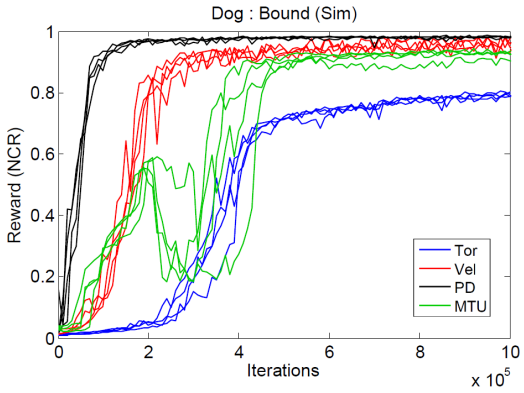

Figure 13: Learning curves from different random network initializations. Four policies are trained for each actuation model.

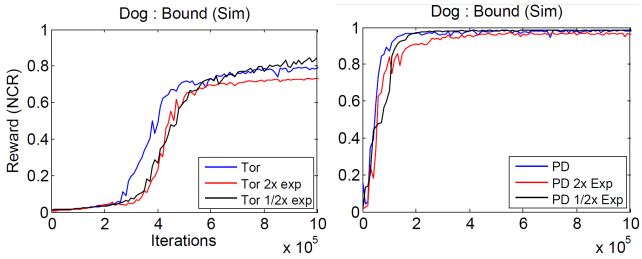

Figure 14: Learning curves comparing the effects of scaling the standard deviation of the action distribution by 1x, 2x, and 1/2x.

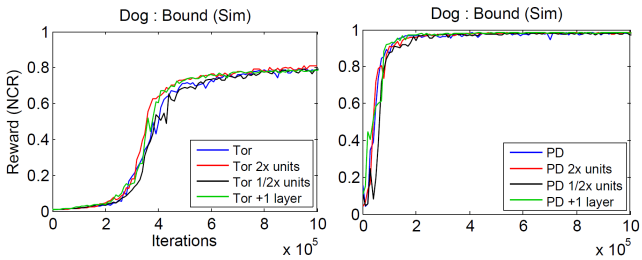

Figure 15: Learning curves for different network architectures. The network structures include, doubling the number of units in each hidden layer, halving the number of units, and inserting an additional hidden layer with 512 units between the two existing hidden layers.

While the MTU actuation parameters were optimized via automated actuator optimization, the actuation parameters for the other models, e.g. PD gains, were manually specified. To analyze the policies' sensitivity to the manually specified parameters, we trained PD policies with gains scaled by 0.25, 0.5, 1, and 2. Figure 16 shows the learning curves using the different sets of actuation parameters. The behaviour of the policies appears robust to changes within a factor of 2. Reducing the gains to 0.25 of their default values exhibits some negative impact to performance.

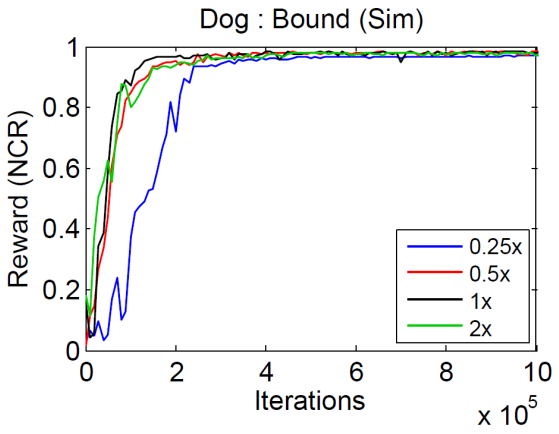

Figure 16: Learning curves for PD policies with different scalings of the PD gains.

To further evaluate the actuation models' performance with respect to different reference motions, we recorded a reference motion of the MTU policy for Dog : Bound (Sim), and retrained new policies to imitate the motion. Learnings curves are available in Figure 17. The performance of the various policies appear to respect the trends observed with other reference motions. Though the reference motion was recorded from an MTU policy, the PD policy still learns more quickly than the MTU policy.

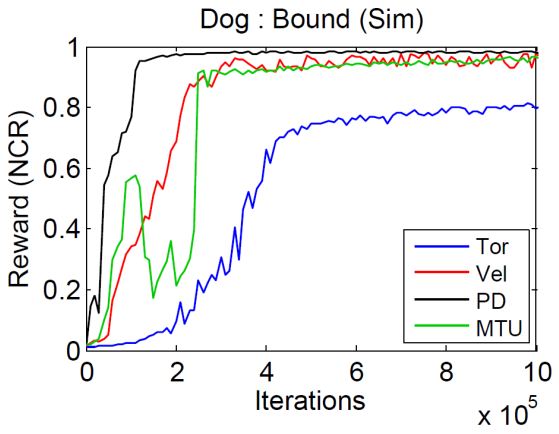

Figure 17: Learning curves for policies trained to imitate a reference motion recorded from an MTU policy.

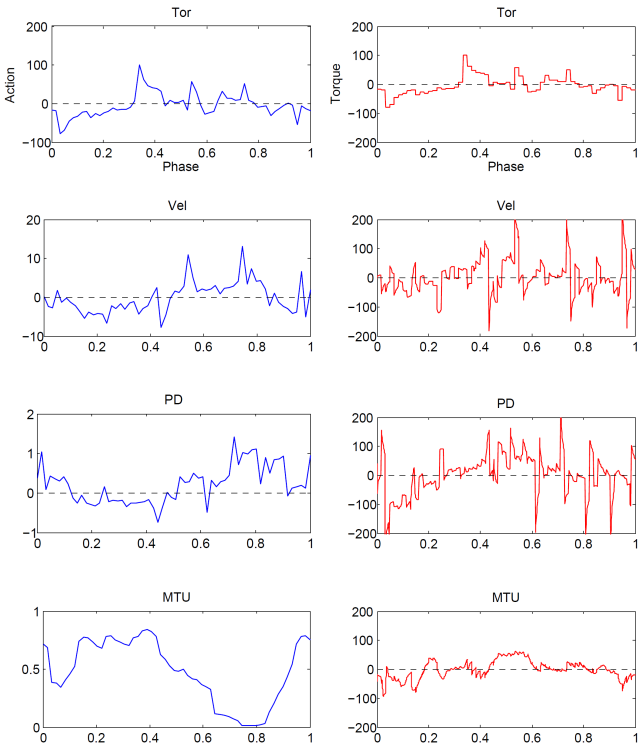

Figure 18: Policy actions over time and the resulting torques for the four action types. Data is from one biped walk cycle (1s). Left: Actions (60 Hz), for the right hip for PD, Vel, and Tor, and the right gluteal muscle for MTU. Right: Torques applied to the right hip joint, sampled at 600 Hz.

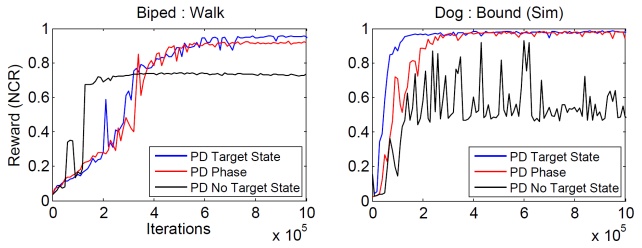

Figure 19: Learning curves for different state representations including state + target state, state + phase, and only state.

Parameter	Value	Description
γ	0.9	cumulative reward discount factor
α_π	0.001	actor learning rate
α_V	0.01	critic learning rate
momentum	0.9	stochastic gradient descent momentum
ϕ weight decay	0	L2 regularizer for critic parameters
θ weight decay	0.0005	L2 regularizer for actor parameters
minibatch size	32	tuples per stochastic gradient descent step
replay memory size	500000	number of the most recent tuples stored for future updates

Table 2: Training hyperparameters.

Character + Actuation Model	State Parameters	Action Parameters	Actuation Parameters
Biped + Tor	58	6	0
Biped + Vel	58	6	6
Biped + PD	58	6	12
Biped + MTU	74	16	114
Raptor + Tor	154	18	0
Raptor + Vel	154	18	18
Raptor + PD	154	18	36
Raptor + MTU	194	40	258
Dog + Tor	170	20	0
Dog + Vel	170	20	20
Dog + PD	170	20	40
Dog + MTU	214	44	282

Table 3: The number of state, action, and actuation model parameters for different characters and actuation models.

Character + Actuation	Motion	Performance (NCR)	Learning Speed (AUC)
Biped + Tor	Walk	0.7662 ± 0.3117	0.4788
Biped + Vel	Walk	0.9520 ± 0.0034	0.6308
Biped + PD	Walk	0.9524 ± 0.0034	0.6997
Biped + MTU	Walk	**0.9584 ± 0.0065**	**0.7165**
Biped + Tor	March	0.9353 ± 0.0072	0.7478
Biped + Vel	March	**0.9784 ± 0.0018**	0.9035
Biped + PD	March	0.9767 ± 0.0068	**0.9136**
Biped + MTU	March	0.9484 ± 0.0021	0.5587
Biped + Tor	Run	0.9032 ± 0.0102	0.6938
Biped + Vel	Run	**0.9070 ± 0.0106**	0.7301
Biped + PD	Run	0.9057 ± 0.0056	**0.7880**
Biped + MTU	Run	0.8988 ± 0.0094	0.5360
Raptor + Tor	Run (Sim)	0.7265 ± 0.0037	0.5061
Raptor + Vel	Run (Sim)	0.9612 ± 0.0055	0.8118
Raptor + PD	Run (Sim)	**0.9863 ± 0.0017**	**0.9282**
Raptor + MTU	Run (Sim)	0.9708 ± 0.0023	0.6330
Raptor + Tor	Run	0.6141 ± 0.0091	0.3814
Raptor + Vel	Run	0.8732 ± 0.0037	0.7008
Raptor + PD	Run	**0.9548 ± 0.0010**	**0.8372**
Raptor + MTU	Run	0.9533 ± 0.0015	0.7258
Dog + Tor	Bound (Sim)	0.7888 ± 0.0046	0.4895
Dog + Vel	Bound (Sim)	0.9788 ± 0.0044	0.7862
Dog + PD	Bound (Sim)	**0.9797 ± 0.0012**	**0.9280**
Dog + MTU	Bound (Sim)	0.9033 ± 0.0029	0.6825
Dog + Tor	Rear-Up	0.8151 ± 0.0113	0.5550
Dog + Vel	Rear-Up	0.7364 ± 0.2707	0.7454
Dog + PD	Rear-Up	**0.9565 ± 0.0058**	**0.8701**
Dog + MTU	Rear-Up	0.8744 ± 0.2566	0.7932

Table 4: Performance of policies trained for the various characters and actuation models. Performance is measured using the normalized cumulative reward (NCR) and learning speed is represented by the normalized area under each learning curve (AUC). The best performing parameterizations for each character and motion are in bold.