Science Robotics
AAAS

robotics.sciencemag.org/cgi/content/full/4/26/eaau5872/DC1

Supplementary Materials for

Learning agile and dynamic motor skills for legged robots

Jemin Hwangbo*, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, Marco Hutter,

*Corresponding author. Email: jhwangbo@ethz.ch

Published 16 January 2019, *Sci. Robot.* **4**, eaau5872 (2019)
DOI: 10.1126/scirobotics.aau5872

The PDF file includes:

Other Supplementary Material for this manuscript includes the following:

(available at robotics.sciencemag.org/cgi/content/full/4/26/eaau5872/DC1)

Supplementary materials

Section S1. Nomenclature

k_c	curriculum factor.
$c_.$	coefficient for a cost term.
v_{AB}^C	linear velocity of B frame respect to A frame expressed in C frame
ω	angular velocity
$\hat{\cdot}$	desired quantity
τ	joint torque
ϕ	angular quantity
v_f	linear velocity of a foot
v_{ft}	tangential velocity of a foot (x, y components)
p_f	linear position of a foot
$v_{c,n}$	linear velocity of the n-th contact point
$i_{c,n}$	contact impulse of the n-th contact
g_i	gap function of the i-th possible contact pair
I_c	index set of all contacts
$I_{c,f}$	index set of foot contacts
$I_{c,i}$	index set of internal contacts
$\lvert\cdot\rvert$	cardinality of a set or l_1 norm
$\lvert\lvert\cdot\rvert\rvert$	l_2 norm
$\mathbf{0}^n$	n-dimensional vector of zeroes
$\mathbf{q}$	generalized coordinate
$\mathbf{u}$	generalized velocity

Section S2. Random command sampling method used for evaluating the learned command-conditioned controller.

The motivation of having a special sampling method is the limited size of the experimental area. A sufficiently long unconstrained sequence of randomly sampled velocity commands may drive the robot outside the limits of the physical space available. We therefore use the following sampling scheme. We first sample a command from the distribution described in table S1. Then we simulate a position trajectory by assuming that the body follows the velocity command perfectly. If the position of the body goes outside the perimeter of the available space, we reject the sampled command, reset the position to the previous position, and re-

sample a velocity command. This loop continues until the desired number of commands are sampled.

Section S3. Cost terms for training command-conditioned locomotion and high-speed locomotion tasks

We used a logistic kernel to define a bounded cost function $K : \mathbb{R} \rightarrow [-0.25, 0)$ as

$$K(x) = -\frac{1}{e^x + 2 + e^{-x}}. \tag{1}$$

This kernel converts a tracking error to a bounded reward. We found it more useful than Euclidean norm, which is a more common choice. An Euclidean norm generates a high cost in the beginning of training where the tracking error is high such that termination (i.e. falling) becomes more rewarding strategy. On the other hand, the logistic kernel ensures that the cost is lower-bounded by zero and termination becomes less favorable. Many other bell-shaped kernels (Gaussian, triweight, biweight, etc) have the same functionality and can be used instead of a logistic kernel.

The symbols used in this section are defined in section S1. Note that many cost terms are also multiplied by the time step Δt since we are interested in the integrated value over time. Explanation on the curriculum factor k_c can be found in subsection "Training in simulation".

angular velocity of the base cost ($c_w = -6\Delta t$)

$$c_w K(|\omega^I_{IB} - \hat{\omega}^I_{IB}|) \tag{2}$$

linear velocity of the base cost ($c_{v1} = -10\Delta t, c_{v2} = -4\Delta t$)

$$c_{v1} K(|c_{v2} \cdot (v^I_{IB} - \hat{v}^I_{IB})|) \tag{3}$$

torque cost ($c_\tau = 0.005\Delta t$)

$$k_c c_\tau ||\tau||^2 \tag{4}$$

joint speed cost ($c_{js} = 0.03\Delta t$)

$$k_c c_{js} ||\dot{\phi}^i||^2 \quad \forall i \in \{1, 2..., 12\} \tag{5}$$

foot clearance cost ($c_f = 0.1\Delta t, \hat{p}_{f,i,z} = 0.07$ m)

$$k_c c_f (\hat{p}_{f,i,z} - p_{f,i,z})^2 ||v_{ft,i}||, \ \forall i, g_i > 0, i \in \{0, 1, 2, 3\}, \tag{6}$$

foot slip cost ($c_{fv} = 2.0\Delta t$)

$$k_c c_{fv} ||v_{ft,i}||, \ \forall i, g_i = 0, i \in \{0, 1, 2, 3\} \tag{7}$$

orientation cost ($c_o = 0.4\Delta t$)

$$k_c c_o ||[0, 0, -1]^T - \phi_g|| \tag{8}$$

smoothness cost ($c_s = 0.5\Delta t$)

$$k_c c_s ||\tau_{t-1} - \tau_t||^2 \tag{9}$$

Section S4. Cost terms for training recovery from a fall

We use $\texttt{angleDiff} : \mathbb{R} \times \mathbb{R} \to [0, \pi]$ that computes the minimum angle difference between two angular positions to define a cost function on the joint positions. The symbols used in this section are defined in section S1.

torque cost ($c_\tau = 0.0005\Delta t$)

$$k_c c_\tau ||\tau||^2 \tag{10}$$

joint speed cost ($c_{js} = 0.2\Delta t$, $c_{jsmax} = 8$ rad/s)

$$\text{If } |\dot{\phi}^i| > |c_{jsmax}|, \quad k_c c_{js} ||\dot{\phi}^i||^2 \quad \forall i \in \{1, 2...12\} \tag{11}$$

joint acceleration cost ($c_{ja} = 0.0000005\Delta t$)

$$k_c c_{ja} ||\ddot{\phi}^i||^2 \quad \forall i \in \{1, 2...12\} \tag{12}$$

HAA cost ($c_{HAA} = 6.0\Delta t$)

$$\text{If } |\phi_{roll}| < 0.25\pi, \quad k_c c_{HAA} K(\texttt{angleDiff}(\phi^{HAA}, 0)) \tag{13}$$

HFE cost ($c_{HFE} = 7.0\Delta t$, $\hat{\phi}^{HFE} = \pm 0.5\pi$ rad (+ for right legs))

$$\text{If } |\phi_{roll}| < 0.25\pi, \quad k_c c_{HFE} K(\texttt{angleDiff}(\phi^{HFE}, \hat{\phi}^{HFE})) \tag{14}$$

KFE cost ($c_{KFE} = 7.0\Delta t$, $\hat{\phi}^{KFE} = \mp 2.45$ rad)

$$\text{If } |\phi_{roll}| < 0.25\pi, \quad k_c c_{KFE} K(\texttt{angleDiff}(\phi^{KFE}, \hat{\phi}^{KFE})) \tag{15}$$

contact slip cost ($c_{cv} = 6.0\Delta t$)

$$k_c c_{cv} \frac{\sum_{n \in I_c} ||v^I_{c,n}||^2}{|I_c|} \tag{16}$$

body contact impulse cost ($c_{cimp} = 6.0\Delta t$)

$$k_c c_{cimp} \frac{\sum_{n \in I_c \setminus I_{c,f}} ||\dot{i}^I_{c,n}||}{|I_c| - |I_{c,f}|} \tag{17}$$

internal contact cost ($c_{cint} = 6.0\Delta t$)

$$k_c c_{cint} |I_{c,i}| \tag{18}$$

orientation cost ($c_o = 6.0\Delta t$)

$$c_o ||[0, 0, -1]^T - \phi_g||^2 \tag{19}$$

smoothness cost ($c_s = 0.0025\Delta t$)

$$k_c c_s ||\tau_{t-1} - \tau_t||^2 \tag{20}$$

Table S1. Command distribution for training command-conditioned locomotion. During training the command was varied randomly as shown in this table. The range was selected to match the capabilities of the existing controllers.

	min	max
forward velocity	-1.0 m/s	1.0 m/s
lateral velocity	-0.4 m/s	0.4 m/s
turning rate	-1.2 rad/s	1.2 rad/s

Table S2. Command distribution for training high-speed locomotion. During training the command was varied randomly as shown in this table. Only the forward velocity command has a large variation since this task focuses only on high speed.

	min	max
forward velocity	-1.6 m/s	1.6 m/s
lateral velocity	-0.2 m/s	0.2 m/s
turning rate	-0.3 rad/s	0.3 rad/s

Table S3. Initial state distribution for training both the command-conditioned and high-speed locomotion. The initial state is randomized to make the trained policy more robust.

	mean	standard deviation
base position	$[0, 0, 0.55]^T$	1.5 cm
base orientation	$[1, 0, 0, 0]^T$	0.06 rad (about a random axis)
joint position	$[0, 0.4, -0.8, 0, 0.4, -0.8, 0, -0.4, 0.8, 0, -0.4, 0.8]^T$	0.25 rad
base linear velocity	$\mathbf{0}^3$	0.012 m/s
base angular velocity	$\mathbf{0}^3$	0.4 rad/s
joint velocity	$\mathbf{0}^{12}$	2 rad/s

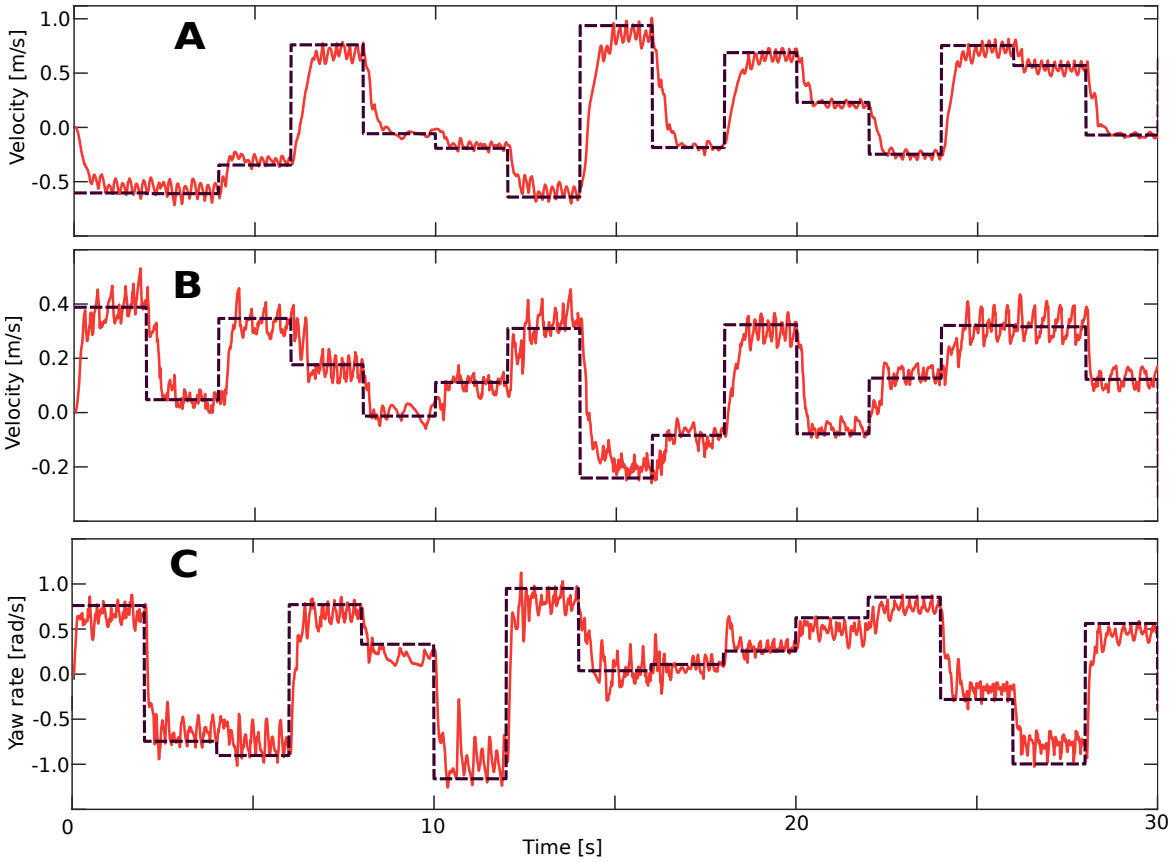

Figure S1. Base velocity tracking performance of the learned controller while following random commands. (**A**) Forward velocity, (**B**) Lateral velocity, (**C**) Yaw rate. For all graphs, the dotted lines represent the commanded velocity and the solid lines represent the measured velocity. All commands are followed with a reasonable accuracy even when the commands are given in a random fashion.

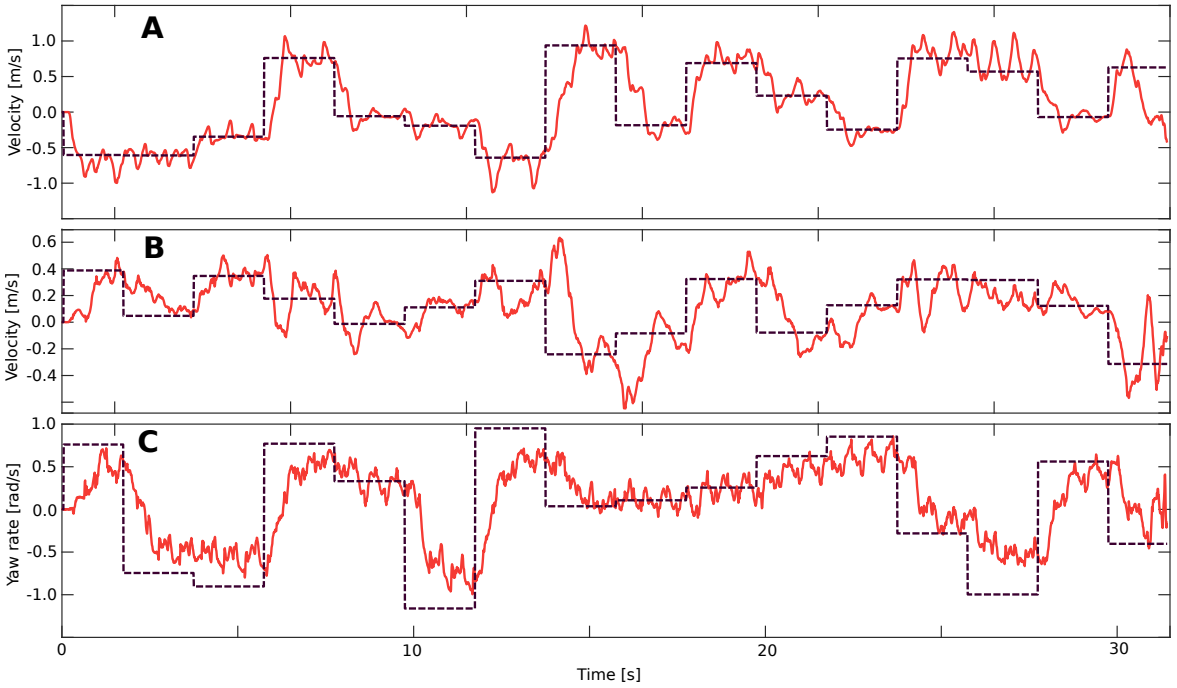

Figure S2. Base velocity tracking performance of the best existing method while following random commands. (**A**) Forward velocity, (**B**) Lateral velocity, (**C**) Yaw rate. For all graphs, the dotted lines represent the commanded velocity and the solid lines represent the measured velocity. The tracking performance is significantly worse than the learned policy.

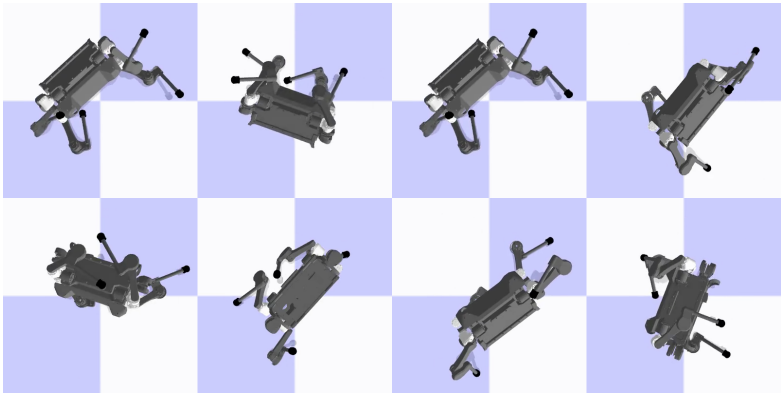

Figure S3. Sampled initial states for training a recovery controller.