Supplementary Materials: Reference-free Learning Bipedal Motor Skills via Assistive Force Curricula

Fan Shi, Yuta Kojio, Tasuku Makabe, Tomoki Anzai, Kunio Kojima, Kei Okada, and Masayuki Inaba

The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan,
shifan@jsk.imi.i.u-tokyo.ac.jp,
http://www.jsk.t.u-tokyo.ac.jp

1 Reward details

The desired hybrid whole-body locomotion behavior contains the following aspects:

- tracking the target command as r^{cmd}
- smooth joints command as r^{a}
- low energy consumption as r^{e}
- foot being parallel to ze ground during contact as r^{ζ_f}
- low frequency in the contact switch as r^{c}
- low contact velocity in the tangent plane as r^{c_v}
- minor oscillation of the robot body as r^{ω}

The target command reward is defined as the following:

$$r^{cmd} = k_{cmd} * e^{-k_{e_{cmd}}||\delta \boldsymbol{s}_{cmd}||^2} \tag{1}$$

where $\boldsymbol{s}_{cmd}$ denotes the difference with the desired command.

The smooth motion is expected to minimize the differentiation of joints angle commands as the following:

$$r^{a} = k_a * ||\boldsymbol{\theta}_d - \boldsymbol{\theta}_{d_{prev}}|| \tag{2}$$

where $\boldsymbol{\theta}_{d_{prev}}$ denotes the joints angle command at the previous time point.

The energy consumption is represented by the joints power as follows:

$$r^{e} = k_e * ||\boldsymbol{\theta}_{\boldsymbol{\tau}}^T \cdot \dot{\boldsymbol{\theta}}|| \tag{3}$$

where $\boldsymbol{\theta}_{\boldsymbol{\tau}}$ denotes the joints torque vector.

Foot orientation during contact is represented as follows:

$$r^{\zeta_f} = k_{\zeta_f} * e^{-k_{e_{\zeta_f}}||\zeta_f||^2} \tag{4}$$

where ζ_f denotes the foot orientation angle with respect to the ground.

To avoid high-frequency contact switch (which is impossible for real robots), the contact switch is punished as the following:

$$r^c = k_{f_c} * ||\boldsymbol{f}_c - \boldsymbol{f}_{c_{prev}}|| \tag{5}$$

where $\boldsymbol{f}_c$ and $\boldsymbol{f}_{c_{prev}}$ denote the feet binary contact state at the current and previous time point.

Foot tangent velocity during contact could avoid the unexpected slip. The reward is as follows:

$$r^{c_v} = k_{c_v} * (||\boldsymbol{v}_{f_l}^{tan}|| + ||\boldsymbol{v}_{f_r}^{tan}||) \tag{6}$$

where $\boldsymbol{v}_{f_l}^{tan}$ and $\boldsymbol{v}_{f_r}^{tan}$ denote the foot velocity in the tangent plane.

To reduce the oscillation in robot body, the angular velocity is to be minimized as follows:

$$r^{\omega} = k_{\omega} * ||\boldsymbol{\omega}|| \tag{7}$$

where ω denotes the torso's angular velocity.

In locomotion task, to avoid the feet dragging behavior, the foot clearance is encouraged as the following:

$$r^{f_d} = \textstyle\sum_{i \in \{l,r\}} (k_{f_d} * e^{-k_{e_{f_d}} ||f_{d_i} - h_d||^2}) \tag{8}$$

where f_d and h_d denote the foot distance in z axis and the desired foot height.

To gradually reduce the assistive force, the assistance reward is represented as follows:

$$r^T = k_T * (||\boldsymbol{F}_{T[0:3]}|| + ||\boldsymbol{F}_{T[3:6]}||) \tag{9}$$

where $\boldsymbol{F}_{T[0:3]}, \boldsymbol{F}_{T[3:6]}$ denote the two sets of assistance force individually.

Here, $k_{cmd}, k_a, k_e, k_{\zeta_f}, k_{f_c}, k_{c_v}, k_{\omega}, k_{f_d}, k_T$ denote the weight of each reward term.

2 Sim-to-real transfer

Simulation calibration

We operate the robot with a previously developed model-based controller [3] and record the joints position data during walking. The next step is to conduct the same controller in the simulation. By sampling the parameter space, we record the total difference in joints position. The parameter set with minimal joints position difference is selected as the suitable parameters for the simulation. The simulation calibration procedure could be summarized as the following:

$$\theta_{sim}^* = \arg\min_{\theta_{sim}} \left[\sum_{t=0}^{t_f} ||\boldsymbol{q}_{sim}(t) - \boldsymbol{q}_r(t)|| \right] \tag{10}$$

where θ_{sim} denotes the parameters to estimate for the simulation, $\boldsymbol{q}_{sim}(t)$ and $\boldsymbol{q}_r(t)$ denote the joints position in the simulation and real robot.

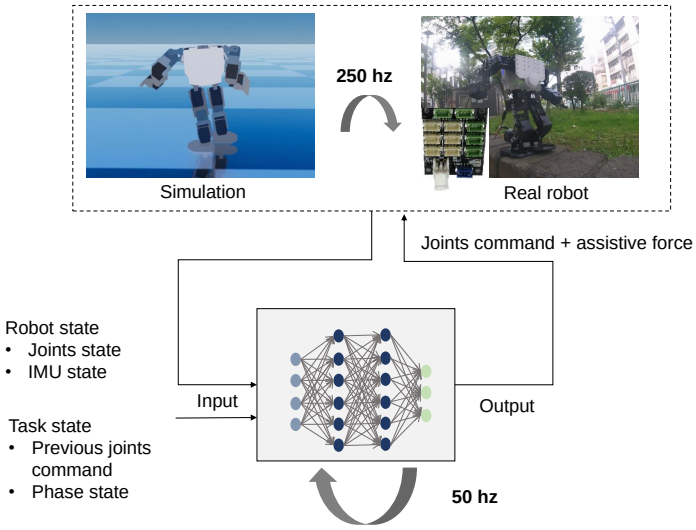

Fig. 1. System overview of the simulation and real robot.

Noises addition To simulate the noisy input of the real robot, we add the additional noises to the joints position and IMU data.

- Joint position in observation with additive noise $\sim \mathcal{N}(3.0)\,\mathrm{deg}$
- Euler angle in observation with additive noise $\sim \mathcal{N}(3.0)\,\mathrm{deg}$
- Angular velocity in observation with additive noise $\sim \mathcal{N}(5.0)\,\mathrm{deg}$
- Initial configuration with additive noise $\sim \mathcal{N}(4.0)\,\mathrm{deg}$

Disturbance injection Applying external disturbance to the robot is another method to robustify the learned controller. During the learning process, the humanoid robot is applied with the random-direction disturbance force on its center of gravity, left hand, and right hand. The magnitude of external force is 15% of the weight with 1.0-second gap time in average.

3 Experiments

3.1 Locomotion

To evaluate the performance of our learned whole-body walking controller, we compare with a model-based controller in [3]. This controller is developed based on capture point with simplified linear inverted pendulum model (LIPM), and successfully demonstrated on multiple humanoid robot platforms with reactive push recovery ability against large external force.

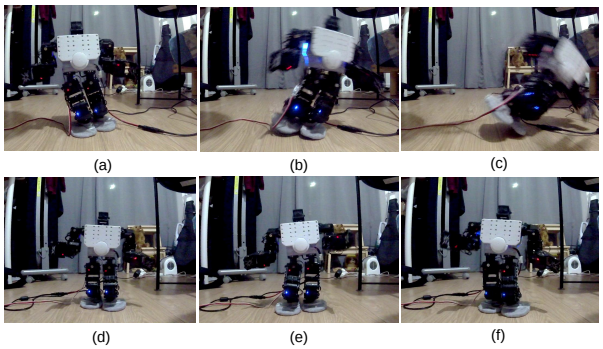

Fig. 2. Comparison of model-based controller and learned controller walking on rubber socks. (a)-(c) are the failure in model-based controller because of the influence of unexpected friction changes, (d)-(f) are the stable performance of our learned controller.

Performance under environment changes

In the first KXR experiment, the ground is randomly put with small metal pipes, which are very easy to slide with indoor wooden ground.The model-based controller falls down about every 3 steps, while the learned controller is very robust to the sliding obstacles. One of the reasons is the learned whole-body controller has less angular momentum on the feet because of the compensation from arms' motion, which shows the advantage of whole-body controller as well.

In the second KXR experiment, we put the baby socks on the feet of the robot, which has the high-friction-coefficients design with rubber dots to increase friction force. The model-based controller failed to walk because of the friction coefficients being different with its pre-tuned value. In contrast, our learned controller shows robustness against the unknown changes as Fig. 2.

Performance comparing with previous model-based controller

In the simulation, we evaluate the performance of both controller on the JAXON humanoid robot. The first experiment is to evaluate the consumed torque during locomotion. The robot is commanded with the same walking speed ($0.2m/s$) in baseline controller and our learned controller. The plot of joints torque is as Fig. 3, in which the averaged normalization values of joints torque are as the following:

Controller	Averaged total torque	Averaged legs torque
Model-based controller [3]	$211.55Nm$	$207.88Nm$
Our learned controller	$213.00Nm$	$192.85Nm$

Because of our learned controller generates more dynamic arm motion compared to the almost static arm motion in the previous model-based controller, the total torque is 1.7% larger. If only comparing the joints torque on the dual

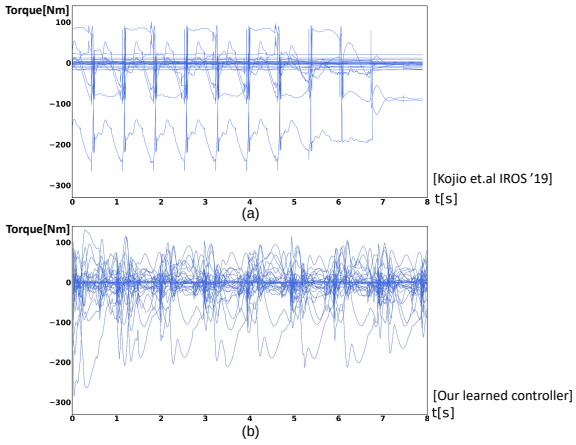

Fig. 3. Comparison of joints torque of model-based controller and learned controller during locomotion, in which the average walking speed is $[0.2, 0, 0]m/s$. (a) denotes the model-based controller, (b) denotes our learned controller.

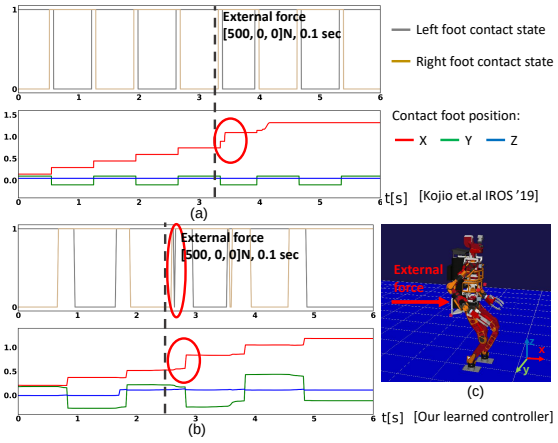

Fig. 4. Comparison of push recovery from backward force in model-based controller and learned controller during locomotion. (a) denotes the model-based controller, (b) denotes our learned controller, (c) is the experiment visualization.

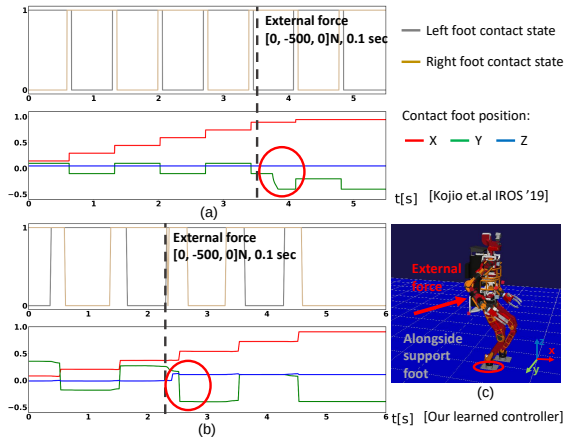

Fig. 5. Comparison of push recovery from rightward force in model-based controller and learned controller during locomotion. (a) denotes the model-based controller, (b) denotes our learned controller, (c) is the experiment visualization.

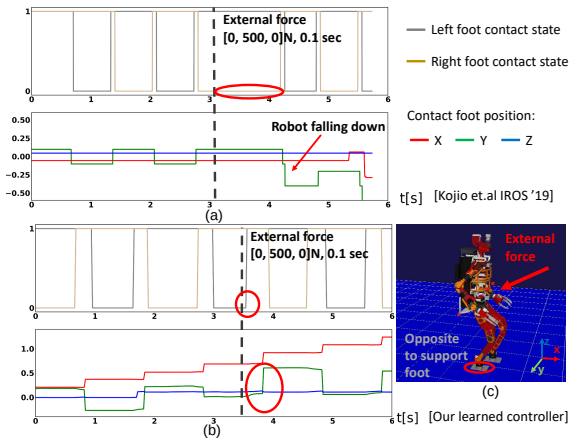

Fig. 6. Comparison of push recovery from leftward force in model-based controller and learned controller during locomotion. (a) denotes the model-based controller, (b) denotes our learned controller, (c) is the experiment visualization.

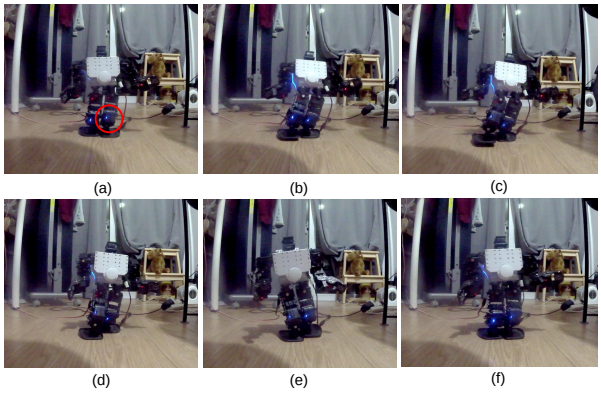

Fig. 7. Learned controller based on the asymmetrical humanoid robot with a fixed left knee motor, in which the robot successfully learns whole-body locomotion behavior with our proposed learning framework.

legs, the learned controller requires 90.5% torque of the model-based controller. In addition, the joints torque plot is more smooth in our learned controller as Fig. 3.

The second experiments are to compare the robustness of both controller under unexpected large external force. In comparison, the robot torso is applied with $500N$ external force with $0.1s$ time as [3], in which the external force is applied in the lateral and backward direction. When applying in the backward direction, both controller shows reactive behavior with the adjustment of foot landing position as Fig. 4, in which the swing foot takes a large step to recovery from the sudden backward push.

When applying external force in the lateral direction, there are two cases, which are the force is alongside or opposite to the support foot. For the first case, the swing foot in both controller shows the reactive behavior by taking a large step to recovery from the lateral force as Fig. 5. For the second case, the model-based controller failed by attempting to extend support foot contact time to recovery the balance, while our learned controller shows the dynamic behavior with immediate contact switch to change the support foot for the recovery motion.

Based on these comparison, our learned controller shows more robust behavior against external force, even for the corner case of the previous model-based controller.

Learning under different hardware settings

To fully verify our learning framework, we learn the locomotion behavior on multiple humanoid robot platforms with different configuration settings. We also

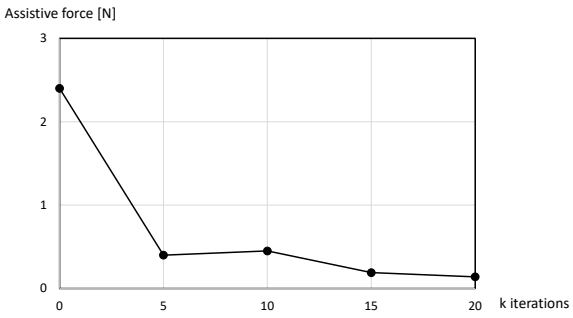

Fig. 8. Trend of assistive force during learning KXR locomotion in Stage 2.

testify our learning framework on the asymmetrical KXR robot with fixed motors as Fig. 7. The proposed framework successfully learns the locomotion behavior for all the tested robots.

3.2 Dynamic skills

Bipedal jumping

The joints torque plot is demonstrated in the submitted paper. Here we add the trend plot of assistive force during learning as Fig. 9. When each assistance is lower than 10% of the robot weight, we will switch to Stage 3 for learning without the assistive force.

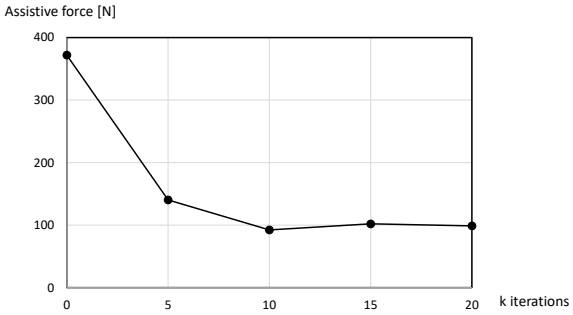

Fig. 9. Trend of assistive force during learning Atlas bipedal jumping in Stage 2.

Rotation jumping

In addition to the plot showed in the submitted paper, we also analyze the

joints torque distribution. The Atlas simulation we utilized in the experiment is the open-source DRC version with 174-kg mass [1], which is almost 2 times of the new Atlas robot. Therefore, to achieve dynamic skills requires more joints torque compared to the new Atlas robot showed in the video [2].

For continuous rotational jumping in Fig. 10, the joints torque during the behavior is as Fig. 11. For 360° rotation jumping in Fig. 12, the joints torque is as Fig. 13.

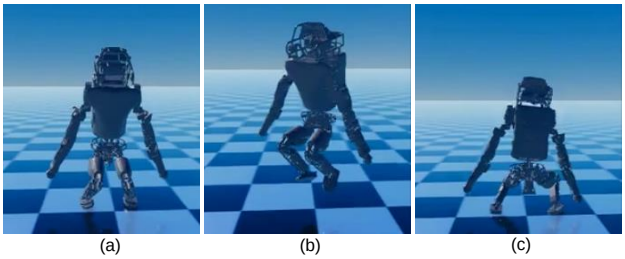

Fig. 10. Learned controller on a 180° rotation jumping.

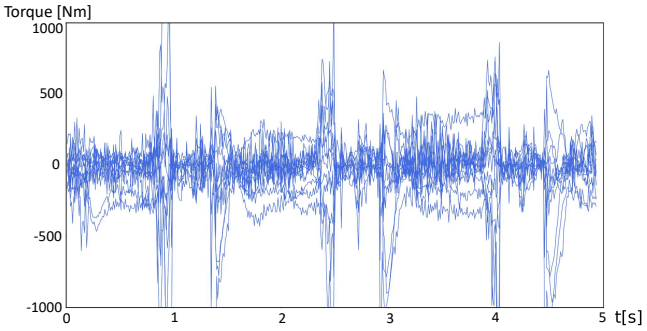

Fig. 11. Joints torque of Atlas robot during continous rotation jumping.

References

1. Atlas open-source robot model. https://github.com/team-vigir, accessed: 2022-05-01
2. Boston dynamics atlas. https://www.bostondynamics.com/atlas, accessed: 2022-05-01

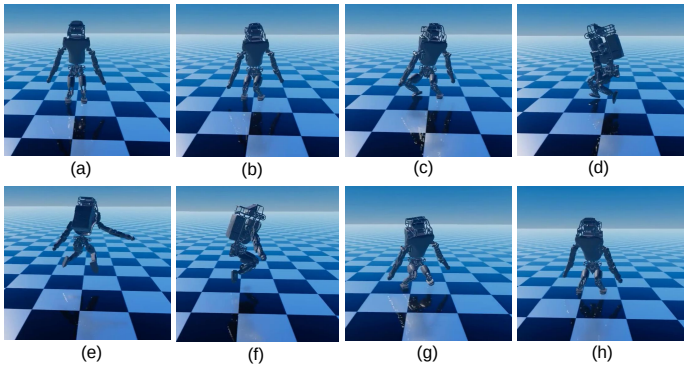

Fig. 12. Learned controller on a 360° rotation jumping.

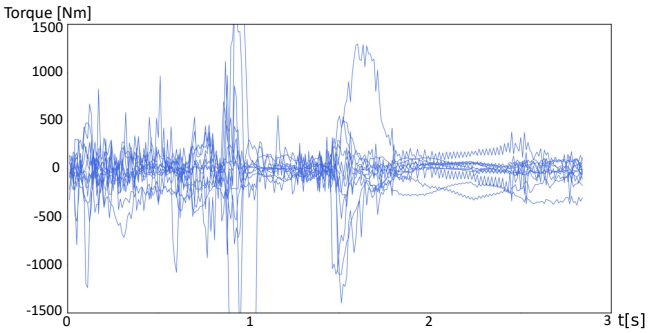

Fig. 13. Joints torque of Atlas robot during a 360° rotation jumping.

3. Kojio, Y., Ishiguro, Y., Sugai, F., Kakiuchi, Y., Okada, K., Inaba, M., et al.: Unified balance control for biped robots including modification of footsteps with angular momentum and falling detection based on capturability. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 497–504. IEEE (2019)