Supplementary Materials: Minimizing Energy Consumption Leads to the Emergence of Gaits in Legged Robots

Zipeng Fu CMU
Ashish Kumar UC Berkeley
Jitendra Malik UC Berkeley
Deepak Pathak CMU

All result videos and analysis are compiled on the project webpage: https://energy-locomotion.github.io.

1 Experimental Setup and Training Details

Hardware Details: We use Unitree's A1 robot as our hardware platform [1]. A1 is a medium sized quadruped with 18 DoFs out of which 12 are actuated. Its mass is about 12 kg. We measure the joint positions and velocities of the 12 joints from the motor encoders and roll and pitch angles from the IMU sensor. To sense the foot contacts with ground in the real world, the foot sensor readings are taken on-board. The feet of the robot are deformable rubber membranes with filled air and air pressure sensors attached. When the foot membranes are deformed, readings from the pressure sensor will increase. We binarize the pressure reading by setting a threshold. Whenever the pressure reading from a foot is above that threshold, we treat it as a foot contact. The deployed policy uses position control for the joints of the robots. We use a PD controller to convert target joint position to torque with fixed gains (K_p = 55 and K_d = 0.8).

Simulation Setup: We use the A1 URDF [1] to simulate the A1 robot in the RaiSim simulator [2]. We generate complex terrains using the inbuilt fractal terrain generator (`flat terrain for structured gait emergence`: number of octaves = 2, fractal lacunarity = 2.0, fractal gain = 0.25, frequency = 10Hz, amplitude = 0.23m; `uneven terrain for unstructured gait emergence`: number of octaves = 2, fractal lacunarity = 2.0, fractal gain = 0.25, frequency = 20Hz, amplitude = 0.27m). We simulate each episode for a maximum of 1000 steps and terminate the episode earlier if the height of the robot drops below 0.28m, magnitude of the body roll exceeds 0.4 radians or the pitch exceeds 0.2 radians. The control frequency of the policy is 100Hz, and the simulation time step is 0.0025s.

Environmental Variations: All environmental variations with their ranges are listed in Table 1 of the main paper. The environment information of e_t in RMA [3] includes center of mass position and the payload (3 dimensions), motor strength (12 dimensions), friction (1 dimension), linear speed in x direction v_x (1 dimension), linear speed in y direction v_y (1 dimension) and yaw speed ω_{yaw} (1 dimension), making it a 19-dim vector. We smooth v_x, v_y and ω_{yaw} by using exponential averaging with values of history time steps and a smoothing factor of 0.2.

State Space: The state is 30 dimensional containing the joint positions (12 dimensions), joint velocities (12 dimensions), roll and pitch angles of the torso, and binarized foot contact indicators (4 dimensions). We did not use any classical state estimator to measure the base velocity or orientation.

Action Space: The action space is 12 dimensional corresponding to the target joint position for the 12 robot joints. The speed up policy learning, the policy network outputs the delta target joint positions, which are converted to target joint positions by adding the joint angles of the initial standing state. The delta target joint positions at HAA, HFE and KFE are restricted to $[-0.15, 0.15]$, $[-0.4, 0.4]$ and $[-0.4, 0.4]$ in radians respectively. The predicted 12-dim target joint angles $a = \hat{\boldsymbol{q}}$ is converted to torques $\boldsymbol{\tau}$ using a PD controller: $\boldsymbol{\tau} = K_p\left(\hat{\boldsymbol{q}} - \boldsymbol{q}\right) + K_d\left(\hat{\dot{\boldsymbol{q}}} - \dot{\boldsymbol{q}}\right)$. K_p and K_d are manually-specified gains, and the target joint velocities $\hat{\dot{\boldsymbol{q}}}$ are set to 0.

5th Conference on Robot Learning (CoRL 2021), London, UK.

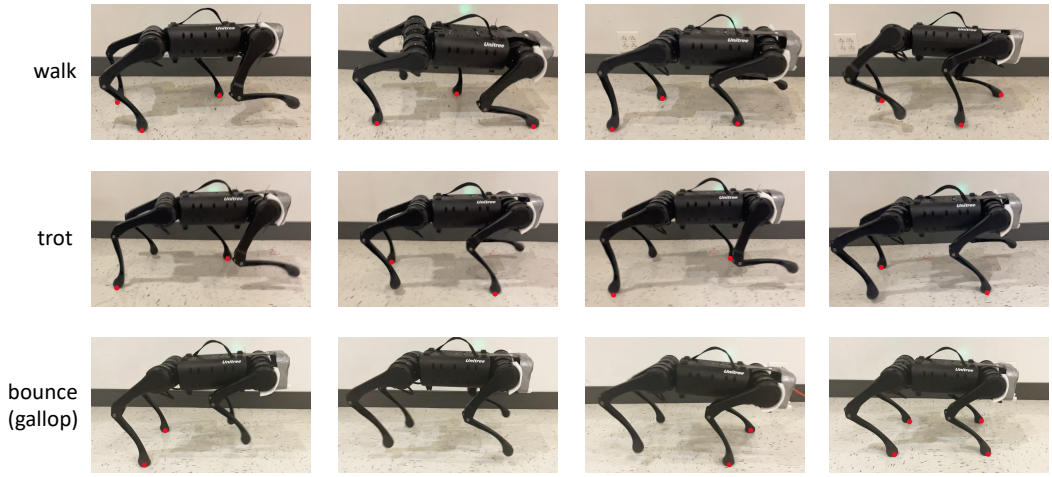

Figure 1: Key frames of the walking gait, trotting gait and bouncing (galloping) gait. Red dots indicate foot contacts. Videos: https://energy-locomotion.github.io

Figure 2: Walking with 1 kg payload (two bottles of 500 ml water are strapped on the back of the robot). The 1 kg payload is out of the normal perturbation of environment parameters used in simulation training shown in Table 1 in the main paper.

Reward: We use the energy consumption-based reward throughout all settings. On uneven terrain for unstructured gaits, we can add the extra penalties from [3] like minimizing ground impacts to explicitly reduce the risks of hardware wearing because complex terrain can otherwise damage the hardware quickly. However, these extra penalties are not responsible for the unstructured gait emergence in complex uneven terrain.

Hyperparameters of Policy Learning: PPO [4] and Adam optimizer [5] are used to train the base policy and the environmental factor encoder in RMA [3]. The total number of training epoch is $15,000$. During each epoch, a batch of $100,000$ state-action transitions, which is split into 4 mini-batches, is sampled from the current policy. Each mini-batch is used for 4 times to compute the loss and backpropagation. The total loss is the surrogate policy loss plus half the value loss. The action log probability ratio is clipped between 0.8 and 1.2, and the target value is clipped to $0.8 - 1.2$ times the range of the value that is computed in previous iteration. We lower bound the standard deviation of the parameterized Gaussian action space to 0.2 to encourage exploration. We set λ and γ in the generalization advantage estimation [6] to 0.95 and 0.998 respectively. We set the learning rate of the optimizer to $5\text{e}{-4}$, β to $(0.9, 0.999)$, and ϵ to $1\text{e}{-8}$.

2 Additional Experiment Results

2.1 Emergence of Walking, Trotting and Bouncing

In Figure 1 of the Supplementary, we show four key frames of the walking at low target speed (0.375 m/s), trotting at median target speed (0.9 m/s) and bouncing (galloping) gaits at high target speed (1.5 m/s). At high speed, the robot gallops for the first few seconds, where the front two legs leave the ground first and also contact the ground first after the flight phase. Then, it switches to using bouncing gait where the four legs hit the ground at approximately the same time.

2.2 Robustness Tests

In addition to Figure 5 of the main text, Figure 2 in the Supplementary shows the the key frames of walking gait under 1 kg payload. The walking gait is robust to the extra payload that is out of the normal perturbation of environment parameters used in simulation training shown in Table 1 of the main paper. For more qualitative results on robustness, please refer to videos at https://energy-locomotion.github.io.

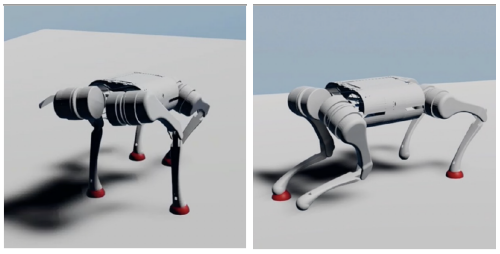

Figure 3: Two examples of emerged failure gaits when trained on flat ground without fractal variations. Both failure gaits are tilted and unstable.

3 Additional Ablation Studies

Fractal Terrain during Training In Figure 3, we show the key frames of 2 gaits at low and median speeds when the policies are trained with energy minimization (Section 2.4) but on a flat terrain without fractal perturbations (Section 2.3). All 10 training trials converge to unnatural and unstable gaits. We found that adding fractal terrain also facilitates a larger foot clearance and robustness which improves real-world hardware deployment.

4 Details of the MPC baseline

We use a reference implementation [7] of convex MPC [8] for Unitree A1. The parameters for gait generation to enforce constraints on ground reaction forces are fine-tuned for walking gait, trotting gait and bouncing gait. For the walking gait, we test at target speeds v^{target} in the range from 0.1m/s to 0.7m/s every 0.1m/s. We set the duty factors of 4 legs to $[0.8, 0.8, 0.8, 0.8]$, initial leg phases to $[0, 0.25, 0.5, 0]$, and the stance duration to $-0.5v^{\text{target}} + 0.75$ seconds. Only the Rear-Left foot is initialized in swing phase. For the trotting gait, we test at target speeds v^{target} in the range from 0.5m/s to 1.5m/s every 0.1m/s. We set the duty factors of 4 legs to $[0.6, 0.6, 0.6, 0.6]$, initial leg phases to $[0.9, 0, 0, 0.9]$, and the stance duration to $-0.15v^{\text{target}} + 0.325$ seconds. The Front-Right and Rear-Left feet are initialized in swing phase. For the bouncing gait, we test at target speeds v^{target} in the range from 1.0m/s to 2.0m/s every 0.1m/s. We set the duty factors of 4 legs to $[0.35, 0.35, 0.35, 0.35]$, initial leg phases to $[0, 0, 0, 0]$, and the stance duration to 0.04 seconds. All feet are initialized to the stand phase.

5 Details of Gait Transition

We first train our fixed-velocity policies as described in the paper which lead to a walking policy π_{walking}, a trotting policy π_{trotting} and a bouncing (galloping) policy π_{bouncing}. Each gait policy has its own encoder μ to embed the environmental parameters e_t which are only available during simulation. We then treat these policies as experts and collect demonstration data from them to self-supervise and bootstrap the initial training phase of the velocity-conditioned policy Π. We represent the command velocity inputs at these three velocity modes as three one-hot vectors of length three. Notice that these three policies serve as experts at three different velocity modes at low (0.375 m/s), median (0.9 m/s) and high (1.5 m/s), but there is no expert demonstrations at other target velocities sampled from the continuous range of 0.375 m/s - 1.5 m/s. To learn motor skills and smooth gait transition at continuous intermediate velocities, we rely on the velocity-conditioned RL rewards, resample the target velocity every 200 steps, and represent the command velocity inputs as interpolations between the three velocity modes (e.g. 1.2 m/s is represented as [0, 0.5, 0.5] and 0.5 m/s is represented as [0.238, 0.762, 0]). The velocity-conditioned policy Π cannot rely on environmental parameters, since these environmental parameters are not available when the policy is deployed on the robot. Instead, the velocity-conditioned policy Π relies on 20 history steps of observations (x_{t-1} to x_{t-20}) and actions (a_{t-2} to a_{t-21}) which implicitly contain information about environmental parameters and are encoded by the adaptation module ϕ into a 8-dimensional vector fed to Π. The velocity-conditioned policy Π is trained by minimizing self-supervised L2 loss at the three velocity modes minus expected returns at randomly sampled velocities from the continuous range of 0.375 m/s - 1.5 m/s. We linearly anneal the L2 loss to zero during initial 2500 training

epochs, and then velocity-conditioned policy is optimized with only RL loss at randomly sampled velocities for additional 2500 training epochs.

References

[1] X. Wang. Unitree Robotics. https://www.unitree.com/.

[2] J. Hwangbo, J. Lee, and M. Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 2018. URL www.raisim.com.

[3] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. In *Robotics: Science and Systems*, 2021.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[6] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations*, 2016.

[7] E. Coumans, X. B. Peng, and Y. Yang. Convex MPC controller for A1. https://github.com/google-research/motion_imitation/.

[8] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018.