A More about Random Pattern Generator

Figure 7 illustrates the steps for the random pattern generator to create a template for G = PACING. To generate a template in general, we first sample $T \in [24, 28]$ (step 1). Since the control frequency is 50 Hz, this corresponds to a cycle length of $0.48 \sim 0.56$ seconds. We then sample a foot-ground contact length ratio within the cycle $r_{\text{contact}} \in [0.5, 0.7]$, Tr_{contact} therefore gives the number of '1's and $T(1 - r_{\text{contact}})$ the number of '0's in each row (step 2). Proper length scaling and bit shifts of these ones and zeros are necessary to produce feasible foot contact patterns on a real robot (step 3). For G = BOUND, we shorten the foot-ground contact time to 60% of the sampled value (i.e., $r_{\text{contact}} = 0.6r_{\text{contact}}$), we place the ones at the beginning of the FL and FR rows and shift those in the RL and RR rows by $0.5Tr_{\text{contact}}$ bits to the right. We do no scaling for G = TROT. Finally, we shift the '1's to form complete templates (step 4): we place the ones at the beginning in the FL and RR rows and at the end of the FR and RL rows. We keep r_{contact} untouched for G = PACE, but shrink the cycle length to half its sampled value (i.e., $T = 0.5T$) to make the gait natural and feasible. We place the ones at the beginning in the FL and RL rows and at the end of the FR and RR rows. Finally, for $G \in \{\text{STAND\_STILL}, \text{STAND\_3LEGS}\}$, we perform no scaling and fill in the pattern template matrix with ones. We randomly sample one row and replace it with zeros if G = STAND_3LEGS.

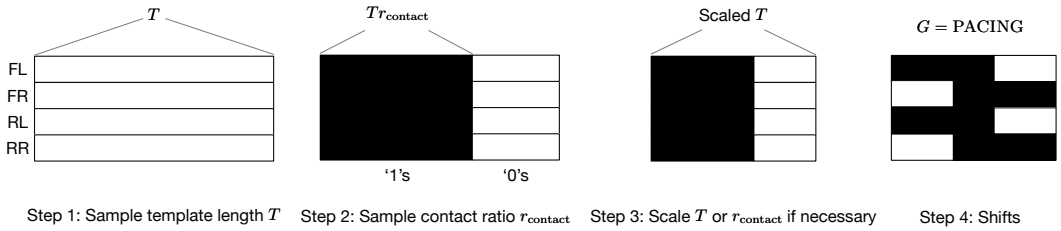

Figure 7: How the random pattern generator works.

B Reward Design

Our reward design is based on those in legged gym [41]. The total reward consists of 8 weighted reward terms: $J = \sum_{i=1}^{8} w_i r_i$, where w_i's are the weights and r_i's are the rewards. The definition of each reward term and the value of the weights are in the following. We put the purpose of each reward term in the bracket at the beginning of the description.

- [Task Reward] Linear velocity tracking reward. $r_1 = e^{-4\times((v_x-\hat{v}_x)^2+v_y^2)}$, where v_x and $\hat{v}_x$ are the current and desired linear velocities along the robot's heading direction, and v_y is the current linear velocity along the lateral direction. All velocities are in the base frame, and $w_1 = 1$.
- [Task Reward] Angular velocity tracking reward. $r_2 = e^{-4\times\omega_z^2}$, where ω_z is the current angular yaw velocity in the base frame and $w_2 = -0.5$.
- [Task Reward] Penalty on foot contact pattern violation. $r_3 = \frac{1}{4}\sum_{i=1}^{4} |c_i - \hat{c}_i|$, where $c_i, \hat{c}_i \in \{0, 1\}$ are the realized and desired foot-ground contact indicators for the i-th foot, and $w_3 = -1$.
- [Sim-to-Real] Regularization on action rate. $r_4 = \sum_{i=1}^{12}(a_t - a_{t-1})^2$ where a_t and a_{t-1} are the controller's output at the current and the previous time steps, and $w_4 = -0.005$.
- [Sim-to-Real] Penalty on roll and pitch angular velocities. We encourage the robot's base to be stable during motion and hence $r_5 = \omega_x^2 + \omega_y^2$, where ω_x and ω_y are the current roll and pitch angular velocities in the base frame. This penalty does not apply to G = BOUND and $w_5 = -0.05$.
- [Sim-to-Real] Penalty on linear velocity along the z-axis. Similar to the previous term, we use this term to encourage the base stability during motion. $r_6 = v_z^2$ where v_z is the current linear velocity along the z-axis in the base frame. This penalty does not apply to G = BOUND either and $w_6 = -2$.
- [Natural Motion] Penalty on body collision. $r_7 = \sum_{i=1}^{K} \mathbb{1}\{F_i > 0.1\}$, where F_i is the contact force on the i-th body. In our experiments $K = 8$ (i.e., 4 thighs and 4 calves) and $w_7 = -1$.

- [Natural Motion] Penalty on deviation from the default pose. $r_8 = \sum_{a_t \in \text{hip}} |a_t|$, where a_t's are the actions (i.e., deviation from the default joint position) applied to the hip joints, and $w_8 = -0.03$.

C Training Configurations

C.1 Control

We use PD control to convert positions to torques in our system. The bases value for the 2 gains are $k_p = 20$ and $k_d = 0.5$. Our control frequency is 50 Hz.

C.2 Gait Sampling

We randomly assign a gait G to a robot at environment resets, and also samples it again every 150 steps in simulation. Of the 5 G's, some gaits are harder to learn than others. To avoid the case where the hard-to-learn gaits die out, leaving the controller to learn only on the easier gaits, we restrict the sampling distribution such that the ratio of the 5 G's are always approximately the same.

C.3 Reinforcement Learning

We use the Proximal policy optimization (PPO) [43] algorithm as our reinforcement learning method to train the controller. In our experiments, PPO trains an actor-critic policy. The architecture of the actor is introduced in Section 3.2.3, and the critic has the identical network architecture except that (1) its output size is 1 instead of 12, and (2) it also receives the base velocities in the local frame as its input. We keep all the hyper-parameters the same as in [41] and train for 1000 iterations. For safety reasons, we end an episode early if the body height of the robot is lower than 0.25 meters. Training can be done on a singe NVIDIA V100 GPU in approximately 15 minutes.

C.4 Domain Randomization

During training, we sample noises $\epsilon \sim \text{Unif}$, and add them to the controller's observations. We use PD control to convert positions to torques in our system, and domain randomization is also applied to the 2 gains k_p and k_d. Table 3 gives the components where noises ϵ were added and their corresponding ranges.

Table 3: Domain randomization settings.

#	Component	Noise Range
1	Base linear velocities	$[-2, 2]$
2	Base angular velocities	$[-0.25, 0.25]$
3	Gravity vector in the base frame	$[-1, 1]$
4	Joint positions	$[-1, 1]$
5	Joint velocities	$[-0.05, 0.05]$
6	k_p	$[-5, 0]$
7	k_d	$[0, 0.25]$

D More Images from the Extended Tests

Figure 8: Images from the extended tests. We show the command for each test on the top-left corner on each row. LLM translated foot contact patterns are shown at the bottom-right corner in each image. Motions better viewed in the supplementary video.