Deep Reinforcement Learning for Multi-contact Motion Planning of Hexapod Robots

Huiqiao Fu[1,2], **Kaiqiang Tang**[1], **Peng Li**[2], **Wenqi Zhang**[2], **Xinpeng Wang**[3], **Guizhou Deng**[3], **Tao Wang**[2] and **Chunlin Chen**[1*]
[1]School of Management and Engineering, Nanjing University, China
[2]Advanced Institute of Information Technology (AIIT), Peking University, China
[3]Southwest University of Science and Technology, China
{hqfu, kqtang}@smail.nju.edu.cn, {pli, wqzhang}@aiit.org.cn,
wangtao@pku.edu.cn, {xpwang, gzdeng}@mails.swust.edu.cn, clchen@nju.edu.cn

Abstract

Legged locomotion in a complex environment requires careful planning of the footholds of legged robots. In this paper, a novel Deep Reinforcement Learning (DRL) method is proposed to implement multi-contact motion planning for hexapod robots moving on uneven plum-blossom piles. First, the motion of hexapod robots is formulated as a Markov Decision Process (MDP) with a specified reward function. Second, a transition feasibility model is proposed for hexapod robots, which describes the feasibility of the state transition under the condition of satisfying kinematics and dynamics, and in turn determines the rewards. Third, the footholds and Center-of-Mass (CoM) sequences are sampled from a diagonal Gaussian distribution and the sequences are optimized through learning the optimal policies using the designed DRL algorithm. Both of the simulation and experimental results on physical systems demonstrate the feasibility and efficiency of the proposed method. Videos are shown at https://videoviewpage.wixsite.com/mcrl.

1 Introduction

Legged robots have redundant degrees of freedom and multiple footholds for passing through challenging environments, and have wide application prospects in disaster rescue, material transportation, planet exploration and other fields [Lee *et al.*, 2020]. However, it is still a challenging task to improve the motion efficiency of the legged robots in unstructured environments. Traditional methods focus on the single-step optimization using kinematic criteria and ignore the global footstep planning, which always leads to a poor passibility in complex environments [Belter *et al.*, 2016; Hwangbo *et al.*, 2019]. Discrete environments are a special case of the unstructured environment which need more efficient and reliable planning methods. These kind of environments can characterize any unstructured environment for the motion planning of legged robots.

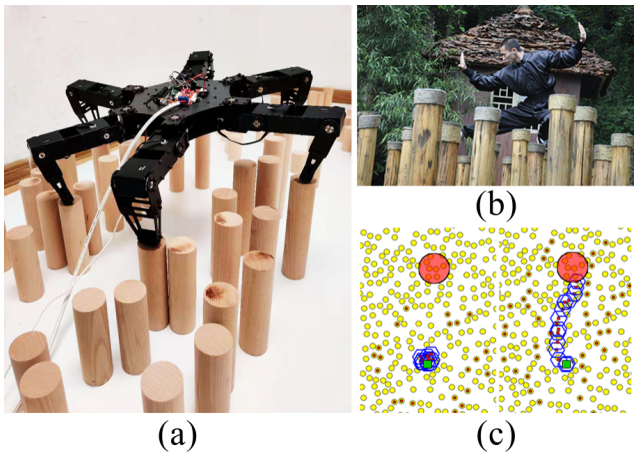

Figure 1: The plum-blossom piles. (a) Plum-blossom piles used in the real environment. (b) Plum-blossom piles in Chinese kung fu. (c) Plum-blossom piles used in the training process with the initial policy (left) and the trained policy (right). The green block is the initial point, the red circle is the target area, yellow dots represent plum-blossom piles, red dots represent footholds and CoM sequences, blue wireframes represent the body of the hexapod robot.

Trajectory optimization (TO) [Betts, 1998] has been proved to be effective for legged locomotion. This TO method generates a motion trajectory that minimizes some measure of performance with a set of constraints. For the multi-contact motion planning problem in discrete environments, such as plum-blossom piles shown in Fig.1, TO is difficult to solve directly and usually needs to be combined with some traditional planning methods, such as A*, Probabilistic Roadmap (PRM) planner, Rapidly-exploring Random Tree (RRT). These methods can plan a feasible trajectory quickly when the dimension of the state and action space are low, but often face the curse of dimensionality problem in the high dimensional or continuous state and action space.

Reinforcement Learning (RL) [Sutton and Barto, 2018] is a machine learning methodology that has witnessed great progress in the artificial intelligence community. Recent breakthroughs of Deep Reinforcement Learning (DRL) algorithms have proved RL to be one of the state-of-the-art

*Contact Author

technologies for complex learning and control tasks [Mnih *et al.*, 2015; Silver *et al.*, 2016]. Based on the framework of Markov Decision Processes (MDP), RL addresses the problem that how an autonomous active agent learns the optimal policies while interacting with an initially unknown environment. The self-learning property from unknown environments makes RL a promising candidate for the optimization and control of real systems [Arulkumaran *et al.*, 2017]. Nevertheless, to the best of our knowledge, the DRL-based multi-contact motion planning for hexapod robots on plum-blossom piles has not been well addressed yet.

In this paper, we develop an integrated DRL-based method for multi-contact motion planning of hexapod robots moving on uneven and randomly generated plum-blossom piles. Our main contributions are threefold.

- We formulate the motion of hexapod robots as a discrete-time finite MDP problem with a specified reward function. The motion policies are optimized using the designed DRL algorithm.
- We build a multi-contact centroid dynamics model for hexapod robots, and formulate the transition feasibility of the state transition using the TO method, which in turn determines the rewards.
- We test the trained policies on different settings of plum-blossom piles, and both of the simulation and experimental results demonstrate the feasibility and efficiency of the proposed method.

2 Related Work

Multi-contact motion planning. Multi-contact motion planning is an important subject in robotics and there have been a variety of planning strategies for legged systems. Some focus on simply choosing the next best reachable footholds ignoring the global footstep planning [Rebula *et al.*, 2007], while others consider the optimal footstep sequence from the start to the goal [Zucker *et al.*, 2011; Mastalli *et al.*, 2015]. Recently, [Ding *et al.*, 2020] used the Monte Carlo tree search algorithm to generate reliable gait and foothold sequences for hexapod robots in a sparse foothold environment. But most of the above methods only consider the kinematic criteria to select footholds. In recent years, the multi-contact TO method has attracted extensive attention in the field of legged locomotion. [Winkler, 2018] used the simplified centroid dynamics and TO to generate a motion trajectory with rich possible behaviors. [Mastalli *et al.*, 2017] searched in the elevation map for legged locomotion on rough terrains using TO.

Transition feasibility. The transition feasibility describes whether the robot can transfer from the current state to the target state. [Tonneau *et al.*, 2018] solved the problem from the perspective of kinematics and dynamics. They first considered a conservative but exact formulation of the dynamics, and then relaxed while preserving the kinematic constraints of the motion. [Fernbach *et al.*, 2018] proposed an efficient dynamic feasibility check based on a conservative and convex reformulation of the problem, and then solved the problem with a Linear Program (LP). Different from the former, [Klamt and Behnke, 2019] used a CNN to output the feasibility and costs values and generated an abstract representation of a detailed planning problem.

RL for legged motion planning. RL has been proved to be effective for legged motion planning on rough terrains [Rivlin *et al.*, 2020]. Unlike traditional planning methods, DRL can deal with complex tasks in high-dimensional continuous state and action spaces. [Shahriari and Khayyat, 2013] proposed a gait generation strategy based on RL and fuzzy reward, and planned the motion strategy on discontinuous terrain through iterative updating. [Peng *et al.*, 2017] separated the responsibilities for planning footholds and executing swing-leg motions and uses the hierarchical DRL to learn locomotion skills. [Tsounis *et al.*, 2020] combined DRL with the model-based motion planning method, and formulated the MDP using the evaluation of dynamic feasibility criteria in place of physical simulation, so as to realize the motion planning on challenging terrains.

3 DRL for Multi-contact Motion Planning of Hexapod Robots

In this section, the integrated DRL-based multi-contact motion planning method is presented for hexapod robots moving on uneven plum-blossom piles. First, we introduce the overall control structure of our method. Second, the motion of hexapod robots is mathematically formulated as an MDP with a specified reward function. Third, a transition feasibility model is proposed, which describes the feasibility of the state transition under the condition of satisfying kinematics and dynamics, and in turn determines the rewards. Finally, using the defined MDP, we propose a DRL-based algorithm for multi-contact motion planning of hexapod robots.

3.1 Overall Control Structure

The task of hexapod robots is to move from the initial point to the target area on uneven plum-blossom piles. We formulate this process as a discrete-time finite MDP. The overall control structure are shown as in Fig. 2. The current state s_t, containing the proprioceptive information $\mathbf{\Phi}_t$, the exteroceptive information $\mathbf{M}_p$ and the target $\mathbf{p}_{\text{target}}$, is first input into a policy network π_θ. We parameterize π_θ as a diagonal Gaussian distribution, the mean of which is output by a Neural Network (NN). First, the coordinates of all plum-blossom piles, which are randomly distributed in the environment, are input into a Graph Attention Network (GAT) [Veličković *et al.*, 2018] with sparse matrix operations. The resulting latent output from the coordinates is concatenated with the remaining part of the state, then fed into a Multilayer Perceptron (MLP). The action a_t is sampled from the diagonal Gaussian distribution and next input into the environment.

According to the input of the action a_t, the environment outputs the undetermined next state s'_{t+1}, which contains both target Center-of-Mass (CoM) and footholds positions, where the target footholds are found by the K-nearest Neighbors (KNN) algorithm according to a_t. We propose a transition feasibility model for hexapod robots, which describes the feasibility of the state transition under the condition of

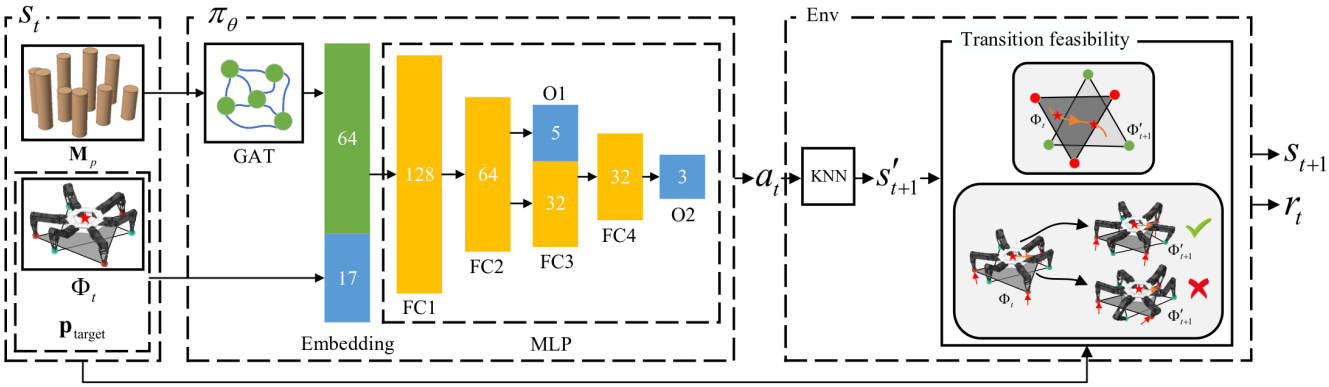

Figure 2: Overview of the proposed control structure.

satisfying kinematics and dynamics, and in turn determines the rewards. The input of the transition feasibility model is s_t and s'_{t+1}, and the output is the next state s_{t+1} and the reward r_t. If the the hexapod robot can transform form s_t to s'_{t+1} according the transition feasibility model, then $s_{t+1} = s'_{t+1}$ and r_t is positive, otherwise $s_{t+1} = s_t$ and r_t is negative. The optimal footholds and CoM sequences are obtained using the optimized policy π^*_θ, which is trained by the designed DRL algorithm. Finally, the hexapod robot follows the optimal sequences by inverse kinematics.

3.2 MDP Formulation

The motion of hexapod robots on plum-blossom piles can be described as a discrete-time infinite MDP of a 4-tuple $\langle S, A, P, R\rangle$, where S is the state space, A is the action space, P is the state transition probability and R is the reward function. In a RL process, an agent interacts with the environment to maximize the cumulative discounted future rewards: $R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$, where $\gamma \in [0, 1]$ is a discount factor and T is the terminate time.

State Space

The state space contains both exteroceptive and proprioceptive measurements. The hexapod robot moves forward with the tripod gait on plum-blossom piles, and there are three legs in contact and others in swing at each time-step t. The state s_t can be expressed as

$$\begin{aligned} s_t &:= \langle \mathbf{M}_p, \mathbf{\Phi}_t, \mathbf{p}_{\text{target}} \rangle, \\ \mathbf{\Phi}_t &:= \langle \mathbf{r}_{Bt}, \boldsymbol{\theta}_t, \mathbf{p}^c_t, \mathbf{c}_t \rangle, \end{aligned} \tag{1}$$

where $\mathbf{M}_p \in \mathbb{R}^{3\times N_p}$ is the coordinates of all the plum-blossom piles in the environment, N_p is the number of plum-blossom piles, $\mathbf{\Phi}_t$ is the proprioceptive information which contains the CoM position of the hexapod robot $\mathbf{r}_{Bt}$, the Euler angles of the body $\boldsymbol{\theta}_t$, the foothold position of the i_{th} contact leg $\mathbf{p}^c_t$ and the state of each leg $\mathbf{c}_t \in \{0, 1\}$. If $\mathbf{c}_t = 0$, legs 1, 3 and 5, as shown in Fig. 3, are in contact. If $\mathbf{c}_t = 1$, legs 2, 4 and 6 are in contact. $\mathbf{p}_{target}$ is the center coordinate of the target area. At each moment when the swing legs and the support legs switched, we constrain the linear velocities $\dot{\mathbf{r}}_{Bt}$ and the angular velocities $\boldsymbol{\omega}_t$ of the hexapod robot to zero.

Action Space

The action at time-step t can be expressed as

$$a_t := \langle \Delta\mathbf{r}_{Bt}, \Delta\boldsymbol{\theta}_t, \mathbf{k}_t \rangle, \tag{2}$$

where $\Delta\mathbf{r}_{Bt} \in \mathbb{R}^3$ is the increment of coordinates of the CoM in the world frame, $\Delta\boldsymbol{\theta}_t \in \mathbb{R}^2$ is the roll angle and pitch angle increment of the body, and the yaw angle increment of the body is zero. $\mathbf{k}_t \in (-1, 1)$ is the selection parameters for the landing points of the swing legs. Specifically, when the state of the CoM gets the next desired state at time-step $t+1$, according to the simplified kinematic model of the hexapod robot as shown in Fig. 3, we can determine the center of the kinematic cube $\bar{\mathbf{p}}_i$ for the i_{th} swing leg. Then we can find the n_p nearest plum-blossom piles to $\bar{\mathbf{p}}_i$ in the environment using KNN. The piles are then arranged in the ascending order from near to far. Finally, the k_p-th plum-blossom pile is selected as the target landing point p^s_i of the i_{th} swing leg at the next time-step $t+1$.

$$k_p = \left\lfloor n_p \cdot \frac{k_{it} + 1}{2} \right\rfloor, \tag{3}$$

where $\lfloor * \rfloor$ denotes the rounding down operation.

Reward Function

The goal of the hexapod robot moving on plum-blossom piles is to reach the target area from the initial point with the shortest path while satisfying all the kinematic and dynamic constraints. We design the reward function as follows:

$$r_t = r_{bt} + r_{kt} + r_{ft} + r_{dt} + r_{gt}, \tag{4}$$

where r_{bt} is the boundary reward penalizing the CoM moving beyond the boundary of the environment, r_{kt} penalizes the state transition violating the kinematic constraint, r_{ft} penalizes the unfeasible state transition calculated by the transition feasibility model. The distance reward forces the robot to move to the target area with the shortest path and is defined as

$$r_{dt} = -\frac{\| r_{Bt} - p_{\text{target}} \|^2}{\| p_{\text{initial}} - p_{\text{target}} \|^2}, \tag{5}$$

where p_{initial} is the initial point. The hexapod robot gets the reward r_{gt} when arriving the target area.

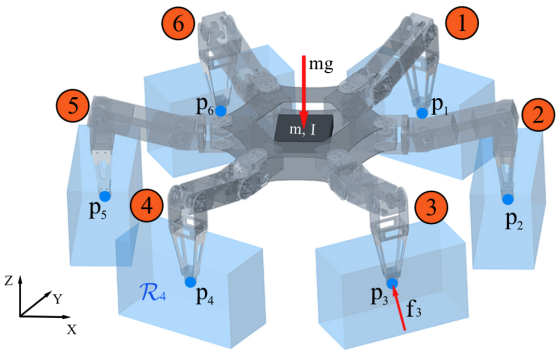

Figure 3: The hexapod robot model used in the transition feasibility. $i \in \{1, ..., 6\}$ is leg number. The kinematic model of the hexapod robot is conservatively approximated by keeping the foot p_i inside the blue cube $\mathcal{R}_i$. The dynamics are approximated by a single rigid-body with mass m and inertia $\mathbf{I}$. The external force includes the thrust on the i_{th} foot $\mathbf{f}_i$ and the gravity on the CoM $m\mathbf{g}$.

3.3 Transition Feasibility

We employ the TO technique to determine the transition feasibility of the state transition, and in turn determine the rewards. In this subsection, we first introduce the formulation of the multi-contact transition feasibility. Then, we give the hexapod robot model and the multi-contact model used in the transition feasibility.

Multi-contact Transition Feasibility Formulation

The multi-contact transition feasibility model determines whether the state transition of the hexapod robot is feasible given the current state $\mathbf{x}_0$ and the target state $\mathbf{x}_T$. The state $\mathbf{x}$ includes the CoM position $\mathbf{r}_{Bt}$, the linear velocities $\dot{\mathbf{r}}_{Bt}$, the Euler angles $\boldsymbol{\theta}_t$ and the angular velocities $\boldsymbol{\omega}_t$.

We formulate the transition feasibility as a nonlinear constrained optimization problem using the direct multiple shooting method and solve it by Nonlinear Programming (NLP). The initial state $\mathbf{x}_0$, the desired final state $\mathbf{x}_T$, the foothold position of the i_{th} contact leg $\mathbf{p}_i^c$, the target foothold position of the i_{th} swing leg $\mathbf{p}_i^s$, and the total duration T are provided, where i from 1 to n_i is the i_{th} contact leg. On plum-blossom piles, the hexapod robot takes the tripod gait as the default gait to complete the state transition, therefore the number of contact legs $n_i = 3$.

The hexapod robot model used in the transition feasibility is shown in Fig. 3. The decision variables of the problem include the contact force $\mathbf{f}_t$ and the state $\mathbf{x}_t$. The constrains include the initial state $\mathbf{x}_{t_0} = \mathbf{x}_0$, the target state $\mathbf{x}_{t_F} = \mathbf{x}_T$, the dynamic model, the kinematic model, the pushing force and the friction cone. The optimizer uses the provided information to find a trajectory for the state $\mathbf{x}$ and the contact force $\mathbf{f}$, and makes the objective function

$$J = \int_0^T \sum_{i=1}^{n_i} \mathbf{f}_i^2 dt \tag{6}$$

meet the local minimum. We use $F_{tf} \in \{0, 1\}$ to describe the state transition. If the trajectory satisfies all the given constraints, then the transition is feasible and $F_{tf} = 1$, otherwise the transition is unfeasible and $F_{tf} = 0$.

Name	Body	Coxa	Femur	Tibia
Length/mm	238	60	120	130
Range $1/^o$	-	[-45, 45]	[0, 45]	[-135, -90]
Range $2/^o$	-	[-45, 45]	[-45, 0]	[-90, -45]

Table 1: Dimension parameters and joint rotation ranges of the hexapod robot.

Kinematic Model

The kinematic model describes the workspace of the body and foots, which avoids the behavior of the hexapod robot violating its own mechanical structure constraints. Since the original kinematic model of the hexapod robot is highly nonlinear, we conservatively approximate it by keeping the i_{th} foot p_i inside the blue cube $\mathcal{R}_i$ in Fig. 3.

In order to get the cubes, based on the dimension parameters of the hexapod robot and the rotation range of each joint shown in Table 1, we randomly sample within the rotation range to generate the point cloud of the foots. According to the point cloud, we can conservatively find the side length of the cube. The workspace of each foot i can be expressed as

$$\begin{aligned} &\mathbf{p}_i \in \mathcal{R}_i \left(\mathbf{r}_B, \boldsymbol{\theta}\right) \\ \Leftrightarrow &\left|\mathbf{R}_Z\left(\alpha_i\right)\left[{}_W^B\mathbf{R}\left[\mathbf{p}_i - \mathbf{r}_B\right] - \bar{\mathbf{p}}_i\right]\right| < \mathbf{b}, \end{aligned} \tag{7}$$

where ${}_W^B\mathbf{R}$ is the rotation matrix from the world frame to the body frame, $\mathbf{R}_Z(\alpha_i)$ is the rotation matrix for rotations around the z-axis, α_i is the deflection angle of the i_{th} coxa relative to the x-axis in the body frame, $\bar{\mathbf{p}}_i$ is the center of the i_{th} cube, $\mathbf{b}$ is half the side length of the cube. For the three contact legs, the kinematic constraints need to be met in the whole state transition process. For the three swing legs, the kinematic constraints are only verified at the final time T.

Dynamic Model

The dynamic model represents the time-dependent aspects of a system, and we approximate it by Single Rigid Body Dynamics (SRBD) [Winkler, 2018]. Then, we can get the Newton-Euler equations of the hexapod robot, which is defined as the SRBD:

$$m\ddot{\mathbf{r}}_B = \sum_{i=1}^{n_i} \mathbf{f}_i + m\mathbf{g}, \tag{8}$$

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) = \sum_{i=1}^{n_i} \mathbf{f}_i \times \left(\mathbf{r}_B - \mathbf{p}_i^c\right), \tag{9}$$

where m is the mass of the hexapod robot, g is the acceleration of gravity, $\ddot{\mathbf{r}_B}$ is the linear acceleration of the CoM. $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the moment of inertia.

The dynamic model of the hexapod robot can be modeled by an ordinary differential equation $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{f})$. The rates of the Euler angles $\dot{\boldsymbol{\theta}}$ can be calculated by the optimized Euler angles $\boldsymbol{\theta}$ and the angular velocities $\dot{\boldsymbol{\omega}}$:

$$\dot{\boldsymbol{\theta}} = C(\boldsymbol{\theta})\dot{\boldsymbol{\omega}} = \begin{bmatrix} 1 & 0 & -\sin\theta_y \\ 0 & \cos\theta_x & \sin\theta_x\cos\theta_y \\ 0 & -\sin\theta_x & \cos\theta_x\cos\theta_y \end{bmatrix} \dot{\boldsymbol{\omega}}. \tag{10}$$

Based on the SRBD, the dynamic model of the hexapod robot is independent of its joint state, but only related to the external forces on the contact legs.

Pushing Force

According to the physics, the force provided by the environment to the hexapod robot can only be thrust, and we set up the following constraint:

$$\mathbf{f}_i \cdot \mathbf{n}\left(\mathbf{p}_i^c\right) \geq 0, \tag{11}$$

where $\mathbf{n}\left(\mathbf{p}_i^c\right)$ is the normal vector of the environmental surface at coordinate $\mathbf{p}_i^c$.

Friction Cone

The friction follows from Coulomb's law pushing stronger into a surface allows exerting larger side-ways forces without slipping. Therefore, the resultant force on each contact leg is always in the interior of the friction cone. The linear approximation is as follows:

$$\left|\mathbf{f}_i \cdot \mathbf{t}_{\{1,2\}}\left(\mathbf{p}_i^c\right)\right| < \mu \cdot \mathbf{f}_i \cdot \mathbf{n}\left(\mathbf{p}_i^c\right), \tag{12}$$

where $\mathbf{t}_{\{1,2\}}\left(\mathbf{p}_i^c\right)$ is the tangential vector of the environment at coordinate $\mathbf{p}_i^c$ and μ is the friction coefficient.

3.4 DRL-based Multi-contact Motion Planning

Based on the proposed transition feasibility model and the formulated MDP, an integrated DRL algorithm for multi-contact motion planning of hexapod robots moving on uneven plum-blossom piles is shown as in *Algorithm* 1.

Given the current state s_t of the hexapod robot, we design a policy network π_θ to map the state s_t to the action a_t. The architecture of π_θ is shown in Fig. 2. This policy network is parameterized as a diagonal Gaussian distribution $\pi_\theta\left(a_t \mid s_t\right) := N\left(a_t \mid \mu_\theta\left(s_t\right), \sigma\right)$, while the mean $\mu_\theta\left(s_t\right)$ is output by a neural network and the standard-deviation parameters σ are standalone parameters. The coordinates of all the plum-blossom piles $\mathbf{M}_p$ are input into a Graph Attention Network (GAT) [Veličković *et al.*, 2018] with a multi-head attention:

$$\vec{h_i}' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right), \tag{13}$$

where $\vec{h}$ is the input node feature. For the plum-blossom piles, the feature is its own normalized coordinate, $\vec{h}'$ is the output node feature, σ is an activation function, K is the number of the independent attention mechanisms, $\mathbf{W}^k$ is the corresponding input linear transformation's weight matrix, $\mathcal{N}_i$ is the neighborhood of node i in the graph. In our work, $\mathcal{N}_i$ represents the 5 points closest to node i. α_{ij} is the normalized attention coefficient:

$$\alpha_{ij} = \mathrm{softmax}_j\left(e_{ij}\right) = \frac{\exp\left(e_{ij}\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(e_{ik}\right)}. \tag{14}$$

The attention coefficient e_{ij} can be calculated as

$$e_{ij} = \mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T\left[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j\right]\right), \tag{15}$$

where $\vec{\mathbf{a}}^T$ is the weight vector and $\|$ represents concatenation. We use the sparse matrix operations in GAT to reduce the storage complexity.

Algorithm 1 DRL for Multi-contact Motion Planning

```
Initialize policy parameters θ_0 and value function parameters φ_0.
for episod k = 0, 1, 2, ..., M do
    Randomly initialize s_0 and p_target.
    for step t = 0, 1, 2, ..., T do
        Run policy π(θ_k), get action a_t.
        Take a_t, observe s'_{t+1}
        Input s_t, s'_{t+1} to the transition feasibility model and get F_tf.
        if F_tf = 1 then
            Get positive reward r_t, s_{t+1} = s'_{t+1}.
        else
            Get negative reward r_t, s_{t+1} = s_t.
        end if
        Store experience (s_t, a_t, r_t, s_{t+1}) in D_k.
    end for
    Compute rewards-to-go R̂_t.
    Compute Â_t using GAE based on value function V_{φ_k}.
    Update policy: θ_{k+1} = arg max_θ L_a(θ).
    Fit value function: φ_{k+1} = arg min_φ L_c(φ).
end for
```

The output of GAT is concatenated with the remaining part of the state s_t and subsequently input into the Fully-Connected Layer 1 (FC1). Since the posture of the hexapod robot at the next moment will affect the selection of the target foothold positions, we divide the output action into two parts. The action $\Delta\mathbf{r}_{Bt}$ and $\Delta\boldsymbol{\theta}_t$ are output by the Output Layer 1 (O1) and then input into the FC4 with the output of FC3. And the action k_t is output by O2. The hexapod robot executes the complete action a_t , and then observes the next state s_{t+1} and the immediate reward r_t.

To train the policy π_θ, a variant of Proximal Policy Optimization (PPO) [Schulman *et al.*, 2017] using clipped loss and a Generalized Advantage Estimation (GAE) critic is used. We compute the policy update via stochastic gradient ascent with Adam and we have

$$\theta_{k+1} = \arg\max_\theta \mathcal{L}_a(\theta), \tag{16}$$

where

$$\mathcal{L}_a(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)\hat{A}_t, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{17}$$

and $r_t(\theta)$ is the probability ration defined as

$$r_t(\theta) = \frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_k}\left(a_t \mid s_t\right)}. \tag{18}$$

The value function is update by regression on the mean-squared error:

$$\phi_{k+1} = \arg\min_\phi \mathcal{L}_c(\phi), \tag{19}$$

where

$$\mathcal{L}_c(\phi) = \mathbb{E}\left(V_\phi(s_t) - \hat{R}_t\right). \tag{20}$$

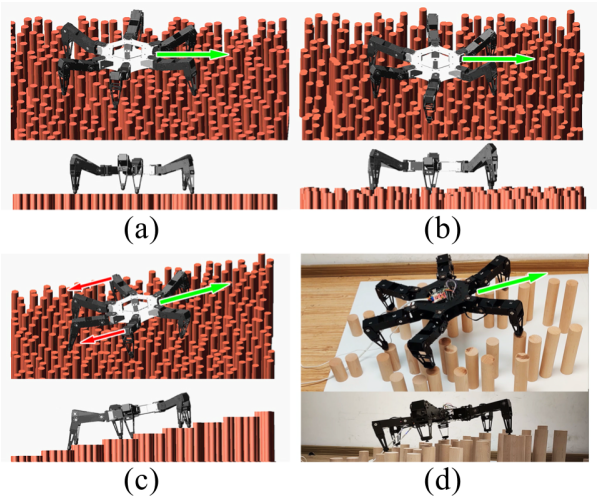

Figure 4: Different types of plum-blossom pile environments. (a) Random plane plum-blossom piles E_1. (b) Random height plum-blossom piles E_2. (c) Random stair plum-blossom piles E_3. (d) Simplified version of E_3 in real world E_4.

4 Experiments

In this section, the multi-contact motion policies of the hexapod robot is trained based on the proposed DRL algorithm, and the policies are tested in both simulation and real plum-blossom pile environments. We assessed the performance of the trained policies through the Average Episode Rewards (AER), the Average Episode Steps (AES) and the Average Success Rate (ASR) in different environment settings. To illustrate the effect of the transition feasibility model, we trained two kinds of policies, one is trained with the proposed transition feasibility model and the other is trained with a normal kinematic model. We use the Average Transferable Rate (ATR) to assess the performance of the later policy.

4.1 Experimental Setup

In order to test our method, we build three different types of simulation plum-blossom pile environments E_1, E_2, E_3 and a real plum-blossom pile environment E_4 as shown in Fig. 4. All the simulation environments present a $1200 \times 1200\ mm^2$ square area consisting of N_p plum-blossom piles, and the E_4 is a simplified version of E_3 in real world. In E_1, we set plum-blossom piles with the same height but random coordinates in the x-axis and the y-axis, and limit the distance between piles to more than $300\ mm$. In E_2, the height of the plum-blossom piles is sampled between $-30\ mm$ and $30\ mm$ randomly on the basis of which in E_1. And the plum-blossom piles in E_3 are set as random stairs. In order to generate E_3, we first set up a standard stair environment. Each step of the stair is $140\ mm$ wide and $25\ mm$ high. Then we project the plum-blossom piles in E_1 onto the surface of the stair in the z-axis direction, and the height of the plum-blossom piles are the same as that of each step of the stair.

4.2 Training Setup

For a training process, we first set a random initial point and a random target area with radius of $100\ mm$. The goal of the hexapod robot is to move successfully from the initial point to the target area with the shortest path. At the beginning of the training process, we first reset the CoM of the hexapod robot to the initial point. Then, the plum-blossom pile nearest to the center of the i_{th} cube $\overline{\mathbf{p}}_i$ is selected as the initial position of the i_{th} contact leg. If the three initial contact legs violate the kinematic constraints, the initial point and the target area are resampled. The initial angle $\boldsymbol{\theta}$ of the body is parallel to the plane formed by the three initial contact legs. So far, the reset of the hexapod robot is completed. Then, for each time-step t, the hexapod robot obtains the current state s_t, executes the action a_t from the policy network, receives a reward r_t and obtains the next state s_{t+1}. When the CoM of the hexapod robot reaches the target area or the maximum number of steps in the current episode reaches 300, the current episode is terminated and a new episode is started. We repeat the above process until the end of the training.

To illustrate the effect of the transition feasibility model, we trained two kinds of policies, one is trained with the proposed transition feasibility model and the other is trained with a normal kinematic model. For the training process of the latter policy, we only check the kinematic constraints of the initial position and the target position at each time-step t, as is done in [Ding *et al.*, 2020]. Moreover, we use a simplified kinematic model in three-dimensional space as described in subsection 3.3.

We train our policy network on a computer with an i7-7700 CPU and a Nvidia GTX 1060ti GPU. The RL algorithm is implemented using Pytorch[1], and the transition feasibility model used in the training process is solved using CasADi[2].

4.3 Experimental Results

According to the learning curve, as shown in Fig. 5, a total of 1 million time-steps are set for training and the whole training process takes about 12 hours in each environment. We tested the trained policies in E1, E2, E3 and E4 and we assumed that the environment information and the position of the robot are known quantities. We assessed the performance of our method through AER, AES and ASR in different environments as shown in Table 2. The fully trained policies can generate valid footholds and CoM sequences which lead the robot to the random target area with a ASR between 90% and 100%. The policies that trained without transition feasibility model can get a even higher ASR, but the ATR is low, which may results in a poor passibility in the real system. Since the algorithm verifies the transition feasibility at each time-step t in the training process, it can be observed in Fig. 6 that the rotation angles of coxae, which are constrained between $-45°$ and $+45°$, always satisfy the kinematic constraints (Take E3 as an example). Another important observation is about the distribution of the legs in different environments. In cases of E1 and E2, the legs are relatively evenly distributed around the CoM. However, in the cases of E3 and E4, leg 2 and leg 5 are placed to the rear for better stability due to the transition feasibility model.

[1]https://pytorch.org/
[2]https://web.casadi.org/

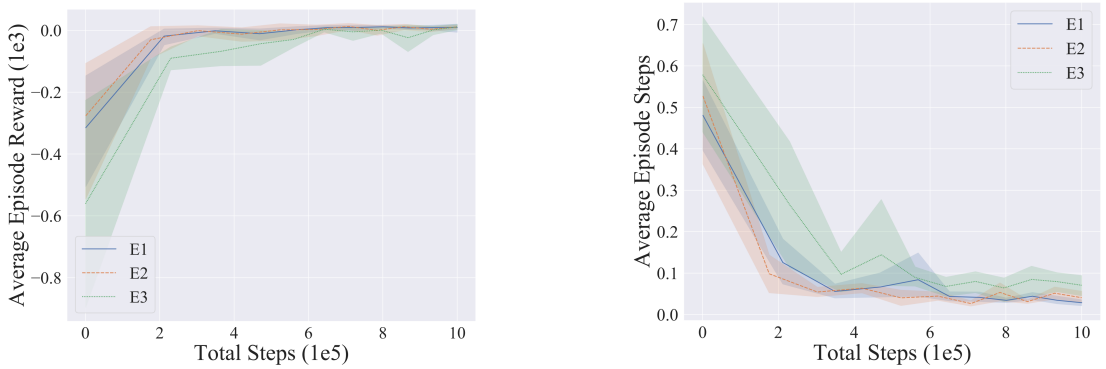

Figure 5: Learning curves for E1, E2 and E3. (a) The average episode reward. (b) Average episode steps which represent the quotient of the steps in a episode and the distance between the initial point and the center of the target area.

N_p	Metric	E1		E2		E3	
		With TFM	Without TFM	With TFM	Without TFM	With TFM	Without TFM
600	AER	2.085	8.285	1.137	7.213	0.466	6.592
	AES	0.035	0.009	0.041	0.052	0.081	0.092
	ASR	100%	100%	100%	100%	99%	100%
	ATR	-	42%	-	36%	-	37%
500	AER	1.723	7.982	1.077	6.514	-0.890	6.558
	AES	0.047	0.010	0.061	0.093	0.148	0.114
	ASR	100%	100%	100%	100%	95%	99%
	ATR	-	35%	-	29%	-	31%
400	AER	0.683	7.234	-0.103	6.125	-2.358	5.918
	AES	0.112	0.031	0.174	0.104	0.270	0.113
	ASR	99%	100%	97%	99%	92%	98%
	ATR	-	28%	-	30%	-	26%

Table 2: The performance of the learned policies in different types of environments with 100 random initial points and 100 random target areas. There are two kinds of policies, one is trained with the Transition Feasibility Model (TFM) and the other is trained without TFM. The metrics include Average Episode Rewards (AER), Average Episode Steps (AES), Average Success Rate (ASR) and the Average Transferable Rate (ATR). N_p is the number of plum-blossom piles in each environment.

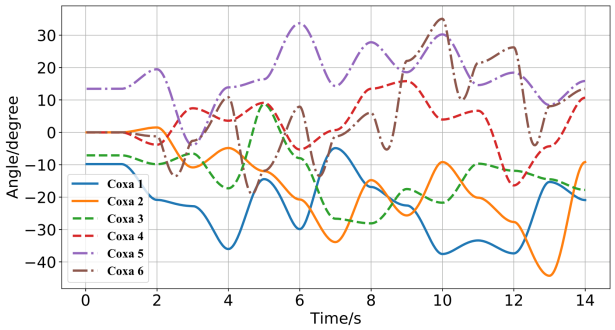

Figure 6: The rotation angles of coxae in E3.

5 Conclusion

In this paper, we presented an integrated DRL method for hexapod robots moving on uneven plum-blossom piles. The multi-contact motion planning problem is mathematically formulated as an MDP with the specified reward function. The promising properties of DRL enable the motion planning algorithm to be implemented in a high dimensional state and action space. The proposed transition feasibility model for hexapod robots ensures that the planned footholds and CoM sequences satisfy the kinematic and dynamic constraints. Both of the simulation and experimental results on physical systems demonstrate the feasibility and efficiency of the proposed method. Our future work will focus on obtaining more robust motion planning policies for dynamic environments and extending the proposed method to more complex real environments.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (No. 2018AAA0101100), the State Key Laboratory of Computer Architecture (ICT, CAS) under Grant No. CARCHB202012 and the National Natural Science Foundation of China (No. 62073160).

References

[Arulkumaran *et al.*, 2017] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[Belter *et al.*, 2016] Dominik Belter, Przemysław Łabźcki, and Piotr Skrzypczyński. Adaptive motion planning for autonomous rough terrain traversal with a walking robot. *Journal of Field Robotics*, 33(3):337–370, 2016.

[Betts, 1998] John T Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.

[Ding *et al.*, 2020] Liang Ding, Peng Xu, Haibo Gao, Zhikai Wang, Ruyi Zhou, Zhaopei Gong, and Guangjun Liu. Fault tolerant free gait and footstep planning for hexapod robot based on monte-carlo tree. *arXiv preprint arXiv:2006.07550*, 2020.

[Fernbach *et al.*, 2018] Pierre Fernbach, Steve Tonneau, and Michel Taïx. Croc: Convex resolution of centroidal dynamics trajectories to provide a feasibility criterion for the multi contact planning problem. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–9. IEEE, 2018.

[Hwangbo *et al.*, 2019] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019.

[Klamt and Behnke, 2019] Tobias Klamt and Sven Behnke. Towards learning abstract representations for locomotion planning in high-dimensional state spaces. In *International Conference on Robotics and Automation*, pages 922–928. IEEE, 2019.

[Lee *et al.*, 2020] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47), 2020.

[Mastalli *et al.*, 2015] Carlos Mastalli, Ioannis Havoutis, Alexander W Winkler, Darwin G Caldwell, and Claudio Semini. On-line and on-board planning and perception for quadrupedal locomotion. In *IEEE International Conference on Technologies for Practical Robot Applications*, pages 1–7. IEEE, 2015.

[Mastalli *et al.*, 2017] Carlos Mastalli, Michele Focchi, Ioannis Havoutis, Andreea Radulescu, Sylvain Calinon, Jonas Buchli, Darwin G Caldwell, and Claudio Semini. Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion. In *IEEE International Conference on Robotics and Automation*, pages 1096–1103. IEEE, 2017.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[Peng *et al.*, 2017] Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics*, 36(4):1–13, 2017.

[Rebula *et al.*, 2007] John R Rebula, Peter D Neuhaus, Brian V Bonnlander, Matthew J Johnson, and Jerry E Pratt. A controller for the littledog quadruped walking on rough terrain. In *IEEE International Conference on Robotics and Automation*, pages 1467–1473. IEEE, 2007.

[Rivlin *et al.*, 2020] Or Rivlin, Tamir Hazan, and Erez Karpas. Generalized planning with deep reinforcement learning. *arXiv preprint arXiv:2005.02305*, 2020.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Shahriari and Khayyat, 2013] Mohammadali Shahriari and Amir A Khayyat. Gait analysis of a six-legged walking robot using fuzzy reward reinforcement learning. In *13th Iranian Conference on Fuzzy Systems*, pages 1–4. IEEE, 2013.

[Silver *et al.*, 2016] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction (2nd Edition)*. MIT press, 2018.

[Tonneau *et al.*, 2018] Steve Tonneau, Pierre Fernbach, Andrea Del Prete, Julien Pettré, and Nicolas Mansard. 2pac: Two-point attractors for center of mass trajectories in multi-contact scenarios. *ACM Transactions on Graphics*, 37(5):1–14, 2018.

[Tsounis *et al.*, 2020] Vassilios Tsounis, Mitja Alge, Joonho Lee, Farbod Farshidian, and Marco Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2):3699–3706, 2020.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Winkler, 2018] Alexander W Winkler. *Optimization-based motion planning for legged robots*. PhD thesis, ETH Zurich, 2018.

[Zucker *et al.*, 2011] Matt Zucker, Nathan Ratliff, Martin Stolle, Joel Chestnutt, J Andrew Bagnell, Christopher G Atkeson, and James Kuffner. Optimization and learning for rough terrain legged locomotion. *The International Journal of Robotics Research*, 30(2):175–191, 2011.