# Bipedal Robot Walking on Complex Surface

Yansong Jia, Xinyu Li

March 5, 2023

## 1 Why

Bipedal robots can be used for emergency rescue, disaster relief, exploration, and other tasks. In the real world, bipedal robots need to move in complex and unstable environments, such as uneven ground, slopes, steps, dust, gravel-covered ground, etc., which may cause the robot to lose balance and fall down. However, traditional control methods have difficulties in dealing with complex nonlinear systems. We mainly try to train the walking control algorithm through deep reinforcement learning, which can make the robot realize stable walking in an unstable environment, so as to play a role in various practical application scenarios.

## 2 Conventional Algorithms

Bipedal robot walking is a very popular problem, people commonly establish the dynamics model of the robot, and then use the traditional modern control technology of LQR or MPC to control the robot walking.

The simplified dynamic model of the bipedal robot is:

$$\ddot{x}_c = \frac{g}{h_0}(x_c - x_a) + \frac{1}{mh_0}(\tau_a - \tau_h) + \frac{F(t)}{m} \tag{1}$$

where $x_c$ is the projection of the joint on the ground, $x_a$ is the foot point, $m$ is the mass of the robot, $h_0$ is the height of the joint, $\tau_a$ and $\tau_h$ are the torques of the robot leg, and $F(t)$ is the force applied on the joint.

LQR solves control problems by minimizing a quadratic cost function, a combination of the current state cost. After linearizing the system, we construct a discrete-time state space model and define the Q and R matrix, which are the gain of the cost function of the state vector and the gain of the cost function of the input respectively. The gain of the whole State Feedback System, K can be calculated by the Q and R matrix calculated previously. Finally, the whole robotic closed-loop system can stabilize the disturbances of the road surface, and make the robot walk stably.

## 3 Problem Statement

In this task, we set up a planar biped robot (with 2 degrees of freedom for each leg) and set up a scene with a ladder, stumps, and traps. The biped robot will maintain its balance and continue to move forward in this scene. If the robot falls within the set time, it will be considered a failure of the current mission. The biped robot model used in this project is shown in Figure 1.

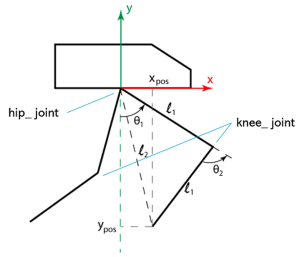


Figure 1: Model of a biped robot

# 4 RL Cast

## 4.1 State Space

The state is a $1 \times 24$ vector, which is shown in Table 1.

Table 1: State Space

| Index | State | Min Value | Max Value |
|---|---|---|---|
| 0 | hull angle | 0 | $2\pi$ |
| 1 | hull angular velocity | -Inf | Inf |
| 2 | velocity x | -1 | 1 |
| 3 | velocity y | -1 | 1 |
| 4 | hip joint 1 angle | -Inf | Inf |
| 5 | hip joint 1 speed | -Inf | Inf |
| 6 | knee joint 1 angle | -Inf | Inf |
| 7 | knee joint 1 speed | -Inf | Inf |
| 8 | leg 1 ground contact flag | 0 | 1 |
| 9 | hip joint 2 angle | -Inf | Inf |
| 10 | hip joint 2 speed | -Inf | Inf |
| 11 | knee joint 2 angle | -Inf | Inf |
| 12 | knee joint 2 speed | -Inf | Inf |
| 13 | leg 2 ground contact flag | 0 | 1 |
| 14-23 | lidar readings | -Inf | Inf |

## 4.2 Action Space

The action is a $1 \times 4$ vector, containing every action $\frac{Torque}{Velocity}$, which is shown in Table 2.

Table 2: Action Space

| Index | Action |
|---|---|
| 0 | hip1 |
| 1 | knee1 |
| 2 | hip2 |
| 3 | knee2 |

## 4.3 Reward Structure

Each fall is awarded -100 points, a small number of negative points for driving the joint to rotate, and positive points for moving forward. A total of 300 points is enough to win.

## 4.4 Nerual Network Structure

We used TD3 and SAC separately for training and compared the results of the two training methods. The gait adapted to different environments was generated by PMTG, and the robot was trained to walk under this gait by TD3 and SAC, so as to realize the walking control of biped robots in unstable environments.

# 5 RL algorithms and envisioned results

We will use Twin Delayed DDPG (TD3) and Soft Actor Critic (SAC) reinforcement learning algorithms to make the bipedal robot walk stably. The TD3 algorithm improves the performance of the Deep Deterministic Policy Gradient (DDPG) algorithm, and SAC optimizes the off-line policy.

Compared with the traditional method like LQR control, by reinforcement learning method, the upright pole will be more stable with a simpler mathematical model.

These tests will be performed in the environment of OpenAI gym, and we will discuss the results of different methods.