EVALUATION OF HOPPING ROBOT LOCOMOTION FOR PLANETARY EXPLORATION IN A 3D SIMULATOR

Virtual Conference 19–23 October 2020

Kosuke Sakamoto[1], Auke Jan Ijspeert[2], Takashi Kubota[3]

[1]The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo, Japan, E-mail: k_sakamoto@ac.jaxa.jp
[2]Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 14, CH-1015 Lausanne, Switzerland, E-mail: auke.ijspeert@epfl.ch
[3]ISAS/JAXA, Sagamihara, Kanagawa 252- 5210, Japan, E-mail: kubota.takashi@jaxa.jp

ABSTRACT

Hopping robots, called hoppers, are expected to act on rough terrain, such as disaster areas, planetary environments, or the both. The uncertainties of the hopping locomotion in such environments are big, and hence we need path planing algorithms to traverse in uncertainty environments. This study investigated hopping as a locomotion strategy for robotic planetary rovers. First, we designed the hopping and wheeled robot, then confirmed the performance in a 3D simulator, Pybullet. We made a planetary-like rough terrain in the simulator, and confirmed that the robot could roll and hop like the real device. In order to improve the locomotion strategy, we tested the use of reinforcement learning. Control policies were learned that allow the robot to move and to reach the goal, but more simulations are needed to better validate the approach in various conditions.

1 INTRODUCTION

How to expand planetary exploration area? Planetary surface exploration has been conducted using wheeled vehicle robots, called rover. Lunokhod 1 and 2, and Chang'E-3 and -4 had been developed to explore the Moon and succeeded in their missions. The Martian surfaces had been explored by Sojourner, Spirit and Opportunity, and has been being explored by Curiosity. Recently, various environments are expected to research by robots. However, such environments are often hard to traverse using wheeled rovers. Hopping rovers, called 'hopper,' are one of the solutions to perform on challenging terrains. Hoppers are expected to act in disaster areas[1], or celestial bodies[2], [3], [4], [5]. In the September 2019, the MINERVA-II, the hopper developed by JAXA/ISAS, succeeded the landing on the surface of the asteroid "Ryugu", the locomotion, and taking the photos (Fig. 1) [6]. This achievement indicates that the planetary surface explorations by hoppers become active more and more. For example, various environments, such as a Recurring Slope Lineae (RSL) on Mars [7], are expected to research by the hoppers. However, there are many

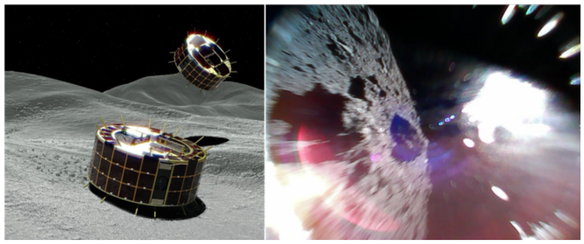

Figure 1: MINERVA II and the contribution. Left: the image of MINERVA II on asteroids. Right: the real photo what MINERVA II took while hopping on asteroid Ryugu. Image credit by JAXA[6].

challenges to carry out actual planetary surface explorations by a hopper/hoppers. One of the challenges is the path or motion planning problems. We cannot know the details of the conditions of planetary surfaces before robots arrive at and explore a celestial body. The environments have uncertainties of locomotion. In addition, planetary surfaces are almost covered with granular media, called regolith. The sandy terrains might cause the stuck. Therefore, we need hopping path/motion planning algorithms in order to investigate such terrains, or environments by the hopper.

The contribution of this paper is to validate the hopper performance in planetary-like environments using a 3D simulator. In order to develop the hopper, the validation of the performance using 3D simulator is important because it is difficult to test in actual planetary environments. First, the design of the hopper is shown. Next, the performance of the hopper is tested in various terrains. Finally, we describe the hopping locomotion generated by reinforcement learning.

2 THE ROVER DESIGN

This work employs a two-wheeled robot equipped with a hopping mechanism as shown in Fig.2. This design allows different modes of locomotion and improves the traversability of the robot on rough terrains. The robot uses the wheels on relatively flat terrain and

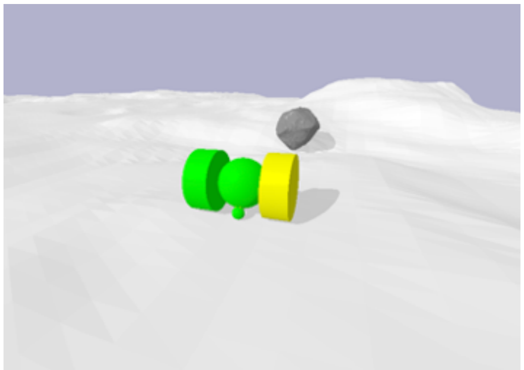

Figure 2: The image of the hopping robot and the planetary like terrain

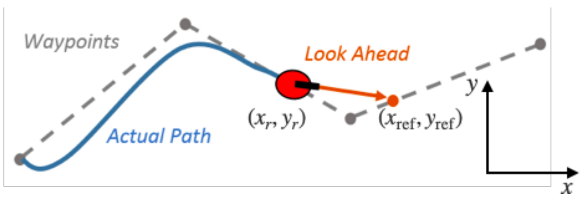

Figure 3: The image of pure pursuit algorithm[10]

gentle slopes, and to change directions. Hopping locomotion is used to clear an obstacle, step, cliff, or to escape from a stuck position.

3 SIMULATION STUDY

This section describes the details of the simulation of this study. As a 3D simulator, this study employs "Pybullet [8]" in order to validate the performance of the robot in planetary environments. First, the conventional wheel controller is validated. Next, the hopping locomotion in planetary environments is

3.1 Performance of Wheel Locomotion

One of the important subjects of the robot control is to follow a path accurately, which is generated by a path planning algorithm. Pure pursuit [9] is one of the path following algorithms. Figure 3 shows an image of pure pursuit. The algorithm is turning controller to reach a target point on the reference path. The calculation cost of the pure pursuit is light enough, hence the algorithm is suit for the planetary rover. The distance between the robot and a target point is constant, called "Look ahead distance (L)". The angle error α between the robot and a target point is described as below:

$$\alpha = \arctan\left(\frac{y_{ref} - y_{\mathrm{r}}}{x_{\mathrm{ref}} - x_r}\right) - \theta \tag{1}$$

where $(x_r, y_r), (x_{\mathrm{ref}}, y_{ref})$ *and* θ denote the position of the robot, the target point, and the look ahead direction of the robot. The turning velocity ω is calculated as follows:

$$\omega = \frac{2v_r sin\alpha}{L} \tag{2}$$

where v_r denotes the velocity of the robot. Figure 4 shows the simulation of a wheel locomotion by pure pursuit. The robot perfume on flat plane (left figure) and rough terrain (right figure). The rough terrain is modeled as a planetary environment.

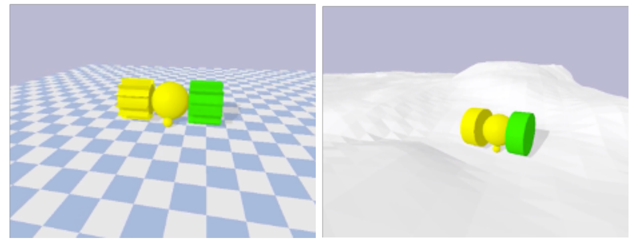

Figure 4: The simulation of pure pursuit; (Left) flat terrain; (RIght) rough terrain

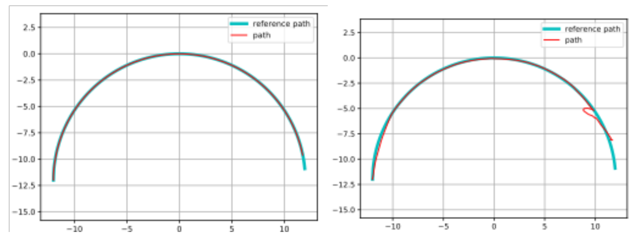

Figure 5: The results of path following(circle); (Left) flat terrain; (RIght) rough terrain

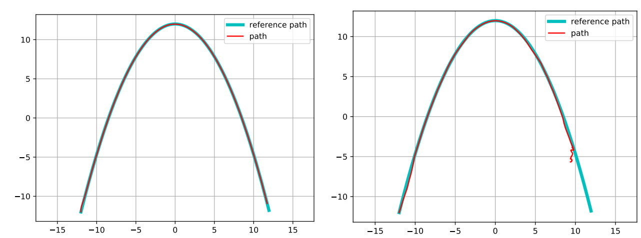

Figure 6: The results of path following(parabola); (Left) flat terrain; (RIght) rough terrain

The results are shown in Fig.5 and Fig.6. The reference paths are circle (Fig. 5) and parabola (Fig. 6), expressed as blue line. The actual paths are expressed as red line. The results indicate that the pure pursuit algorithm performs well on flat plane. On the other hand, the robot does not follow the reference path with accuracy. This is because the roughness of terrain, or slopes. These results show that planetary rovers need other locomotion systems to traverse on rough terrain.

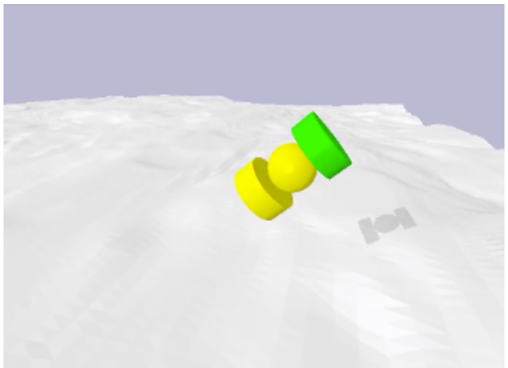

Figure 7: The simulation of hopping locomotion

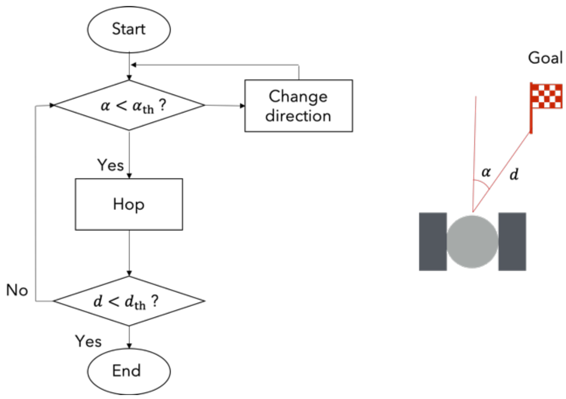

Figure 8: The algorithm of hopping locomotion

3.2 Performance of Hopping Locomotion

This section presents the performance of hopping locomotion. We confirm that only wheeled locomotion does not perform well on rough terrain. The simulation of hopping is shown in Fig. 7. The algorithm of hopping locomotion is shown in Fig. 8. If the angle error α between the robot direction and the goal is more than the constant value α_{th} , the robot rotates on the spot. If the α is less than the α_{th} , the robot hop. After hopping, the distance d between the robot and the goal is larger than the constant distance d_{th} , the robot continues hopping. If not, the robot finishes the locomotion. Figure 9 shows the hopping locomotion simulation on the moon, the mars, and the earth. The results indicate that the larger hopping distance is, the smaller the gravitational level is. The robot needs three times hopping on the moon, and four times on the earth. However, the robot cannot reach the goal. The robot stuck on the sunken place and could not escape form the place. This is why the robot have to choose a hopping pattern depends on the terrain condition. In addition, the combination of hopping and wheeled locomotion can traverse more flexibly on various terrains.

4 REINFORCEMENT LEARNING FOR HOPPING

In order to improve hopping performance, we use the reinforcement learning. Reinforcement learning is one of the most active field of machine learning. Robotics is one of the applications of reinforcement learning. The advantages of the reinforcement learning for robotics are that it does not depend on the environment, and it can be applied in case of changing the environment where a robot act [11].

4.1 The Design of Parameters

We use policy gradient which is one of the methods of reinforcement learning. The method learns the action policy which maximize the value function. This simulation uses the discount factor $\gamma = 0.9$, and the learning rate $\alpha_{learn} = 0.01$. The reward function r is designed as below:

$$r = 1/d \tag{3}$$

If the robot can reach the goal, the robot gets the reward +500. If the distance between the robot and the goal become larger than a constant distance, the robot gets the penalty -500. In addition, if the robot cannot reach the goal in 40 seconds, the robot gets the penalty -400, and the simulation is stopped.

4.2 The Results of the Simulation

Figure 10 shows the locomotion generated by reinforcement learning. The rewards history is shown in Fig. 11. The total number of learning is 500. We observe that the reward function converges to -125 around 300 times of iterations. The reason of why the reward function converges to negative value is that the robot did not reach the goal in 40 seconds while learning. After finished learning, we apply the weight of neural network to the hopping simulation, and we can confirm that the robot can reach the goal. However, we also confirm the unrealistic locomotion, such as driving on single wheel on slope. For the future works, we will modify the design of reward function, add the penalty or limitation to the each actions.

5 CONCLUSION

This paper presents the hopping and wheeled locomotion performance in planetary environments using 3D simulator. Only wheels or hopping locomotion does not perform well in planetary environments. The reinforcement learning are applied to the combination locomotion, and the robot could reach the goal. As future works, we need modified the action list or reward function in order to create realistic motions.

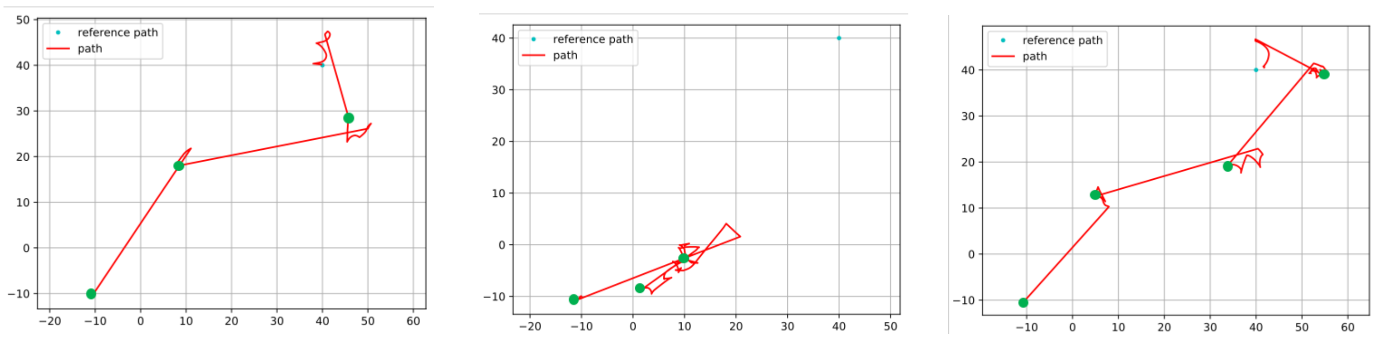

Figure 9: The results of hopping trajectory. The green points show the hopping spot. The red lines show the trajectory. Left: The trajectory in the moon environment(1/6G); Center: The trajectory in the mars environment(1/3G); Right: The trajectory in the earth environment(1G).

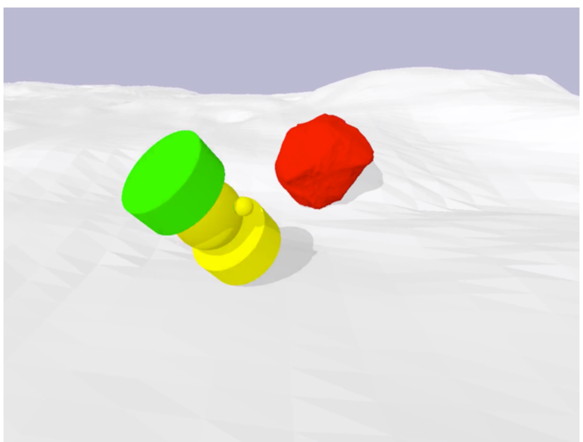

Figure 10: The simulation of locomotion generated by reinforcement learning

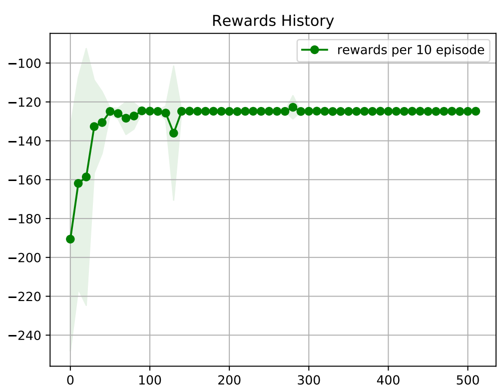

Figure 11: The history of the rewards while learning

References

[1] H. Tsukagoshi, M. Sasaki, A. Kitagawa, and T. Tanaka (2005) Design of a higher jumping rescue robot with the optimized pneumatic drive. In: *proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1276–1283.

[2] R. G. Reid, L. Roveda, I. A. Nesnas, and M. Pavone (2014) Contact dynamics of internally-actuated platforms for the exploration of small solar system bodies. In: *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Saint-Hubert, Canada, p. 9.

[3] D. Mège, J. Gurgurewicz, J. Grygorczuk, Ł. Wiśniewski, and G. Thornell (2016) The highland terrain hopper (hopter): Concept and use cases of a new locomotion system for the exploration of low gravity solar system bodies. *Acta Astronautica*, vol. 121, pp. 200–220.

[4] S. Montminy, E. Dupuis, and H. Champliaud (2008) Mechanical design of a hopper robot for planetary exploration using sma as a unique source of power. *Acta Astronautica*, vol. 62, no. 6-7, pp. 438–452.

[5] T. Yoshimitsu (2004) Development of autonomous rover for asteroid surface exploration. In: *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 2529–2534.

[6] JAXA. Minerva-ii1: Images from the surface of ryugu. Available at: http://www.hayabusa2.jaxa.jp/en/topics/20180927e_MNRV/.

[7] J. Levy (2012) Hydrological characteristics of recurrent slope lineae on mars: Evidence for liquid flow through regolith and comparisons with antarctic terrestrial analogs. *Icarus*, vol. 219, no. 1, pp. 1–4.

[8] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. Available at: http://pybullet.org

[9] R. C. Coulter (1992) *Implementation of the pure pursuit path tracking algorithm*. tech. rep., Carnegie-Mellon UNIV Pittsburgh PA Robotics INST.

[10] Pure pursuit controller. Available at: https://jp.mathworks.com/help/robotics/ug/pure-pursuit-controller.html

[11] R. S. Sutton and A. G. Barto (2018) *Reinforcement learning: An introduction*. MIT press.