Motion Control for Legged Robots with Deep Reinforcement Learning

ジョーンズ ウィリアム ワトキン
氏　名　JONES WILLIAM WATKIN
研究科，専攻の名称　東北大学大学院工学研究科（博士課程）航空宇宙工学専攻
学 位 論 文 題 目　Motion Control for Legged Robots with Deep Reinforcement Learning
論 文 審 査 委 員　主査　東北大学教授　吉田　和哉　東北大学教授　林部　充宏
東北大学教授　岡谷　貴之

論文内容要約

The exploration of space provides unparalleled insights into the wonders of the Universe and humanities place within it. Recent advances in launch vehicle technologies are reducing costs required to move mass into orbit, thus enabling the realisation of a variety of sociological and technological concepts. If humanity is to become an interplanetary species, we first must understand the local environment of Earth, its resource, and habitability.

The Moon, with its close proximity to Earth and resources thought to exist at its poles, is an ideal place for humans to learn how to live and adapt in outer space. Potential networks of underground tunnels could provide naturally made shelter from damaging environmental conditions present on the surface, as well as relatively untouched lunar material for scientific analysis. Mars, with its orbital characteristics and global environment not dissimilar to Earth could be another planet in the Solar System within which life has evolved. Exploration of areas less affected by the harsh atmospheric conditions, such as sub surface lava tunnels, would again provide good scientific insight and potential habitat for future settlements. In addition, the exponential increase in population on Earth is resulting in the accelerating depletion of natural resources used to develop and build new technologies. The prospect of mining of asteroids, if the initial costs can be overcome, is one that would solve this problem as the amount of resources they hold is more than enough to supply global markets. In-situ exploration of different classes of satellites also provides more information to help answer questions relating to the formation of the Solar System.

The question that then arises is the means by which we explore such environments. Most of these places are far too risky for humans to explore in person, or are simply inaccessible with current technology. Therefore robotic exploration is the best choice. Following then from this is the question of the approach to design of the robotic architectures that are to explore areas of interest on the Moon, Mars, and asteroids. On Earth, billions of years of evolution has resulted in a wealth of biological form that is highly adapted to fully exploit its local habitat so that it has a good chance of survival. When considering animals that live in habitats similar to those we wish to explore in outer space, there are different evolutionary aspects that have emerged to suit specific environments. Animals that live in habitat characterised by rocky, steep terrain are almost always legged, but have different anatomical structure depending on the specifics of their environment and the means by which they live. Some have evolved to be fast and efficient in order to catch prey effectively, whereas some have evolved specialised hands and feet that enable them to stay secure in precarious locations. Inspired by these animals highly tailored to suit their environment, we can use the biomimetic approach in design to develop robotic

hardware capable of moving in extreme terrain. Once the hardware is made, the issue of its control is one that can also take a biomimetic approach.

When looking at the way animals control themselves in the natural world, there is an overall structure in the means by which they use experience to base future decisions upon. Simply put, when animals make decisions that result in conditions not in their favour, the notion of a negative result is understood by the animal, and it remembers not to make similar decisions in future. Likewise, if a decision it makes turns out to be good, it may want to repeat it. Deep reinforcement learning is a machine learning technique that uses interaction with an environment to gain an understanding of it through the experience of reward due to change of state as a result of actions. Through repeated interaction with an environment, an agent is able to learn which actions result in states that provide different levels of reward. By designing algorithmic architectures that maximise expected future reward for interaction with an environment, the agent can learn policies that result in good performance with respect to the definition of reward. Deep reinforcement learning has recently shown strong evidence it is the means with which the brain uses dopamine reward signals to learn. It therefore follows that the definition of reward is critical in the final policy that is learned by the agent. If the agents task is to maximise reward as defined by the reward function, small differences in the composition of the reward function can have drastic effects on the optimal policy learned by the agent.

In this work, deep reinforcement learning is applied to the problem of motion control for three different types of legged robot in a variety of terrain. Investigation into the critical nature of the reward function is discussed when training a legged robot to walk on a flat plane. It is shown that differences in defining reward that encourages a legged robot to move forward result in drastically different learned policies. The result of this analysis is then used to investigate the difference in learned policies in lunar gravity when changing the cost due to torque use. Three different values of cost were trained with and an upper limit was found, beyond which the agent was unable to learn sustained forward motion due to too strong a penalty for torque use. The policy learned with the largest amount of cost for torque displayed highly dynamic, fluid motion, with repeated full flight phases throughout the duration of the episode, in line with analysis of gait cycle with respect to the Froude number. In contrast, policies learned with weaker torque cost displayed more unnatural motion, with high frequency, low amplitude oscillations present in the legs, however they took significantly less time to converge compared to the policy with strongest torque cost. This result showed that for the same legged robot, a reward function used to learn motion for Earth applications can be used to learn different motion in the same environment under lunar gravity, through modification to coefficients of cost. This shows that knowledge of environment dynamics combined with a systematic approach to reward function design is highly useful when developing new RL frameworks in different environments.

A comparable approach is used to train another legged robot to move up and down a variety of inclined planes without providing any direct knowledge of location or profile of the planes. The only information passed to the agent is that of the robot state. In effect, the robot was blind. Even though the robot was unable to see the location or inclination of the planes, the agent was able to learn policies that successfully moved the robot onto the inclines for 5 out of 6 of the environments, where the failure case was likely due to no policy actually existing for the legged robot model used. The resultant policies showed that the distributed algorithm used was able to train the agent to memorise the location of the intersection of the planes and understand their inclination via indirect haptic feedback from the change in state of the robot upon encounter. The motion generated for moving up

inclines was highly dynamic, and symmetrical gait patterns emerged, occasionally with full flight phases, which were able to keep enough control to stay within the episode termination conditions defined in the training framework. The learned motion for moving down inclination was far less dynamic. The robot learned to exploit the extra gravitational potential energy available in the problem, so the joints were far less active than they were for moving up inclines. In addition, the agent learned to drop the base of the robot towards the plane, in doing so increasing stability by lowering the CoM. In some cases, the robot even dragged its legs on the plane to control its motion with friction, rather than use opposing torques in the leg joints. This may have arisen from cost due to torque use in the reward function, or simply learned as a control technique to again lower CoM. This chapter demonstrated the benefit of large rate of data collection that distributed algorithms allow, in doing so learning a variety of complex control problems for different terrain when the environmental data available to the agent for learning is restricted.

Finally, reinforcement learning is used to train a legged climbing robot with grippers to free climb in a variety of terrain at different inclinations using an abstraction of the environment. As the mechanical performance of the gripper defines the points within the environment that the robot can make secure contact with, the environment can be simplified such that the robot is only able to place its grippers on such points. The reinforcement learning approach is then used to train an agent to select a series of grippable points on a 2D plane that allow the robot to move to a goal position whilst satisfying various constraints. In addition, a quantitative measure of stability is included as a component of reward that is able to change the overall stability of final policy dependant on its strength. Training on maps with reducing grippable point density then provides a quantitative measure of the probability of success for various densities. It is found that lower densities have less probability of success, however the overall stability of the robot stays relatively constant, indicating that it is the specific spatial relationship between a set of grippable points that defines valid stances. Lastly, the same approach is extended to the case of free climbing in simulated 3D terrain based on the geometrical configuration of the real robots gripper, where the only modification to the learning framework required is to include the 3D location of the grippable points in the state array.