Continuous Action Solutions in the *Lunar Lander* Problem

Samuel A. Roberts, Spyridon Samothrakis, Simon M. Lucas

***Abstract*—**

I. Introduction

This paper describes an investigation into solving a continuous problem based on a familiar arcade game, *Lunar Lander*.

II. Motivation

A. Solving Continuous Problems with Continuous Actions

Previous research into using MCTS to provide solutions for continuous physics-based problems have examined cases such as the Physical Travelling Salesman Problem [1]. Within these continuous contexts, discrete actions have proven to be quite effective, especially with use of macro-actions and some degree of higher level planning [2].

Lunar Lander is an excellent test case to study for many reasons, due to its inherently multi-objective nature. With the objectives to both land quickly while also minimising velocity changes, which cost fuel which must be conserved, it has been an interesting scenario to examine the properties and constraints imposed just by the environment alone. While there have been attempts to solve this deceptively complex problem before using discrete macro-actions [3], in this paper the problem is being approached as one that can be better solved with continuous actions.

III. Methodology

A. Software Implementation

B. Environment and Physics

1) Environmental Properties: The properties of the base environment of *Lunar Lander* are that it is frictionless, and that it is a two-dimensional plane with horizontal wrapping. This can also be conceptualised as a cylinder. Anything that passes from the left edge of the playing field moves to the right instantaneously, and vice versa.

The other feature of note is the jagged landscape, a simplistic representation of the noisy, crater-filled lunar landscape the game represents. This landscape is constructed as a series of line segments, with each vertex of the jagged landscape being distributed equally horizontally, and randomly vertically.

Each vertex of this landscape is generated sequentially.

2) Spaceship Physics: The spaceship within this game world is modelled as an a circular mass with some basic physical properties, such as a position within the game world $\boldsymbol{s}$, a velocity $\boldsymbol{v}$, an orientation θ, an angular velocity ω and a radius of a bounding sphere r used for collision detection with the landscape.

This collision detection method is based on the lowest point of the spaceship's bounding circle coming into intersection with the highest point of the landscape for the line perpendicular between the ship and the bottom of the playing field, as can be seen in Figure 1.

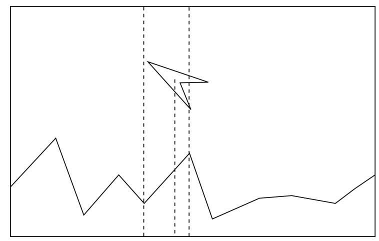

Fig. 1: The point closest to the spaceship on the landscape lies in between two of the defined vertices of the landscape, and requires interpolation to calculate the y co-ordinate from the x co-ordinate of the ship.

As the landscape is stored as a series of line segments, interpolation must be used in the event the ship is not perfectly aligned with one of the landscape axes. The point that lies on the line segment that the ship is being checked against is calculated through simple linear interpolation based on which two vertices the ship is horizontally closest to. For the ship's centre $\boldsymbol{s}$, the left nearest landscape vertex $\boldsymbol{p}^l$ and the right nearest landscape vertex $\boldsymbol{p}^r$, the point of collision $\boldsymbol{p}^c$ against the landscape is calculated as

$$\boldsymbol{p}^c_x = \boldsymbol{s}_x \boldsymbol{p}^c_y = \boldsymbol{p}^l_y + v(\boldsymbol{p}^r_y - \boldsymbol{p}^l_y) \tag{1}$$

where v is a value between 0 to 1 used for interpolation, and can be calculated as follows.

$$v = \frac{\boldsymbol{s}_x - \boldsymbol{p}^l_x}{\boldsymbol{p}^r_x - \boldsymbol{p}^l_x} \tag{2}$$

Collision is then true if the following statement is true:

$$\boldsymbol{s}_y + r \geq \boldsymbol{p}^c_y \tag{3}$$

The ship colliding with the landscape constitutes the end of the *Lunar Lander* game. The conditions surrounding this collision, including speed, orientation of the ship, and fuel used, constitute whether the nature of the collision is a success or a failure.

C. Continuous Action MCTS

1) Heuristic:

IV. Results

V. Conclusion

Acknowledgements

Samuel Roberts is supported by an EPSRC PhD Studentship.

References

[1] D. Perez, E. J. Powley, D. Whitehouse, P. Rohlfshagen, S. Samothrakis, P. I. Cowling, and S. M. Lucas, "Solving the physical travelling salesman problem: Tree search and macro-actions," *IEEE Transactions on Computational Intelligence and Games*, 2014.

[2] E. J. Powley, D. Whitehouse, and P. I. Cowling, "Monte carlo tree search with macro-actions and heuristic route planning for the physical travelling salesman problem," *IEEE Transactions on Computational Intelligence and Games*, 2012.

[3] S. A. Roberts and S. M. Lucas, "Measuring interestingness of continuous game problems," *IEEE Transactions on Computational Intelligence and Games*, 2013.