PROVING LUNAR AUTONOMOUS MICRO-ROVING MISSIONS WILL SUCCEED

Varsha Kumar (varshak@andrew.cmu.edu)

1. Abstract

Micro-rovers are light, inexpensive, and must complete their missions within stringent time constraints. To date, a micro-rover mission to the lunar pole has not occurred. To be successful, such a mission must complete its exploratory goals within 14 Earth days. This paper explains the first formal proof of why a micro-rover mission will succeed under the 14-day time constraint. Contributions include a bound on the time taken to avoid an obstacle as 8/V for the particular model and 18/V in general, where V is the wheel velocity of a rover.

2. Introduction

2.1 Micro-Rovers

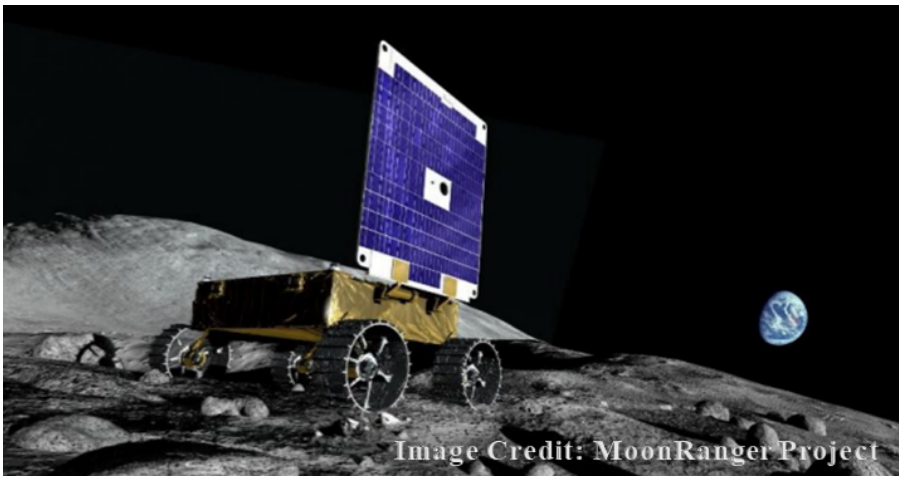

Figure 2.1: A lunar micro-rover. MoonRanger is an autonomous micro-rover under development by Carnegie Mellon University. Other micro-rovers include PitRanger, also under development by Carnegie Mellon, and Pragyan, a pioneering micro-rover developed by the Indian Space Research Organiation (ISRO) [6][7].

Micro-rovers, rovers approximately 30 kg in mass, provide advantageous cost savings for planetary exploration. Because of their low mass, they are inexpensive to launch and relatively inexpensive to fabricate [1].

However, their low mass precludes carrying equipment that would enable them to survive for long durations in the harsh conditions of space. Isotope heating that would enable a lunar micro-rover to last during the extreme cold of the absolutely black lunar night is too heavy to carry and time-consuming to train personnel to work with. As such, lunar micro-roving missions

must take place during the 14-Earth-day Lunar day. In contrast, traditional rover missions have been multi-month or multi-year in duration [8].

Micro-rovers are also too small to carry a direct-to-earth radio. As such, any communication or data product relay must occur via their landers. Therefore, communication between the rover and Earth can only occur within the limited lander communication range. Exploration beyond such limited communication range compels rover autonomy.

To date, however, there has been no formal proof using KeyMeraX that a micro-rover autonomous mission could occur within the extreme time constraints of a lunar day.

2.2 Autonomous Mission

Because of the constraint that communication with Earth occurs only in lander communication, a micro-rover mission may consist of individual forays outside of communication and returning back to lander communication to relay data products. Given that multiple forays are desired, and time must be spent in between each foray to downlink data and receive subsequent foray targets, an individual foray might be limited to take between 24-36 hours roundtrip [9]. An individual foray to a target must both reach a goal and avoid any obstacles along the way.

2.3 Relevant Rover Parameters

For the purposes of this paper, we set the parameters relevant to a micro-rover mission as follows.

Parameter	**Value**
Rover Velocity	5 cm/second
Foray Distance (One-Way)	500 meters
Foray Time	36 hours
Rover Stopping Distance	0 m
Rover Acceleration Distance	0 m
Robot View	5 m

Table 2.1: Parameters set for the purposes of this paper.

As a note, the parameters above suggest that a 500 meter trek should be completed within 18 hours. Rover stopping distance and rover acceleration distance are both set to zero assuming that the rover is small enough and motor torque is high enough to make acceleration and braking take negligible time and distance.

2.4. Related Work

While mission planning for lunar polar missions is thorough [9], we are not aware of any formal proof of mission success. Formal proofs that use KeymeraX typically focus on Earth-based systems. These include airplanes, cars [10], and trains [11]. Cars and trains are perhaps the most directly related to micro-rovers, and controlling trains on a track inspired the arcing portion of this model. However, these systems require more detailed analysis than what is presented in this paper because of their safety-critical nature. In contrast, this paper models a mission scenario and the work has the liberty of assuming additional knowledge about a scenario.

3. Approach

3.1 Summary of Approach

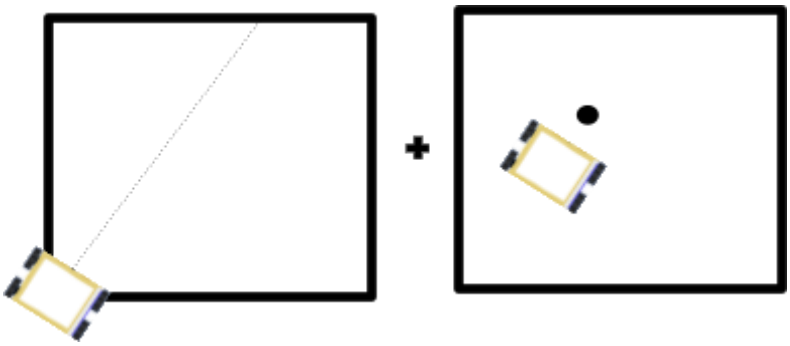

Figure 4.1: Graphical Summary of Approach

The approach first proves that the time to complete a foray with no obstacles is less than the 18 hour limit. Then, to analyze obstacles, a single window is considered. A window consists of the area that the robot is able to see, in our case, 5 m. The assumption is that a five-meter window must be traversed in 648 seconds, the appropriate fraction of time relative to the entire foray duration and distance.

For the purposes of this paper, we consider that there is one obstacle on the desired path within a single window. We then prove that, under certain assumptions, the additional time required to avoid the obstacle is less than the time left over if one were to traverse the window via a straight-line path. This paper does not consider compound obstacles,though such consideration could be an extension of the work.

3.2 Justification of Model Simplifications

The major model simplifications include: analysis in only one direction, considering the path to be along the x-axis, specifying a simple controller for obstacle avoidance, and object shapes.

It is acceptable to analyze in only one direction because equivalent analysis can be conducted for returning to lander communication by setting the lander communication as the goal. Analyzing in one direction does mean that instead of 36 hours, we use 18 hours as the time for a foray.

It is appropriate to consider that the path is along the x-axis because one could always perform a point turn on the rover to align the rover direction with the x axis and reframe the goal coordinates. This point turn does add time to traversing the window, but the additional time is acceptable, as mentioned in the conclusion.

Specifying a controller for obstacle avoidance is also acceptable because we are trying to show that a successful lunar micro-roving mission is possible, not that every controller makes such a mission possible.

Obstacles are shaped as circles for simplicity and because in two dimensions, a crater or a small rock would be roughly circular. The robot edges are calculated using a radius is well, which is an overestimate when considering the circumscribing circle around a rectangular rover.

4. Model Implementation

4.1 Summary of Model Implementation

The implementation consists of two models: one for driving in a straight line for the entire trek, and one for obstacle avoidance in a window

4.2 Relevant Assumptions

Many preconditions are set before the model begins. A summary of important preconditions and justifications for the preconditions is provided in Appendix A.

4.3 Straight-Line Driving Model

The straight-line driving model proves that the rover can complete a foray within a specified time. The main purpose of this model was to discover how to bind time for further analysis.

```
[
  {
      /* Drive along y-axis if far from/past obstacle*/
      {
          {?(abs(x-rogx)>rogr + r + v*T & y=gy);};
          {{?(overdist(x,y,gx,gy) <= gb); v:=0;}++{?(overdist(x,y,gx,gy) > gb); v:=V;}};
          {t:=0;};
          {x'=v, y'=0, globalTime'=1, t'=1 & t<T & globalTime<=timeBound}
      }

  }*@invariant(((overdist(x,y,gx,gy) <= gb & globalTime<=timeBound) | globalTime<timeBound) &
    overdist(x,y,gx,gy) <= ((timeBound-globalTime)*V)+ gb & y=gy & gx >= x &
    dx^2+dy^2=1 & dist(x,y,rogx,rogy) > r+rogr & dx=1) /* loop invariant */
]
(((overdist(x,y,gx,gy) <= gb & globalTime<=timeBound) | globalTime<timeBound)
& dx^2+dy^2=1 & dist(x,y,rogx,rogy) >r+rogr ) /* Goal & Safety condition */
```

Figure 4.1: Straight Line Driving Model

The main components of the is model are to first check whether the rover has reached the goal and should break or continue driving and a simple linear rover dynamics. The main contribution of the straight-line-diving model was the presence of globalTime and a timeBound, creating a recognition that there must be an invariant relationship between velocity, time, and a distance to capture that time is limited to a certain number. In this case, the relationship was

4.4 Obstacle Avoidance Models

The obstacle avoidance strategy is that, when the rover is close enough to the obstacle, it makes a point turn to orient perpendicular to the x-axis. The rover then travels in an arc around the obstacle back to the x-axis. Next, the rover performs another point turn to orient toward the right once again. These three steps complete the obstacle avoidance maneuver.

It is sufficient to model and time-bound a single point turn and the traversal of the arc. As such, there are two models for obstacle avoidance.

The obstacle avoidance models contain two distance metrics, discussed below.

```
/* Overestimate of distance. Used to check if rover is near enough to the goal */
Real overdist(Real ax, Real ay, Real bx, Real by) = max(abs(ax-bx),abs(ay-by));

/* Underestimate of distance. Used to check if rover is far enough from obstacle */
Real dist(Real rx, Real ry, Real ox, Real oy) = min(abs(rx-ox),abs(ry-oy));
```

Figure 4.2 Distance metrics: Overdist is an overestimate of the distance between two points. It is the distance used to see whether the rover is close enough to the goal. An overestimate of distance to the goal is preferable to an under-estimate in this case. Dist is an underestimate of the distance between two points. It is used to estimate the distance between a rover and an obstacle, in which case an underestimate is safer.

4.4.1 Part 1: Point Turn

The point turn model begins with a check that the rover is too close to the obstacle. Then, rver orientation (dx and dy) are varied appropriately to turn the rover to point upwards. The key component of the point turn model is the postcondition that time t<=2*(1-dx)/V. That is to say, the time of a quarter turn is bounded by the time it takes to move upwards and left, each by 1 unit, at velocity V.

```
[
  /* Turn robot Perpendicular to Y-axis */
  /* EVENT-TRIGGERED BECAUSE KNOW WHERE OBSTACLE IS A PRIORI AND
     WE ARE CONCERNED ABOUT TIME TAKEN, NOT LOW_LEVEL ROBOT CONTROL */
  {?(!(abs(x-rogx) >rogr + r  + v*T & y=gy));};
  {v:=V;};
  {t:=0;};
  {dx'=v*dy, dy'=-v*dx, globalTime'=1, t'=1 & dx>=0}
]
(dx^2+dy^2=1 & dist(x,y,rogx,rogy) > r+rogr & t<=2*(1-dx)/V & dx>=0 & (dx=0 ->  t<=2/V))
```

Figure 4.3: Point Turn Model

4.4.2 Part 2: Arc Around Obstacle

The model for arcing around an obstacle must both maintain obstacle avoidance and track time. Maintaining obstacle avoidance is done by recognizing that, because the arc is of radius larger than the obstacle, the rover is always far enough away from the obstacle. Tracking time is done by relating the change to dy to time by suggesting that dy changes at most by 2 in a half-turn, so the time taken to complete the half-turn is at most the change in dy / V, multiplied by 2 to also account for dx.

```
[
  {
     /* Robot state after part 2 */
     /* Recall that so far, we have spent 2/V time circumnavigating an obstacle */
     {?(dx=0 & dx^2+dy^2=1 & dist(x,y,rogx,rogy) > r+rogr );};
     /* Follow an arc of radius larger than the obstacle size until you reach horizontal again */
     {trackr:=*; ?(trackr>0 & trackr > rogr + r & trackr <= maxR);};
     {t:=0;};
     {x'=v/trackr*(y-(y-trackr*dy)), y'=-v/trackr*(x-(x-trackr*dx)),
      dx'=v/trackr*(y-(y-trackr*dy))/trackr, dy'=-v/trackr*(x-(x-trackr*dx))/trackr,
      t' = 1 & dx<=1 }
  }
]
((x-(x-trackr*dx))^2+(y-(y-trackr*dy))^2=trackr^2 & dx^2+dy^2=1 & dist(x,y,rogx,rogy) >r+rogr
  & t<=2*(1-dy)/V
  & gx>=x
  & y>=0
  & dy>=-1
  & (y=0 -> (dy=-1 & dx=0 & t<=4/V)) ) /* Goal & Safety condition */
```

Figure 4.4: Arc Model: The "dist(x, y, rogx, rogy) > r + roger" postcondition maintains obstacle avoidance. The "t<=2(1-dy)/" postcondition bounds time. The "y=0" and associated*

postcondition set the rover up in the right position and orientation to begin a point turn to then continue traveling towards the goal on the right.

5. Proof Methodology

5.1 Straight-Line Driving Model

The interesting case of the proof is when the rover has not yet reached the goal. In this case, we make an argument that the distance to the goal plus the distance from the old x prior to this time step is bounded by the distance from the old x to the goal. Then, we relate the distance from x to old x via time t, noting that t equals the change in globalTime. The combination of relationships uses and proves the invariant "overdist(x,y,gx,gy) <= ((timeBound-globalTime)*V)+ gb". The other portions of the invariant prove by differential invariants, minor cuts, or automatically.

5.2 Obstacle Avoidance Model: Point Turns

This portion proves almost entirely via differential invariants. The interesting component is proving the postcondition "t<=2*(1-dx)/V". While this also proves by a differential invariant, one must first cut in that dx and dy are <=1. It is worth noting that the postcondition includes (dx=0 -> t<=2/V) because it is the case that we are most interested in the run that has dx=0 i.e. the rover point upwards.

5.3 Obstacle Avoidance Model: Arc

This is the most complicated component to prove. To start, one must cut in that the rover is on a circle of radius trackr (x -(x-trackr*dx))^ 2 + (y-(y -trackr * dy))^2 = trackr^2). Then, most components prove via differential invariant. The most important postconditions are the time postconditions "t<=2*(1-dy)/V" and "(y=0 -> (dy=-1 & dx=0 & t<=4/V))". Proving the former, via a cut limiting dx and dy to be less than or equal to 1, enables proving the latter. The latter postcondition is critical because it shows that, in the case where the arc reaches the y-axis, the rover is positioned pointing downward, as desired before the final point turn, and the time taken is bound by 6/V.

6. Conclusions

The models prove that a drive in a straight line will take no longer than calculated using basic physics. The models also provide insight into point turns and arcs. The point turns are proven to take less than or equal to 2/V time, or 40 seconds in our scenario. The arc is proven to take 4/V, or 80 seconds, of time in our scenario. Also, at 0.05 m/s, the rover takes 100s to traverse a window in a straight line. Altogether, the time taken to traverse a window is the time to drive in a straight line (100 s), the time for two point turns (2 times 40 s) and the time to drive in an arc (40 s). Altogether, This is a total of 220 seconds, much less than the requisite 648 seconds.

Even if we were to include a point turn at the beginning of the window traverse to make up for orienting such that the rover drives along an x-axis, the point turn would only take an additional 40 seconds, for a total of 260 seconds to traverse a window. This is still much less than the maximum of 648 seconds.

Note that, as an additional result, any path to avoid an obstacle can be represented as the first quart point-turn (2/V time), an arc less than or equal to a full arc (2* 4/V time), and a point turn less than or equal to 360 degrees (4* 2/V time). As such, an upper bound on the time to avoid an obstacle when coming to an obstacle from a direction and going to an arbitrary point and orientation relative to that direction is 18/V.

7. Alternate Approaches

The most challenging portion of this project was simplifying the problem in a manner that still enabled capturing time. Alternate approaches attempted included modeling the rover as a rover that navigates using point turns and straight drives and a single model that combined all three of the models presented in this paper. The former created complexities in binding the evolution of x or y. The latter created complexities in both tracking increase in time and verifying postconditions of intermediate steps, such as the point turn or the arc.

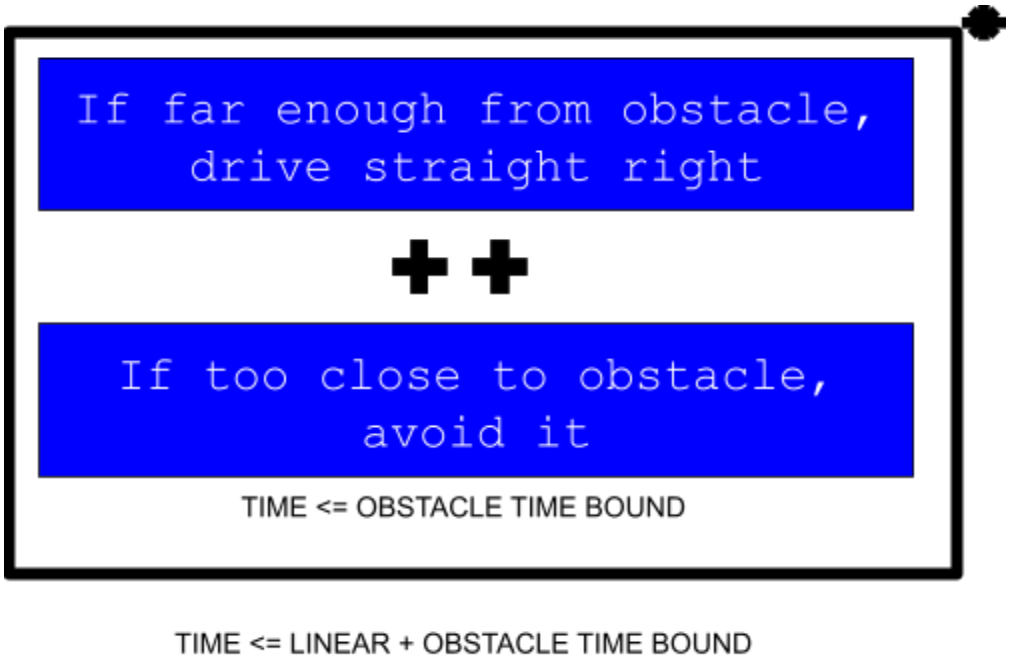

Figure 7.1: A graphical representation of the combination of the three models.

8. Discussion: Additional Scope

While these models are the first steps that prove that lunar micro-roving missions can indeed occur within the time constraints of a lunar day, this model is just the first step. More complex models representing multiple obstacles rather than a single obstacle would be an initial next step. The ideal model would use the rock distribution on the lunar surface to create a map that would be referenced for obstacle presence, might extend beyond considering one-dimensional travel and two-dimensional obstacle avoidance, and would model a more general controller that drives using a discrete set of drive arcs.

9. Acknowledgements

I would like to acknowledge and thank Dr. Platzer and Aditi Kabra for teaching us this semester. I would also like to thank the teaching staff for answering questions quickly and patiently.

10. APPENDIX A: Important Assumptions and Descriptions

Preconditions	**Justification**
globalTime=0	Mission starts at time 0
FOV = 5	Robot sees 5 meters ahead. This is a relatively small number compared to larger rovers. However, it is a reasonable number for the usable visible field from micro-rovers [1].
V=0.05	Robot speed is 5 cm/s
timeBound + circumventTimeBound <=XXX	Fractional time of entire foray to cover a single window.
overdist(x,y,gx,gy)<=FOV	The local window goal must be within the area that the robot can see.
rogr > 0	Obstacle exists
rogr + r < FOV	The obstacle must be smaller than the window size, and the rover should be able to fit in the window as well.
rogy = 0	The obstacle is centered on the x-axis. Otherwise, one direction would be a shorter distance, not stressing the time to circumnavigate the obstacle
rogx > x	The obstacle is to the right of the rover. This is without loss of generality because, if this were not the case,

	The rover could perform a point turn and orient such that the obstacle would be to the right of the Rover.
maxR >= rogr	The Rover should be able to take an arc large enough to go around the obstacle.
y=0 &	The Rover starts on the x-axis.
gx > x & gy = 0	The goal is also to the right of the Rover.
gx - x > V	The goal should be far enough from the rover at the start.
T=1	The time-triggered straight line model has a period of 1 second.
gb >= V	The bound around a goal should be larger than the distance one could travel in 1 time step. This is to facilitate guarantees of reaching the goal.
dx^2 + dy^2 = 1	Dx and dy Should represent a legitimate Rover orientation.
dx=1 & dy =0	Rover oriented to the right.
r >0	Rover has some size.
dist(x,y,rogx,rogy) > r+rogr	The rover does not start out hitting the obstacle.

11. References

[1] Varsha Kumar, Shyam S. Sai, Srinivas Vijayarangan, David Wettergreen, Heather Jones, Patrick Callaghan, Haidar Jamal,and William L. Whittaker. “Formulation of Micro-Rover Autonomy Software,” iSAIRAS, October 2020.

[2] Jan-David Quesel, Stefan Mitsch, Sarah Loos, Nikos Arechiga, and Andre Platzer. “How to Model and Prove Hybrid Systems with KeYmaera: A Tutorial on Safety,” International Journal on Software Tools for Technology Transfer, February 2016.

[3] David Wettergreen and Michael Wagner. “Developing a Framework for Reliable Autonomous Surface Mobility,” iSAIRAS, Turin, September 2012.

[4] StackOverFlow. “How to tell whether a point is to the right or left side of a line,” https://stackoverflow.com/questions/1560492/how-to-tell-whether-a-point-is-to-the-right-or-left-side-of-a-line.

[5] Andre Platzer and Yong Kiam Tan. “An axiomatic approach to existence and liveness for differential equations”. Formal Aspects of Computing, 2021.

[6] William L. Whittaker. “Space Robotics,” 2021.

[7] Indian Space Research Organisation. “Chandrayaan Engineering the Future of Lunar Exploration, ” https://www.isro.gov.in/chandrayaan2-spacecraft.

[8] Britannica. “MarsExploration Rover,” https://www.britannica.com/topic/Mars-Exploration-Rover.

[9] Lydia Schweitzer and Griffin Tang. “Mission Operations For Autonomous Science-Driven Lunar Micro-Roving,” iSAIRAS, October 2020.

[10] Andre Platzer. “Logical Foundations of Cyber-Physical Systems,” Springer.

[11] Aditi Kabra. “Logical Foundations of Cyber-Physical Systems,” Fall 2021.