The ARLISS Project
CS Senior Capstone

Steven Silvers, Zhaolong Wu, Paul Minner, Zachary DeVita

Abstract

Oregon State University's contribution to the ARLISS Project is a collective effort of several engineering disciplines to safely land a soda-can sized satellite, which had been ejected from a rocket at an altitude of 12,000 feet above ground level, and then have it autonomously navigate to a specified set of coordinates. One of the primary objectives of this project is for the satellite to gather data during its expedition, and then, of course, return safely to its destination. This type of technology could potentially be utilized as a means for data collection and exploration of alternate planets. The autonomous driving is accomplished using GPS, and a system of radar and cameras to detect dynamic conditions and avoid obstacles in the satellite's path.

PROBLEM DEFINITION

Autonomously-driving vehicles have recently transitioned from the realm of science fiction to reality. Companies such as Google and Tesla are creating self-driving vehicles, some of which is even being used by the general public. Self-driving vehicles are a new technology, however, and there are still many hurdles to overcome. This project will combine autonomous driving with rocket science to create an autonomously-driven rover which will be launched 12,000 feet into the air, land, and navigate itself to a predefined destination. The problem we are tasked with completing is to get this rover returned safely to a specific set of coordinates after it has left the rocket. Software must be developed which will allow the satellite to separate from its parachute after it reaches the ground. The software then must autonomously navigate the satellite to a nearby GPS beacon. As a secondary challenge, we must also collect data from the rover while it is still in the air. One challenge we face is not knowing what hardware we will actually be working with until it has been developed by the mechanical and electrical engineers that we are working with. Another challenge will be weather. With the satellite being the size of a soda can, a strong wind could easily add 5 to 10 miles to the satellites trek.

PROPOSED SOLUTION

This software will utilize a combination of GPS coordinates and an obstacle avoidance system. Sensors will be attached to the satellite, which the software will then use to analyze its surroundings in a dynamically changing environment. The software will include algorithms for operating the various sensors on the satellite in order to get an effective analysis for the topography of the ground in its path.

The satellite will be navigating across a dry lake bed, so obstacles will consist primarily of cracks in the ground, and tire tracks. Both of these obstacles pose a significant threat to the success of the expedition, and could lead to the satellite getting stuck. Avoiding these obstacles by finding alternate routes around them is the most crucial aspect of the software. A GPS will be used to determine a general direction to travel, but the obstacle avoidance system will determine detour routes around potential obstacles. Additionally, if the obstacle avoidance system fails and the rover does get stuck, the software should attempt to free the rover from whatever is impeding its progress.

As for the secondary objective, the satellite will be gathering atmospheric data during its descent to Earths surface. To do this, we will write software which takes periodic measurements from various sensors, such as a temperature and barometric data, which the satellite will then transmit to the team on the ground in real-time. This data will be collected and displayed to the team in real time via a graph or some similar graphical user interface.

At Expo we will demonstrate our rover's path finding abilities by creating an obstacle course for it to navigate. We will also include a simulation of the data that might have been collected while airborne, and then use this data to show the graphical user interface while it is running in real-time.

PERFORMANCE METRICS

Getting good performance metrics is especially important in our case, because we have three teams, Mechanical Engineering, Electrical Computer Engineering, and Computer Science working together to conquer those problems.

Since the weight of the rover can only be 350 grams and the size can only be that of a soda can, custom hardware will be created to drive the rover. Because of these constrains, the team will need to design an operating system from scratch for the rover which performs the following requirements. First, a data transmission protocol that is reliable enough to handle large amounts of data throughout the entire process, from the deployment at 12,000 feet above ground level, until its arrival at the target. The team will evaluate this from the data transmission rate, processing rate, data integrity, anti-jamming, etc. Second, an efficient autonomous navigation system that leads the rover to the GPS location. This navigation system must also include a crash recovery system. If the navigation software fails to recognize an obstacle, the software should attempt to free the rover from that obstacle. These requirements can be tested by simulating the environment the rover will drive through. Third, the operating system must be efficient. Power will be a significant hurdle in this competition, so creating an operating systems which is as simple and optimized as possible will give our rover the best possible chance of getting to our target destination. The team will evaluate this by using existing optimization tools such as System Power Optimization Tool(SPOT).

Even if the rover doesn't successfully get to it's destination, the project may not be a complete failure. Judging each of these performance metrics separately will give us a better idea of how well our project succeeded than if we just determined if our project succeeded based on if our rover successfully got to it's destination. For example, the rover may run out of battery before reaching it's destination, but we could still determine if our navigation

system was a success or failure. No project goes entirely smoothly, and using this system, our project doesn't have to be perfect to determine if we succeeded.

Steven Silvers

_____

Signature

_____

Date

Zhaolong Wu

_____

Signature

_____

Date

Paul Minner

_____

Signature

_____

Date

Zachary DeVita

_____

Signature

_____

Date

Nancy Squires

_____

Signature

_____

Date