Autonomous Navigation System for the University Rover Challenge

Members: Yu-Shih Chen, Marielle Eusebio, Zeyuan Hou, Wanyue Zhai
Mentors: Prof. Chen-Nee Chuah & Teja Aluru
Department of Electrical and Computer Engineering

INTRODUCTION

In collaboration with the RoveCrest student engineering club, the autonomous driving system was developed to participate in the 2021 University Rover Challenge. It aims at developing AI-driven sensing, perception, and navigation tasks to autonomously travel to a site.

UNIVERSITY ROVER CHALLENGE

The University Rover Challenge (URC) is an international robotic challenge that challenges students to build remotely operated rovers that can accomplish a variety of tasks that might assist astronauts working on Mars in the future.

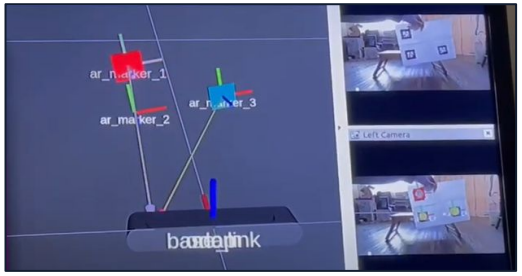
Out of the four missions (Science Mission, Delivery Mission, Equipment Servicing Mission and Autonomous Navigation Mission) we're exclusively focused on the Autonomous Navigation Mission.

The Mission requires rovers to:

- Autonomously navigate to a position given a coordinate
- Stop within 3m of a marked post
- Drive through two posts marked with AR Tags to denote a "gate"
- Avoid obstacles on the terrain

COMPONENTS & SOFTWARE USED

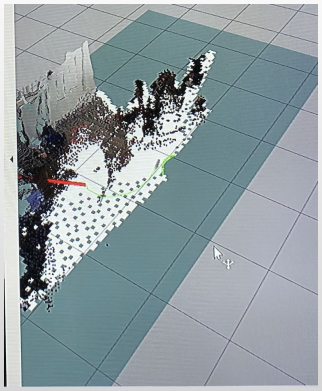
NVIDIA Jetson TX2	Traxxas Slash RC Car
Stereolabs ZED Camera	ROS, Melodic Distribution



RESULTS

Autonomous Mapping & Route Planning

- With input from the ZED Camera, the system can generate a map
- Map continues to update until process is terminated
- Rover moves according to planned route, with steering and throttle commands adjusting accordingly
- Remote goals are set through a ROS Network when car is in operation



AR Tag Detection

- Recognizes either singular tags or bundles in groups of three
- Able to distinguish between each tag and mark with unique identifiers
- Tags and bundles can be seen using the 3D visualization tool RVIZ
- ROS Node created to extract positional data and identify from each tag detected

Pit Detection

- Pit Detection Algorithm able to detect "deep" portions of from the camera input
- ZED Depth Sensing API inaccuracies with depth map
- Dim environments especially difficult to use

Emergency Mechanism

- Car pulls in positional and velocity data from camera
- Car can recognize when it's in a "Emergency" situation and sends a message to the terminal
- Sends car throttle and steering commands in an attempt to escape

FUTURE WORK

Expanding the project requires all the features to work integrated together as well as improved to be more robust.

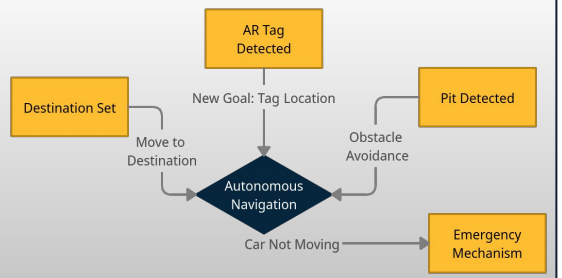
- The AR Tag system needs to be able to send navigation goals to the tag's location
- AR Tags a certain distance from one another need to be recognized as a gate to send goals
- Pit detection works only in 2D, functionality needs to be expanded to 3D
- Improve pit detection algorithm to work in several lighting situations
- The Emergency Mechanism is a singular path of events, needs to have a dynamic response
- Features need to be working all together with some features taking precedence over others

FEATURES

To complete the various tasks outlined in the University Rover Challenge, we focused on 4 main features:

Distance Measurement & AR Tag Detection	Depth Perception & Pit Detection
Autonomous Mapping and Navigation	Emergency Detection & Escape Mechanism

DESIGN



The base of the autonomous navigation system is built off the ROS package move_base, which then all the features are built off. Each of the four features were developed separately from the navigation package to ensure it worked prior to integration.

Fig. NVIDIA Jetson TX2 on RC Car with ZED Camera