ET NavSwarm

Sam Muldrow, Cooper Montgomery, Nate Cannon, Anuj Joshi, Nick Snyder
Computer Science, University of New Hampshire, Durham, NH 03824

Introduction

- The project goal is to create an autonomous swarm of five, four-wheel skid steer rovers
- The rovers are intended to survey dangerous environments on earth and extraterrestrial bodies
- The robots use GPS waypoint following and obstacle avoidance algorithms
- The swarm is intended to utilize graduate student particle swarm optimization (PSO) algorithms
- A PSO algorithm is modeled after herds of animals in nature

Rover and Autonomy

Figure 2: Rover Swarm

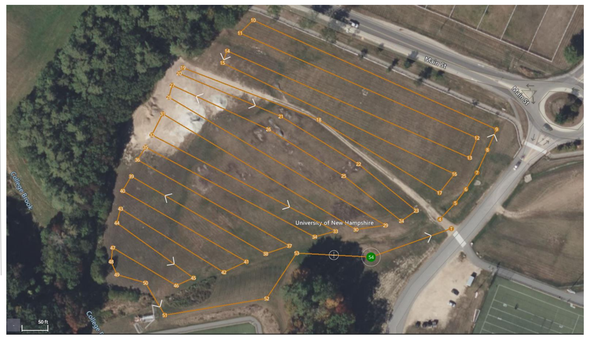

Figure 2: Boulder Field and the swarm's path

Requirements

- Robot should be able to detect an object using LiDAR
- Robot must be able to avoid objects it detects
- Robot should be able to accurately detect its GPS location
- Robot must be able to communicate with other robots using XBees
- Robots should be able to disconnect and reconnect to the ground station software as needed

Methodology

- Each rover functions autonomously using Python and the ROS2 Python module
- ROS2 allows each component of the rover to send and receive commands
- Components include LiDAR, motors, PixHawk 4 (GPS and IMU), ESCs, XBees, and Arduinos
- For testing, five rovers were placed outside and sent waypoints from the ground station via XBee Modules
    - Waypoints followed a lawnmower pattern

Software

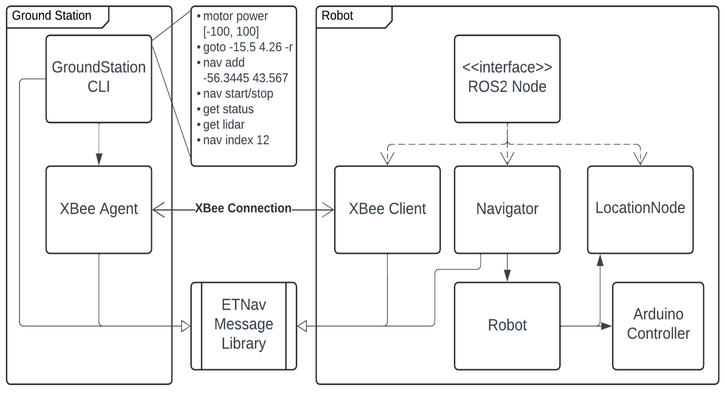

Figure 4: Class Diagram

Figure 4 describes the software systems of the ground station and robots.

- CLI creates ROS messages and sends them via XBee Agent.
- Navigator interprets messages and sensor data to drive.
- Robot controls the motors and reads sensor and position data.
- XBee Client, Navigator and LocationNode are ROS nodes.

Results

- During the field test, two of the five rovers managed to complete the mission
- Robot issues due to hardware issues
- Uneven terrain affected LiDAR and drivetrain systems
- Due to the GPS precision, the waypoint following works optimally when the waypoints are approximately 5 meters apart.
- GPS works most optimally with a clear sky.
- The robot operates better over pavement than grass.

Hardware Interfaces

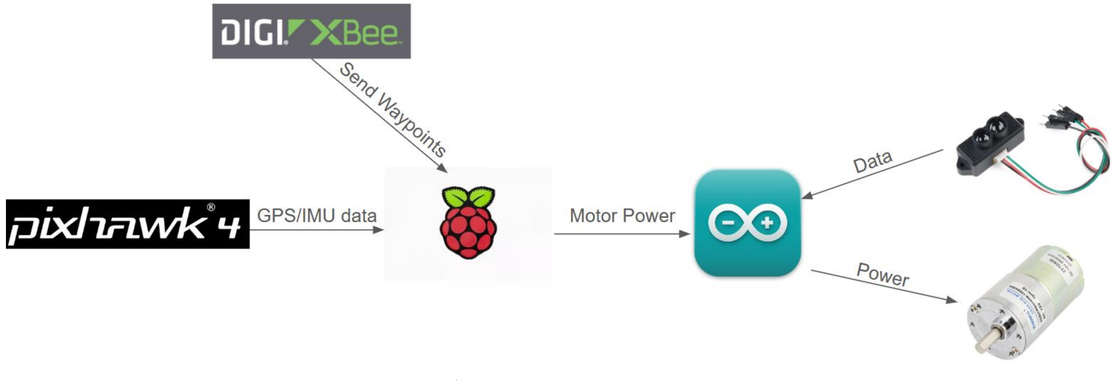

Figure 1: Interaction of Hardware Facilitated by Software Implementation

Figure 1 shows hardware interaction internal to the Rover

- The PixHawk sends GPS/IMU data to the Pi
- The XBee received waypoints from ground station and relays them to the Pi
- The Pi sends the Arduino commands to determine motor power
- Lidar sends data to the Arduino for obstacle avoidance
- Motor power is sent by the Arduino to the ESCs

Acknowledgements

Support from the UNH Computer Science Department and Professor May Win Thein.
Advising from Professor May Win Thein, Professor Matthew Plumlee, and Graduate students ChanLing Beswick and Alex Cook.
Financial support from LunaBot.

References

Ros 2 User Guide ROS 2 User Guide | PX4 User Guide.
Available at: https://docs.px4.io/main/en/ros/ros2_comm.html

Ros 2 Documentation (no date) *ROS 2 Documentation - ROS 2 Documentation: Humble documentation*.
Available at: https://docs.ros.org/en/humble/index.html