NAVSWARM

Team Members: Larry Nguyen, Tim Vigliotte, Ricky Cheng, Jon Ells, Jaiden Evarts, Colin Daly, Connor Fleury, Dylan Russell, Aidan Moore, Nick Miele, Zach Hulecki, J.P. Lapierre, Simon McIntosh, Dylan Brown

Advisors: Professor May-Win Thein, Gregory Hatfield

Hardware

Chassis

- Welded Aluminum

Motors

- DC brushed motor

Suspension

- Stock RC car suspension

Arduino Mega Controller 2560

- Brain of the rover
- Takes input from sensors and sends commands to motors using pre-written code

Electronic Speed Controller (ESC)

- Converts Arduino input signals into motor voltages

Inertial Measurement Unit (IMU)

- Detects rover heading

Global Positioning System (GPS)

- Detects rovers absolute position

XBee and Shield

- Main method of communication within the swarm

Battery

- Powers everything from sensors to motors

Infrared Sensor

- Allow rovers to detect obstacles

Future Hardware (Honors):

Pixhawk Flight Controller

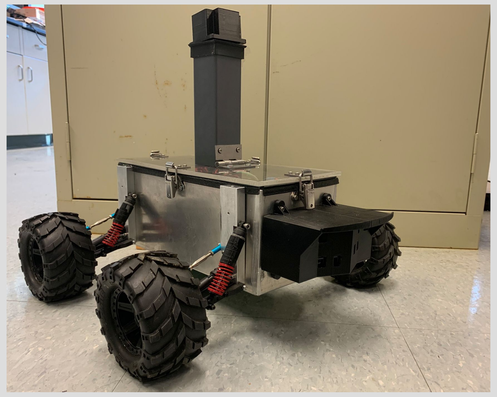
- Allows for ease of rover integration with existing QuadX control hardware

Mission Statement

The goal is to design and test fully autonomous robots in order to implement a particle swarm optimization (PSO) algorithm. This algorithm will then be tested on an open field where the swarm of bots needs to locate and travel to the highest point of elevation.

PSO Testing

- Boulder Field is used as the test location
- Code is uploaded and the IMU is calibrated prior to test
- PSO sends rovers to locations around the test site
- Swarm converges at highest elevation based off simulated potential field



Particle Swarm Optimization (PSO) With Object Avoidance

- Particle Swarm Optimization (PSO) is an algorithm developed off the natural behaviors of birds, fish, and other swarm animals.
- This algorithm uses mathematical principles to have a swarm do optimal movements to efficiently work towards a common goal. This common goal finds the best fitness value of a specific target, such as a maximum elevation.
- The algorithm uses a mix of randomness, the current velocity of the robot, the best fitness values this bot has found, and the best values found by the entire swarm to determine the best locations to send the swarm to.
- The rovers communicate their data in practice by using the XCTU radio communication software, and a brute force networking algorithm to ensure all bots receive necessary data.

New Designs

IMU Tower

- New square, folding IMU tower prevents IMU interference while increasing ease of storage

Infrared Sensor Housing

- The IR sensor housing has been updated to improve the field of view

Winter Treads

- Treads have been designed and tested to allow swarm operation during winter months

Improved Waterproofing

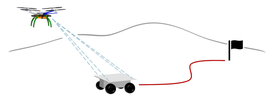
- Modifications of waterproofing of both chassis and new IMU tower further seals electronics from environment

GPS Housing

- Acrylic external GPS housing allows for more satellite connections and stronger signals

Honors Work

- Collaborative effort with Quadx Swarm
- Mission goal: Takeoff, follow and landing of a quadcopter on a modified ET-NavSwarm rover
  - Agents, will communicate throughout the duration of the mission
  - Each vehicle will complete individual missions between takeoff and landing
  - Quadcopter will employ GPS and visual or other close-range tracking for landing protocol



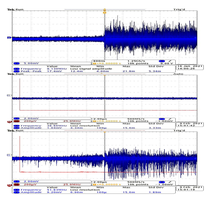
Electrical Engineering

Magnetics Effect on IMU

At Source of Interference

At the Unicorn Horn Height

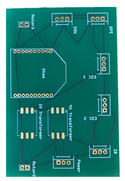
Proposed Location: Chassis Top

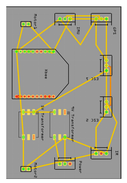


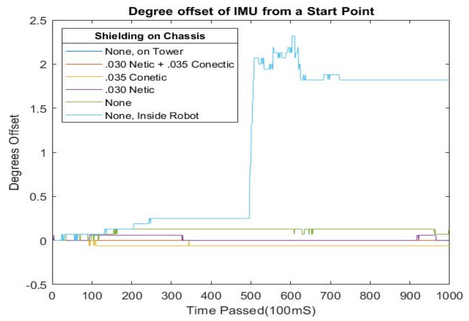
- Indicates Alternatives to using the IMU Tower
- Potential to have similar efficiency with no shielding or tower

PCB

- First adaption
- Easy to mass produce
- Less manual labor
- Easy edit for future edition
- Cost Friendly







Motor Burnout Failure Correction

- What caused this failure
  - Draws .664mA through ESC to power two motors with one ESC
  - When disconnection occurs during high usage total amperage in the ECS connects to a motor and burns out the components
- How to prevent
  - Implementation of Fuse Box into circuitry

Software and Controls

For Each iterations:

Calculate new destination for rovers

Broadcast destination to listeners

While rover is not at destination:

Update GPS and IMU values

Calculate error and adjust rover's path

Move towards goal

If obstacle is detected

Obstacle avoidance

End if

Continue to move towards goal

End while

Turn off motors and broadcast to swarm

End For

- **XCTU**
  - Starts and stops the swarm's RFID communication system
- **Arduino (C++)**
  - Main code - Handles the movement, computation of PSO and hardware interfacing
- **Python**
  - Logger - picks up all broadcasted messages and logs them for debugging and data collection
- **Raspberry Pi**
  - The logger is run on a microscreen using a raspberry pi with the raspbian operating system