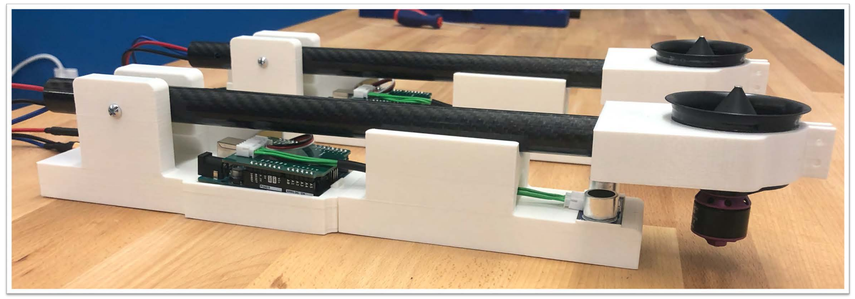
EasyGA
Project Overview

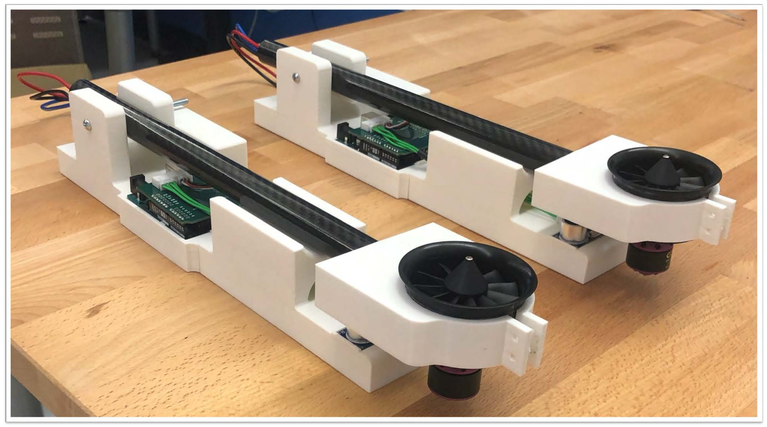
- Build open-source A.I software to apply in real time
- Build multiple A.I test platforms.

PHASE 1: GA ROBOT

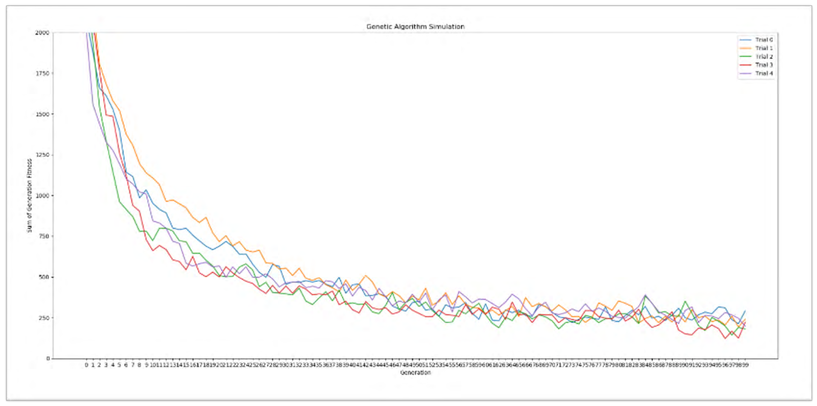
- **Prior Research in Genetic Algorithm to Search the search space for the most optimal hover control.**



- The one DOF device is an evolving altitude controller. The device is made up of an EDF fan, the USRM, a carbon fiber tube, assorted wiring, and two 3D printed parts, the intermediary connector, and platform.

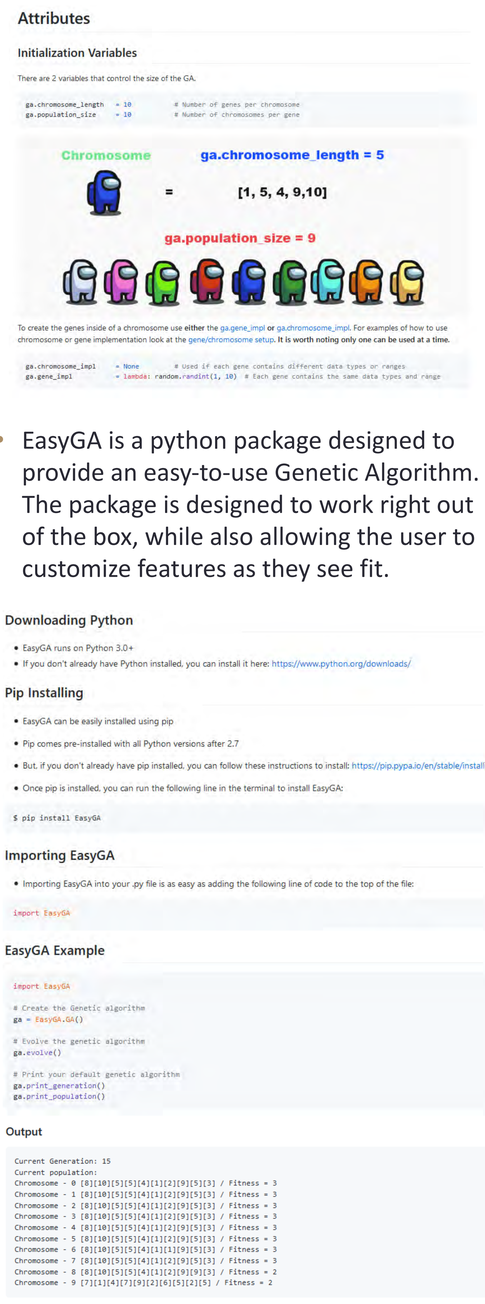


- Simulated run of 5 trails with 10 chromosomes. Population of 40 and 100 generations.



EASYGA SOFTWARE

- **Open-source software that makes genetic algorithms and neural networks easy and useable for anyone.**



- EasyGA is a python package designed to provide an easy-to-use Genetic Algorithm. The package is designed to work right out of the box, while also allowing the user to customize features as they see fit.

Downloading Python

- EasyGA runs on Python 3.0+
- If you don't already have Python installed, you can install it here: https://www.python.org/downloads/

Pip Installing

- EasyGA can be easily installed using pip
- Pip comes pre-installed with all Python versions after 2.7
- But, if you don't already have pip installed, you can follow these instructions to install: https://pip.pypa.io/en/stable/installing/
- Once pip is installed, you can run the following line in the terminal to install EasyGA:

```
$ pip install EasyGA
```

Importing EasyGA

- Importing EasyGA into your .py file is as easy as adding the following line of code to the top of the file:

```
import EasyGA
```

EasyGA Example

```
import EasyGA

# Create the Genetic algorithm
ga = EasyGA.GA()

# Evolve the genetic algorithm
ga.evolve()

# Print your default genetic algorithm
ga.print_generation()
ga.print_population()
```

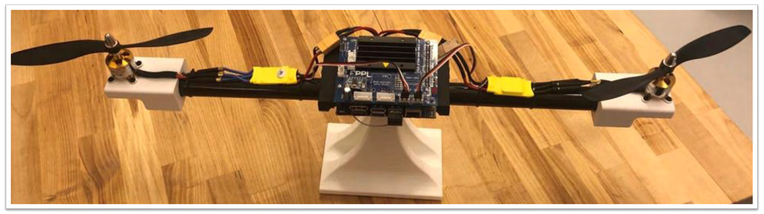
Output

```
Current Generation: 15
Current population:
Chromosome - 0 [8][10][5][5][4][1][2][9][5][3] / Fitness = 3
Chromosome - 1 [8][10][5][5][4][1][2][9][5][3] / Fitness = 3
Chromosome - 2 [8][10][5][5][4][1][2][9][5][3] / Fitness = 3
Chromosome - 3 [8][10][5][5][4][1][2][9][5][3] / Fitness = 3
Chromosome - 4 [8][10][5][5][4][1][2][9][5][3] / Fitness = 3
Chromosome - 5 [8][10][5][5][4][1][2][9][5][3] / Fitness = 3
Chromosome - 6 [8][10][5][5][4][1][1][9][5][3] / Fitness = 3
Chromosome - 7 [8][10][5][5][4][1][2][9][5][3] / Fitness = 3
Chromosome - 8 [8][10][5][5][4][1][2][9][9][3] / Fitness = 2
Chromosome - 9 [7][1][4][7][9][2][6][5][2][5] / Fitness = 2
```

- So easy, that your first genetic algorithm can be made in 5 lines of code. Seriously 5 lines of code.
- Multiple examples to use when your first getting started.
- No need to know every part of the algorithm. Just change what your need or know.

PHASE 2:

Data Driven "Machine Learning" Control Technology



- A one degree of freedom robot that is optimized using a P.I.D controller. This system allows us to apply a GANN "Genetic Artificial Neural Network" to optimize our control network and compare it against a simple P.I.D

Appling GA to Small Autonomous Vehicle

- Phase two of the prototype is controls of multiple small autonomous vehicles.