Arman Paydarfar

M.S. MECHANICAL ENGINEERING – ROBOTICS AND CONTROLS SYSTEMS, COLUMBIA UNIVERSITY

Master’s Thesis

Goal:

- Classification of activity data obtained from piezoresistor based sensors using machine learning techniques
- To determine if machine learning can overcome rudimentary sensor design

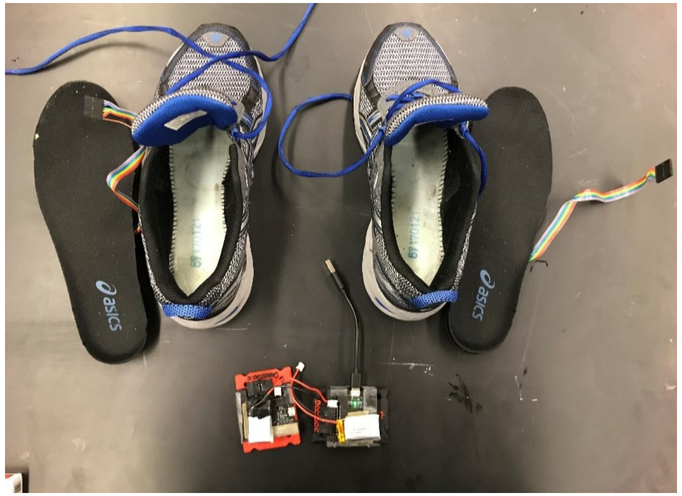
Details:

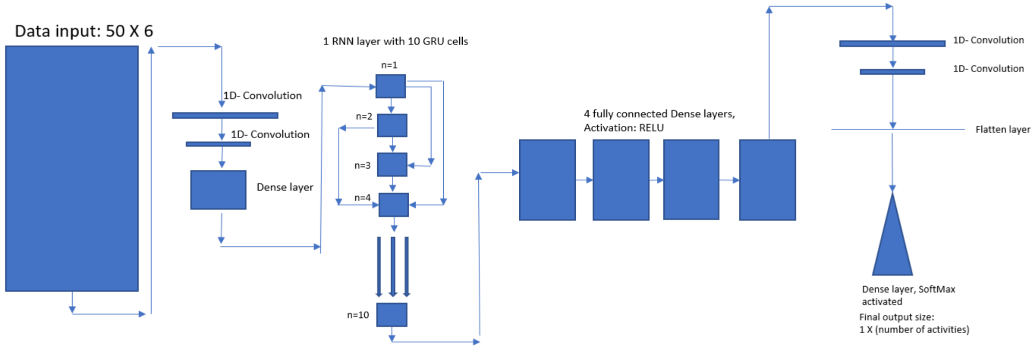
- Data collection was conducted using microcontroller and piezoresistor based sensor system
- Python was used to segment, format and classify data via machine learning
- Utilized recurrent convolutional neural networks for activity classification
- Statistical analysis and data visualization to show results in a succinct way
- Study submitted to IEEE International Conference on Biomedical Robotics and Biomechatronics (February 2020)

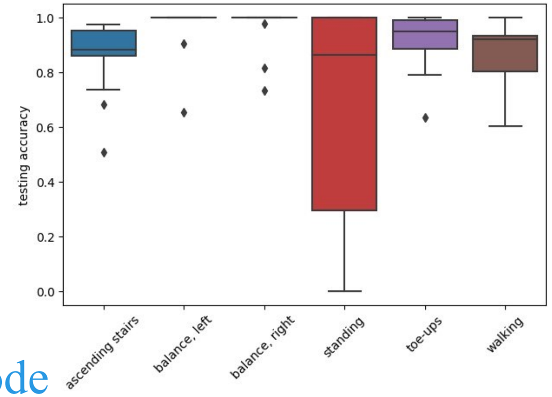
Tools:

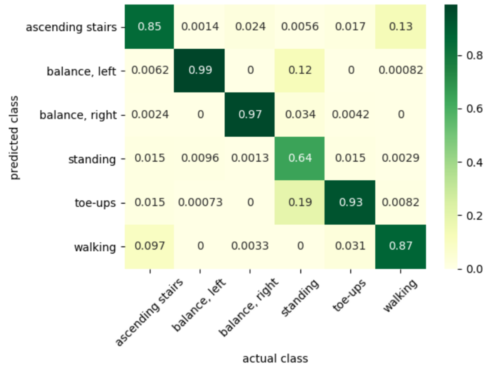
Python, Tensorflow/Keras, Pandas, Deep learning, time series analysis

Segmentation code   RNN and data visualization code   Format inputs code







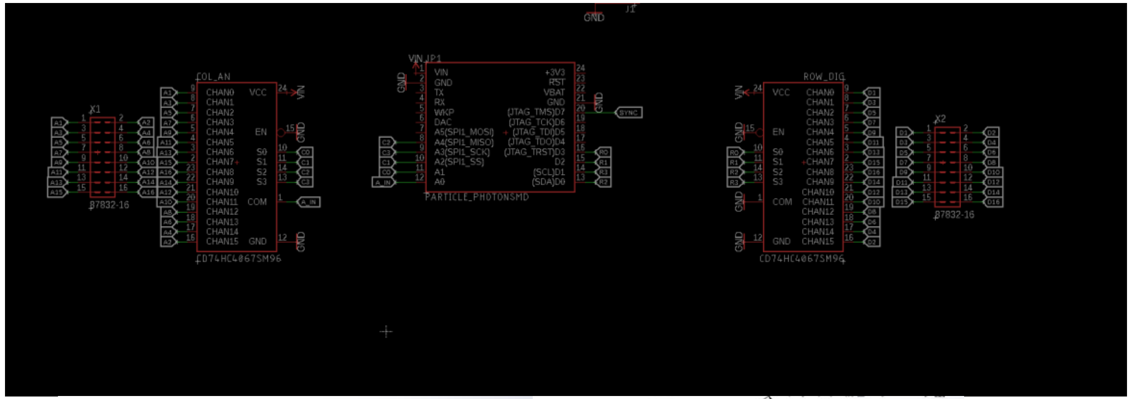


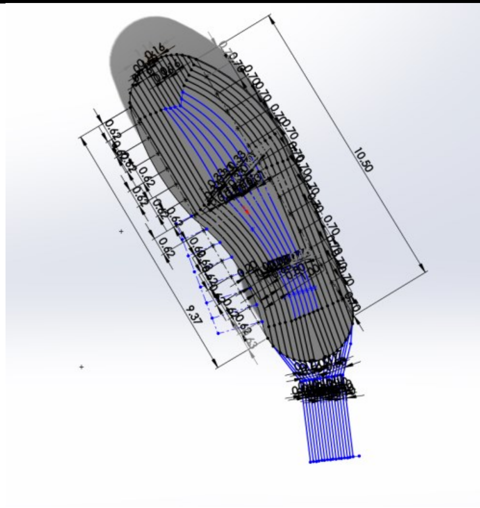
Robotics and rehabilitation laboratory

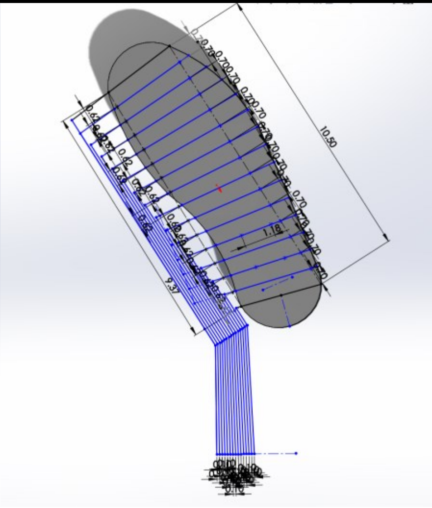
<u>Goal</u>:
- Program microcontroller to collect data with instrumented piezoresistor mat in different circuit configurations

<u>Details</u>:
- Wrote the C++ code and built test circuits, evaluated effectiveness of the sensor system
- Added to base functionality of the microcontroller code to ensure that the system could be used with multiplexers and shift counters, wrote portions of the class to test and validate the system
- Experimented with different designs for future instrumented insoles

<u>Tools</u>:
- C++, embedded programming, circuit testing, soldering, circuit diagrams, reading electronic component diagrams, Solidworks







[C++ code for microcontroller data collection]

[C++ pressure mat object code]

Mechatronics and Embedded Microcomputer Control

<u>Goal</u>:

- Apply different control paradigms to feedback systems

<u>Details</u>:

- On/off control applied to solenoid motor, programmed to respond to timing and feedback
- Stepper motor control (bipolar and unipolar) designed to respond to optical sensor feedback to drive in synchronized, incremental, and wave drive modes
- Brushed DC motor control using tachometer and eddy sensor feedback – driven in proportional, proportional biased, and PID modes

<u>Tools</u>:

C, Assembly language, Mechatronics, Embedded systems

[ON/OFF control assembly code]()

[Stepper motor control in C]()

[DC motor control in C]()

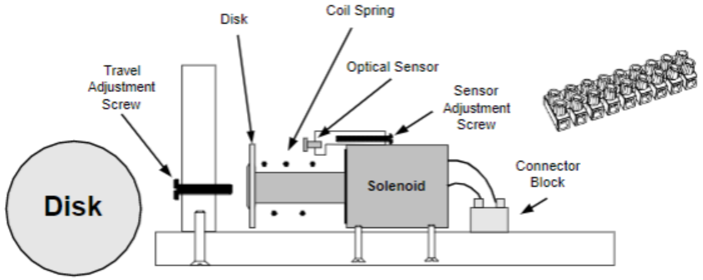


Figure 1. Solenoid Test Stand

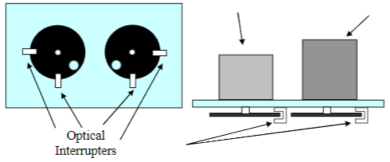


Figure 24. Stepper Motor Test Fixture

The ports assigned to each optical interrupter is shown in Figure 25.

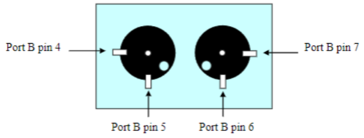


Figure 25. Optical Interrupter Showing Port Pins

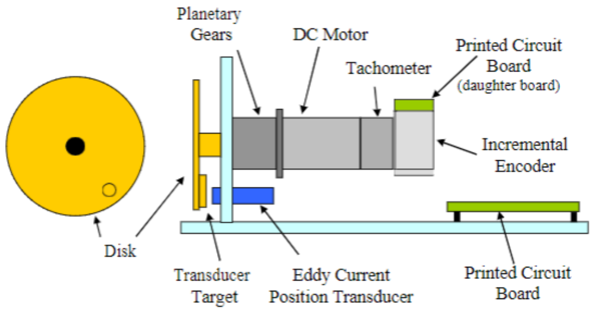
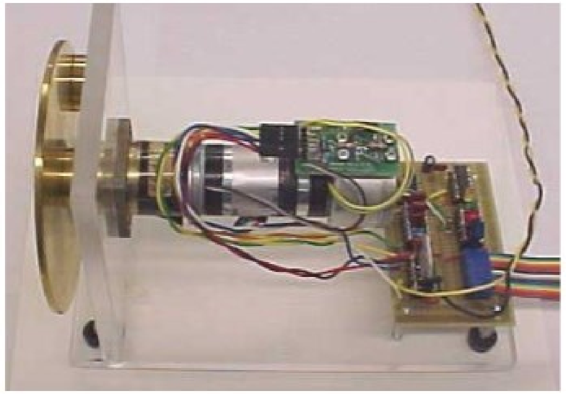


Figure 2. Schematic of the DC Motor System.



ROS for motion planning and sensor fusion

Goal:

- Motion planning implementation on simulated KUKA robot
- Extended Kalman filtering for sensor fusion on simulated 2D robot

Details:

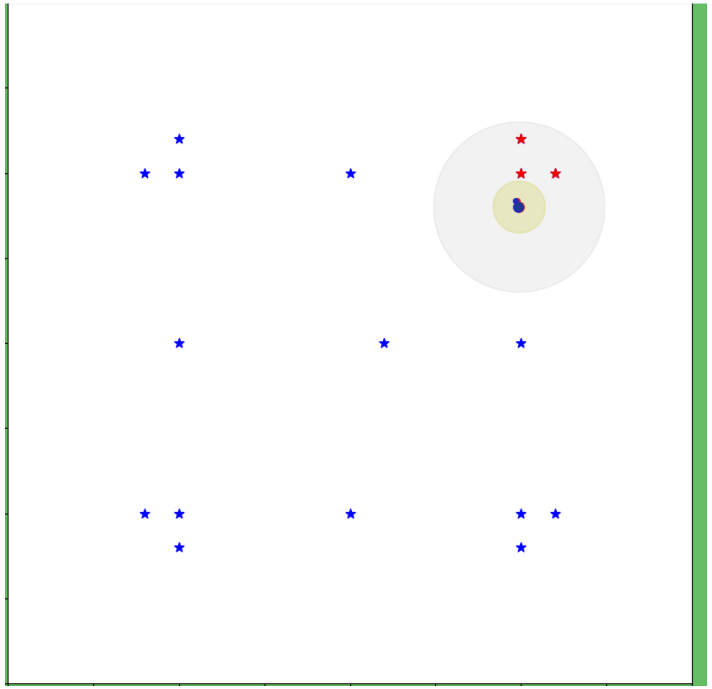
- Implemented rapidly exploring random tree algorithm due to large solution space (too high dimensional for A*)
- Programmed Kuka robotic arm to be able to plan paths in a variety of difficult environments with object obstruction
- EKF implemented with predicted state (red) and true state(blue) shown
- Learned about ROS communication protocol
- Observed when Robot would have lower confidence of position (when there were fewer landmarks in range)

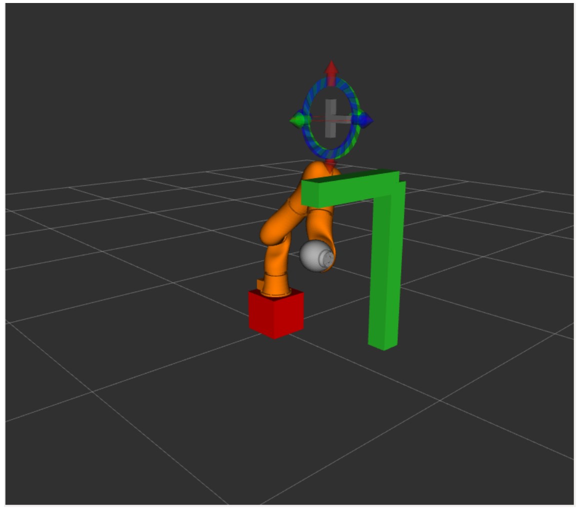
Tools:

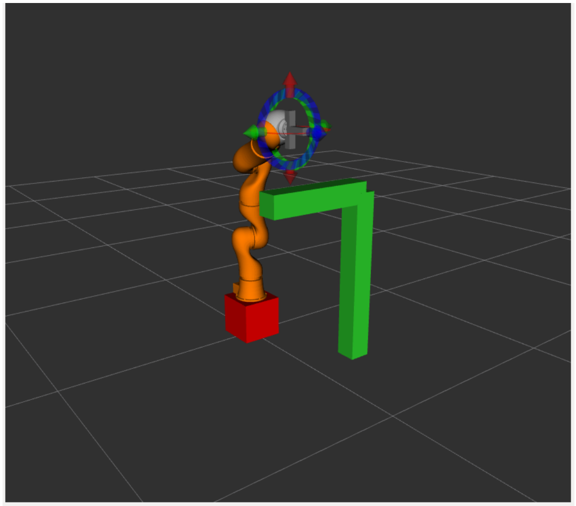
ROS, Python, Motion planning, sensor fusion, numerical inverse kinematics, forward kinematics

Motion planning code

EKF code





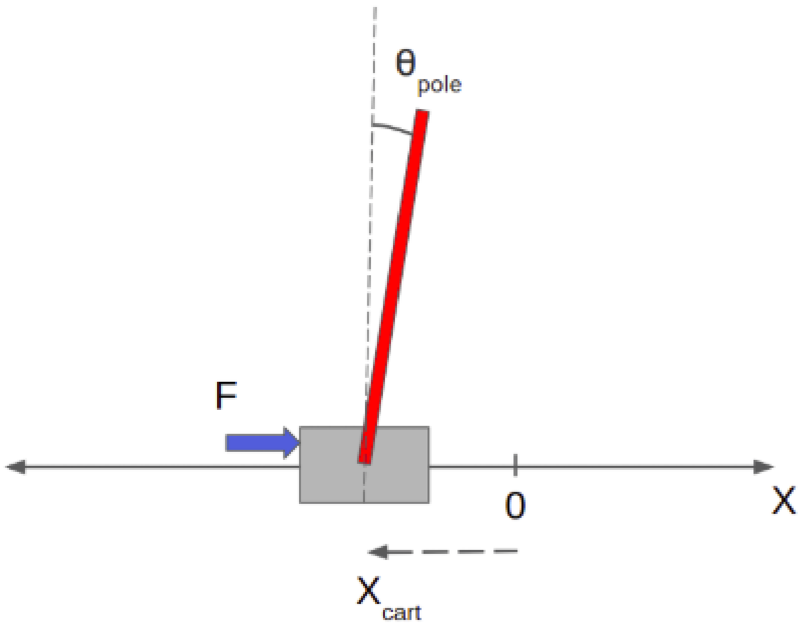


ROS deep Q learning project

Goal:
- Develop an algorithm that will teach simulated robot to play the “cartpole” game

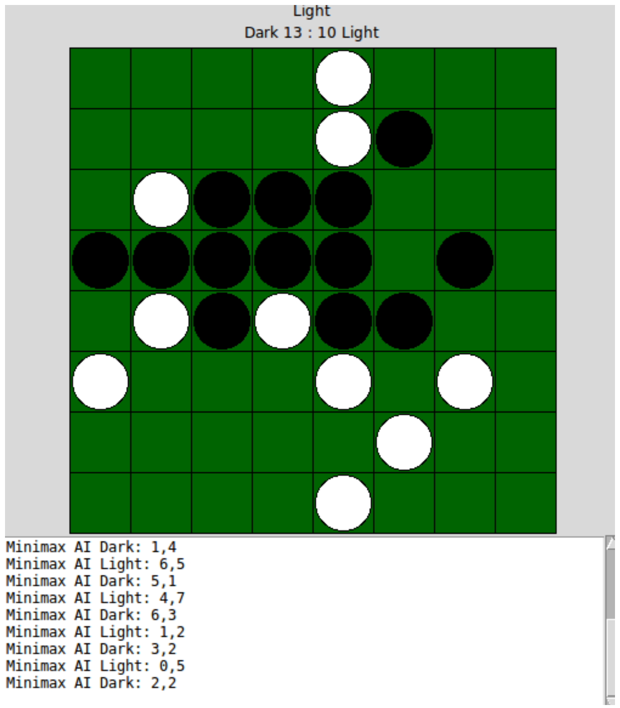
Details:
- Deep Q-learning was used to teach agent to play the game
- Neural network optimization, architecture, and evaluation/training was conducted with Python’s PyTorch toolbox
- Optimal policy was learned by the deep q network, which was able to play the cartpole game and consistently keep the pole upright for over 150 timesteps each time
- ROS service/client API used to gain information about current state of the game

Tools:
Deep learning, ROS, Python, PyTorch



DQN code

AI projects



Npuzzle:

- Breadth first, depth first, and A* search algorithms were used to find optimal solutions
- Different heuristics were tested for the A* search, including the manhatten and misplaced tile heuristics

```
python npuzzle.py
1 4 2
0 5 8
3 6 7
```

Bayesian methods for statistical inference:

- Implemented rejection sampling, likelihood weighting, and gibbs sampling algorithms

Skills learned: Python, AI algorithms

Othello:

- Implemented AI agent to play Othello game against other players and other AI
- Utilized MiniMax search and alpha-beta pruning algorithms

Hidden Markov model implementation:

- Implemented forwards, Viterbi and forward-backwards smoothing algorithms in python to predict weather data
- Was able to test weather predictions as well as tagging different parts of speech

Code:
Npuzzle
Othello
Bayes networks
HMM

Digital Signal Processing

<u>Goal</u>:

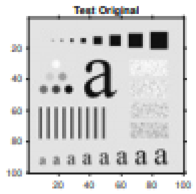
- Apply compressive sensing techniques to enable image reconstruction at less than Nyquist frequency sampling

<u>Details</u>:

- Conduct Fast Fourier transform of the image, to observe the frequency distribution
- Define sampling distributions that may be logical for reconstruction (normal, uniform, triangular, laplace)
- Apply compressive sensing by applying total variation minimization algorithm to minimize l1 error between reconstructed and original image

<u>Tools</u>:

- Compressive sensing, signal processing, MATLAB



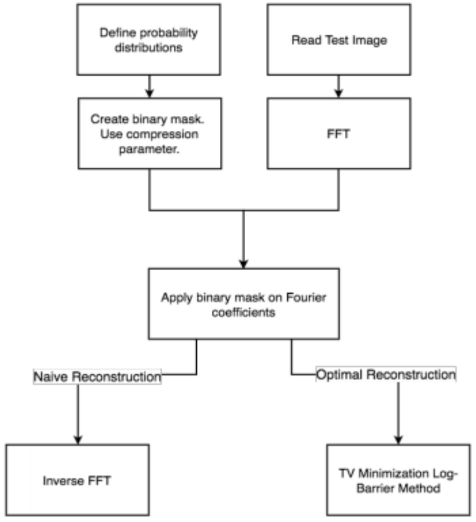


Fig. 2. Flowchart summary of CS-based re-sampling and reconstruction of images.

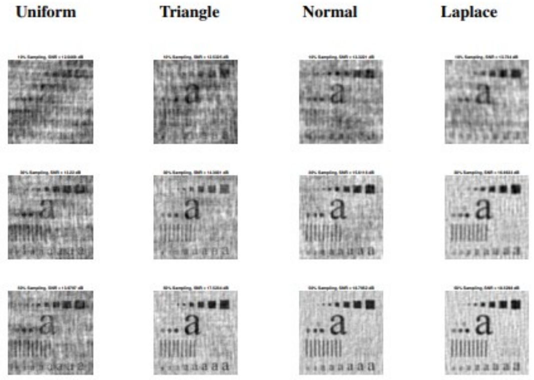


Fig. 4. Naive reconstruction of test image

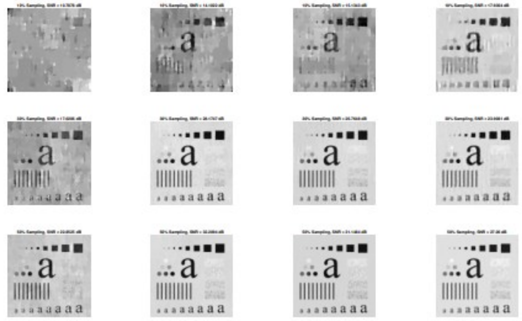


Fig. 5. Optimal reconstruction of test image

Evolutionary Algorithms and Design Automation

Goal:

- Develop a program to self-design simulated robots (using spring and mass building blocks) that will travel the furthest distance using evolutionary algorithms

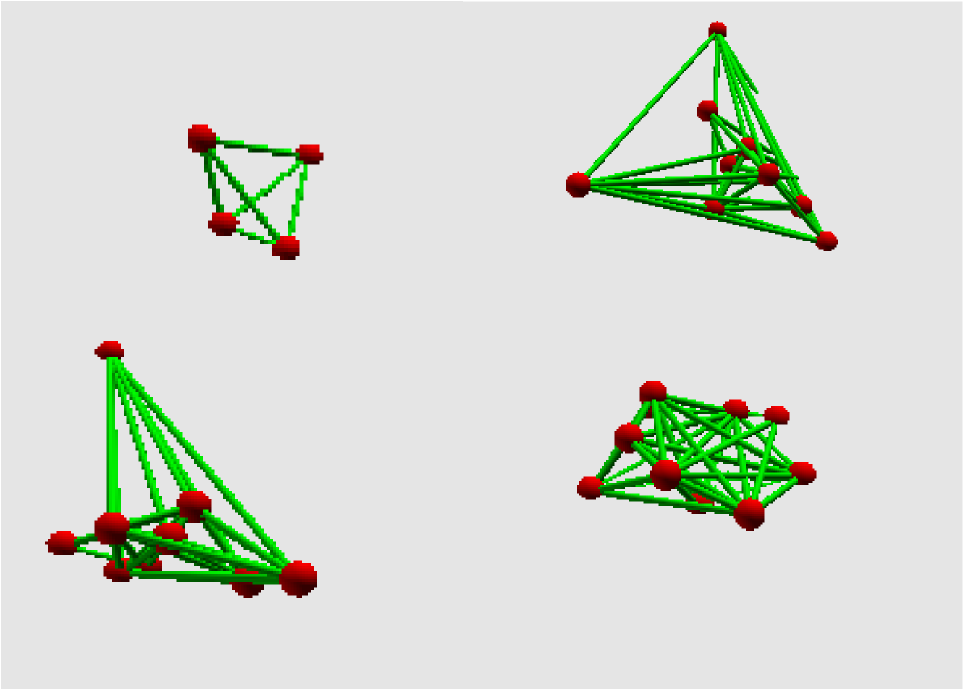
Details:

- Wrote genetic algorithm that automates and evaluates each robot as it is simulated.
- Tracks the performance of best robots, evolutionary paradigm to create next generation of robots from previous
- Visualization toolboxes used to simulate motion and performance of robots

Tools:

Python, Vpython, evolutionary algorithms

[Code sample for phase 1 of project]



Evolutionary Algorithms Project: Equation solver

<u>Goal</u>:

- Find the symbolic equation given a waveform (2D)

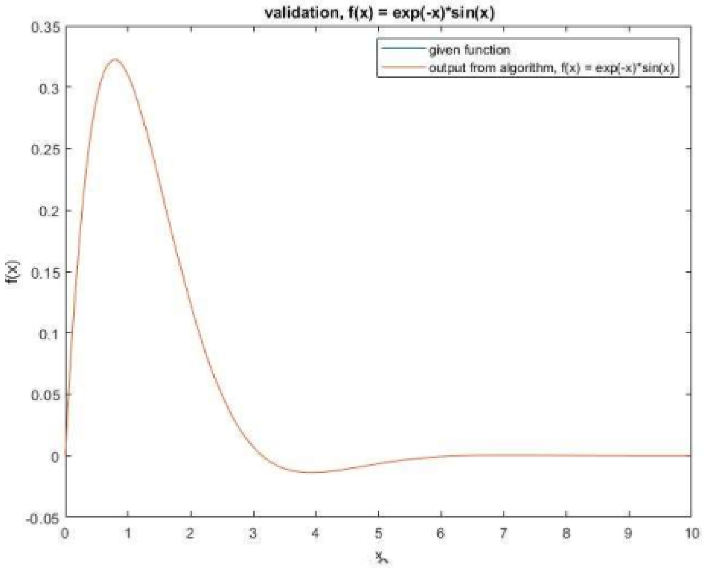
<u>Details</u>:

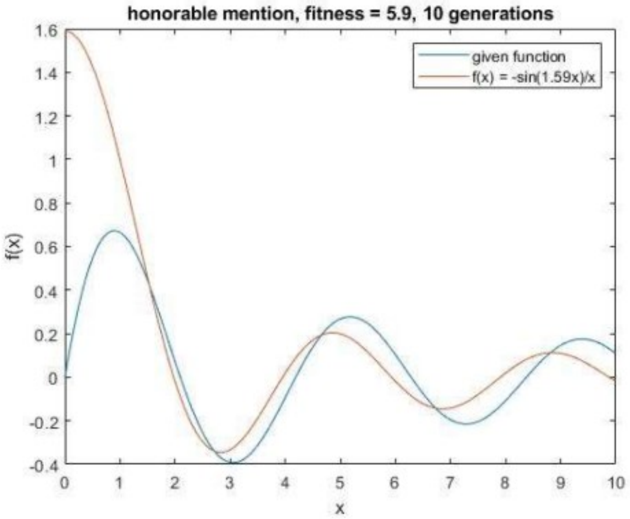
- Genetic programming used with different functional “blocks” for different mathematical operations
- Genetic program searches solution space and “mates” solutions with successful outcomes to produce successful accurate children
- Able to find symbolic equation for most simple equations, modest success with complicated equations
- One major challenge was parsing a symbolic, executable equation from string blocks with mathematical symbols

<u>Tools</u>:

Genetic programming, MATLAB

[Full report, code is at the back (hill climber, genetic programming, random search)]





Computational Fluid Dynamics to simulate Nozzle fuel sprays

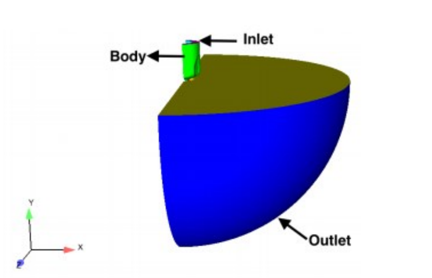
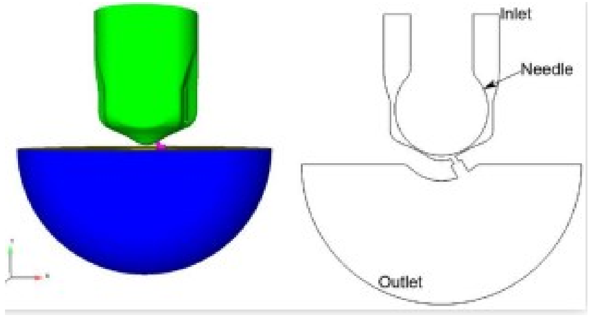


Figure 1: 3D view of the injector geometry, with the inlet, body and outlet labeled



Goal:
- Develop simulation tools to model gasoline fuel injection process

Details:
- Worked very closely with PhD student to further the research, run studies (meshing/geometry, setting up test cases, and analyze/visualize data)
- Difficult due to the multiphase flow properties (gas, and liquid droplets coalescing and interacting)
- Eulerian (field physics) and lagrangian (particle physics) simulation techniques used
- Work was published in 2 international journals
- Research was done to determine single hole and multihole injection interaction of gasoline fuel sprays

Tools:
- Computational fluid dynamics, technical writing, working closely with researchers, heat transfer and fluid mechanics, partial differential equations, mathematical modeling

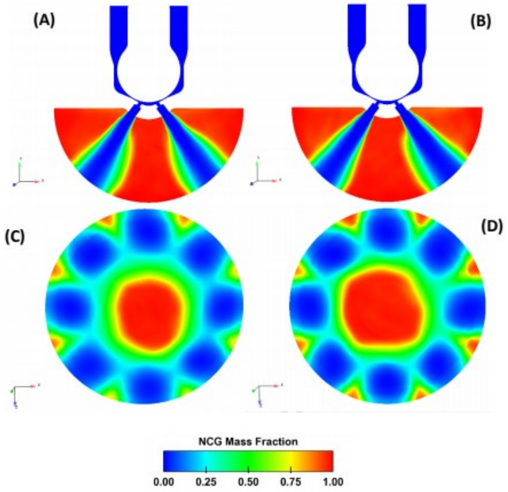
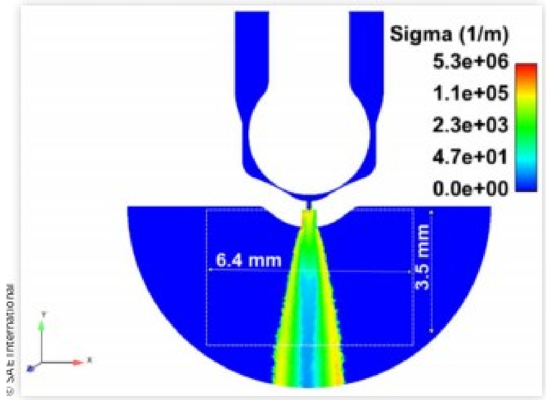


FIGURE 5 XY plane through the computational domain showing the predicted interface-area-density (log-scaled) contours and the region of interest for performing the line integrations.



Publications:
Rachakonda S. K., Paydarfar A., Schmidt D. P., "Prediction of Spray Collapse in Multi-Hole Gasoline Direct Injection (GDI) Fuel Injectors," International Journal of Engine Research, 20(1):18-33, 2019.
Rachakonda, S., Paydarfar, A., and Schmidt, D., "Single-Hole Asymmetric GDI Injector: Influence of the Drill Angle and the Counter-Bore under Flash-Boiling and Non-Flash-Boiling Conditions," SAE Int. J. Engines 11(6):1031-1048, 2018.