1220 Mecaslin St NW Apt 1309
30318, Atlanta, GA

+1 (404) 797 1791

/in/mouhyemen

github.com/mouhyemen

mouhyemen.khan@gmail.com

# PORTFOLIO

MOUHYEMEN AHMED KHAN

## SAFE LEARNING OF QUADCOPTER DYNAMICS USING CONTROL BARRIER FUNCTIONS

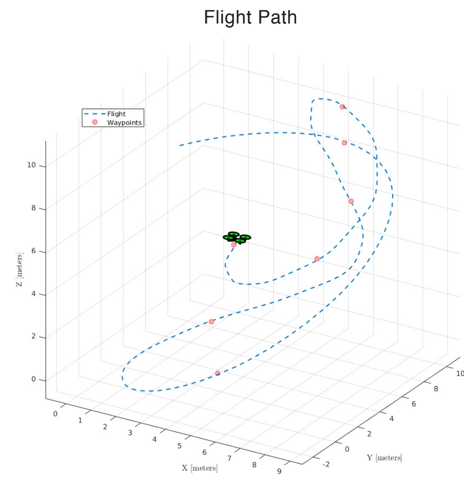
June, 2018 – Present

**Project**
Control theory & Machine Learning Research Project at Georgia-Tech

**Description**
Machine learning techniques have demonstrably proven to improve the dynamics of safety-critical robotic systems. In order to do so, acquisition of data for learning has to be done in a safe and robust manner. The goal is to learn the dynamics of the system while improving its dynamics.

Implemented a cascaded non-linear controller for a quadcopter and simulated its dynamics on MATLAB. Trajectory generation was done using minimum snap trajectory formulation. To compensate for external disturbances such as wind or aerodynamic drag, Gaussian Processes are used to model and learn such unmodeled dynamics.

Control barrier functions appear as QP-formulated schemes with barrier limits imposed on certain states of the system, especially, position and velocity. Exponential Control barrier functions are leveraged for relative degree higher than one. Eventually, the pipeline is being tested and improved to port onto a hardware platform and collect physical experimental results.




**Software**
MATLAB, Python

**Hardware**
Crazyflie 2.0

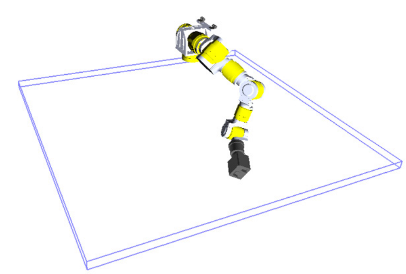
## SEMI-PARAMETRIC ONLINE CONTROL OF 7-DOF MANIPULATOR

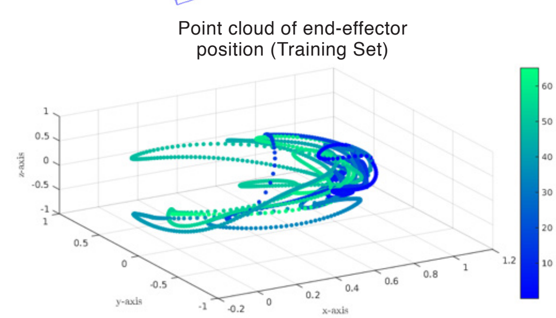
April, 2018 – June, 2018

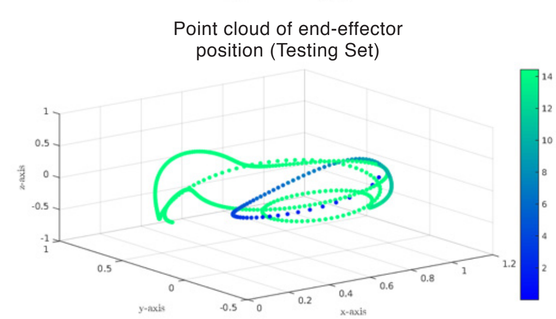
**Project**
Control theory & Machine Learning Research Project at Georgia-Tech

**Description**
Data-driven model control brings together the benefits of both model-free control and purely rigid body dynamics based control. Instead of just doing model-free control, which requires an immense amount of data and computational overhead, or purely parametric method which does not account for unmodeled dynamics, the goal is to implement a semi-parametric method.

Simulated the 7-DOF manipulator on Dynamics Animation and Robotics Toolkit (DART). The controller implemented was based on QP-formulation to perform inverse dynamics. The non-parametric component (unmodeled dynamics) was captured by a Gaussian zero-mean noise and variance. Gaussian Process regression was done on the dataset collected to learn the non-parametric component and hyperparameters for controling the arm with better accuracy. The normalized mean-square error across all 7 joints was on average 11 times better than purely parametric approach.






**Software**
C++, MATLAB
(GPML toolbox)

**Framework**
Dynamics Animation and
Robotics Toolkit (DART)

## SUPERVISED MACHINE LEARNING CLASSIFIERS ON POINT CLOUD DATA

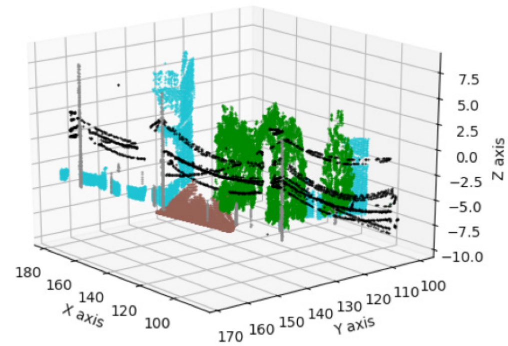
Jan, 2018 – April, 2018

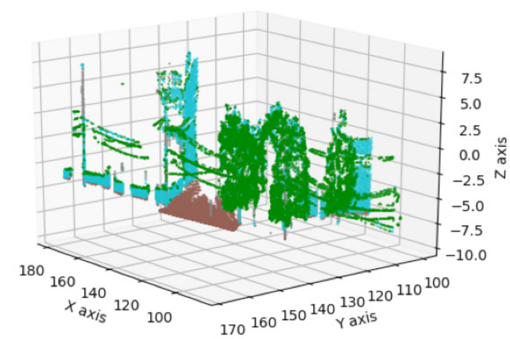
### Project
Statistical Techniques in Robotics, Course Project at Georgia-Tech

### Description
Given the point cloud information of Oakland Bridge in Pittsburgh, machine learning classifiers should classify five different classes – vegetaion, wire, pole, ground, and façade.

The classifiers implemented were – Gradient Descent (98% accuracy), Bayesian Linear Regression (84% accuracy), Gaussian Process Regression (92% accuracy), and Linear Support Vector Machine (72% accuracy).

### Software
MATLAB, Python (OpenCV, Scikit, Point Cloud Library




## UNKNOWN MAZE NAVIGATION WITH TURTLEBOT 3.0 USING ROS/GAZEBO

Aug, 2017 – Dec, 2017

### Project
Introduction to Robotics Research, Final Project at Georgia-Tech

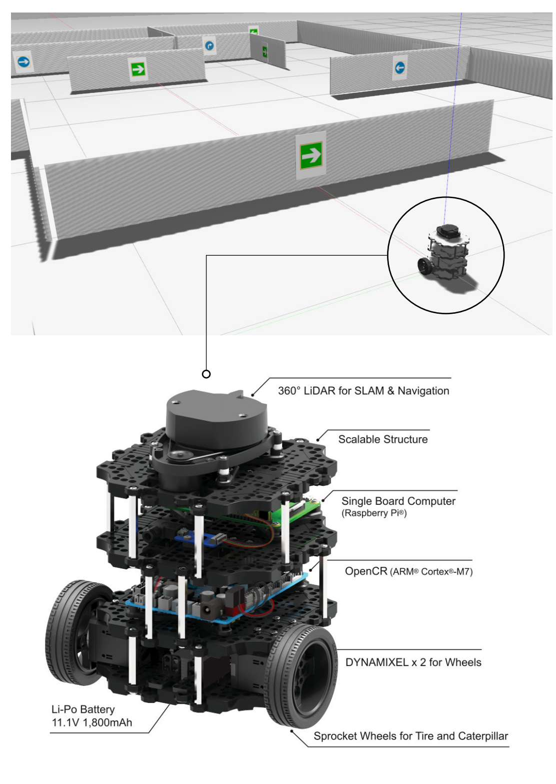
### Description
TurtleBot 3.0 augmented with LIDAR and HD Pi-camera module with on-board Linux-based processing unit, the robot should navigate to the goal in an unknown maze by detecting road signs while avoiding obstacles.

Detected road signs by leveraging SIFT features extraction and k-means clustering classifier. The navigation stack in ROS was utilized for trajectory planning and navigation from current state to goal state. The entire algorithmic pipeline was run using ROS architecture and tested on Gazebo. As part of the course project requirement, the pipeline was also demonstrated on hardware.

### Hardware
TurtleBot 3.0, LIDAR, HD Pi-camera, On-board Raspberry Pi 3.0)

### Software
Python (OpenCV, Scikit))

### Framework
ROS, Gazebo



## OBJECT DETECTION AND CLASSIFICATION USING POINT CLOUD DATA

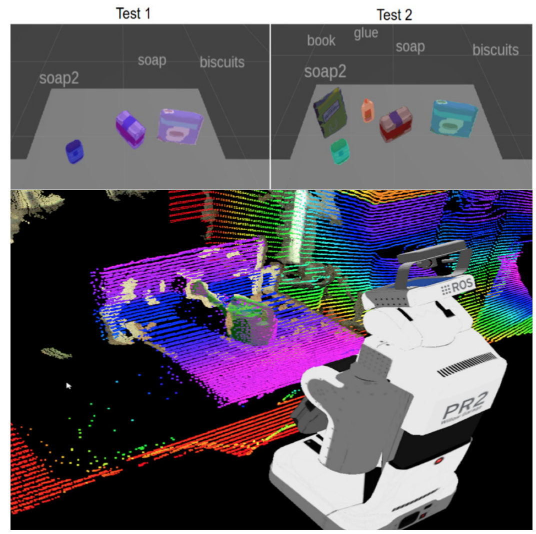
Aug, 2017 – Dec, 2017

**Project**
Machine Learning & Perception, Udacity Robotics Nanodegree Term 1

**Description**
Fitted with an RGB-D (Red, Green, Blue, Depth) sensing camera, with data type in the form of point cloud, the PR-2 Robot should detect and place objects placed in front of it in the designated box.

Implemented perception pipeline which included downsampling voxel elements and performing two layers of filtering – statistical outlier and passthrough. RANSAC plane fitting was performed to differentiate inliers and outliers. Euclidean clustering was performed on point clouds associated with each object placed in front of the robot.

Feature extraction was performed based on histograms generated using HSV color space and surface normals on clustered elements. Support Vector Machines classifier was used to train and test the data set. Successfully detected objects with 92% accuracy overall over different test sets.




**Software**
Python (OpenCV, Scikit, Point Cloud Library)

**Framework**
ROS, Gazebo

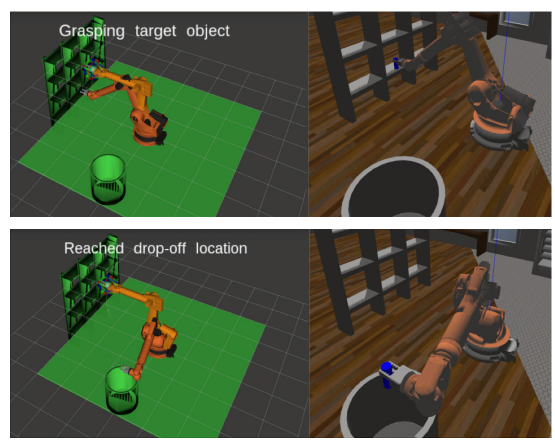
## PICK & PLACE ON 6-DOF KUKA ARM 210 USING ROS/GAZEBO

Aug, 2017 – Dec, 2017

**Project**
Forward/Inverse Kinematics, Udacity Robotics Nanodegree Term 1

**Description**
Inspired by Amazon's Pick & Place Challenge, the project involves performing kinematic computations for picking up items from the shelf and dropping them in the bin.

Performed forward kinematics to find the end-effector's location in the base frame using Denavit-Hartenberg convention. Inverse kinematics was performed for computing six joint angles given the end-effector's position. The simulation environment used was Gazebo and the pipeline was implemented using ROS architecture.




**Software**
Python

**Framework**
ROS, Gazebo

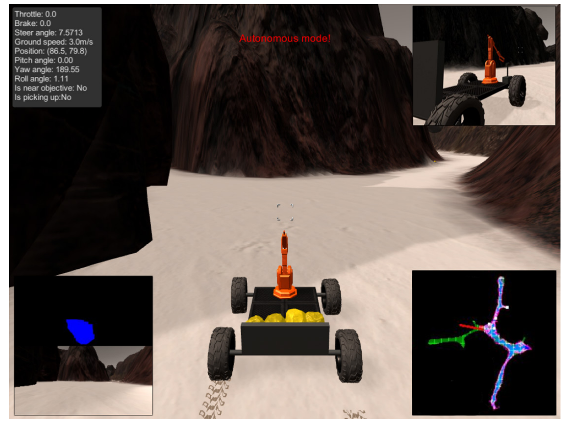
## AUTONOMOUS NAVIGATION OF NASA MARS ROVER ON UNITY

Aug, 2017 – Dec, 2017

**Project**
Search & Sample, Udacity Robotics Nanodegree Term 1

**Description**
Inspired by the NASA Sample Return Robot Competition, the project involves navigating through an unknown MARS terrain autonomously and picking up six rock samples scattered randomly.

Implemented the perception pipeline using OpenCV for exploring unknown regions of the map with over 98% fidelity, obstacle outlining, and detecting the rock samples. The navigation routine for the rover was implemented based on finite state machines and successfully collected all six samples. The environment was simulated in Unity Engine and provided by Udacity developers.



**Software**
Python (OpenCV)

## CYBER-PHYSICAL SYSTEM TESTBED FOR QUADCOPTERS

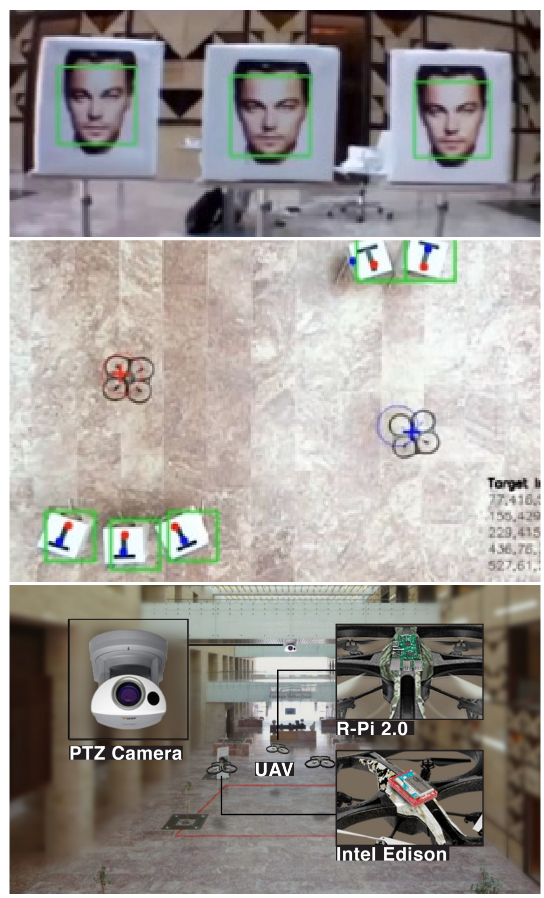
Nov, 2014 – May, 2016

**Project**
Drone-Be-Gone Testbed Research Project at Carnegie Mellon University at Qatar

**Description**
With the advent of low-cost drones with improved mobility and vision, they become extremely useful in inspection and surveillance operations.

Developed a low-cost, agile, and generic cyber-physical testbed for quadcopters. The quadcopter platforms utilized were Parrot AR.Drones 2.0 augmented with external processing units for off-loading vision and navigation algorithms. Two external processing units were used to run the algorithms – Intel Edison and Raspberry Pi.

With indoor 2-D localization using an overhead PTZ camera, and autonomous navigation for the quadcopters, coverage algorithms were tested and deployed for studying area and target coverage.




**Software**
Python (OpenCV),
C++, Node.JS)

**Hardware**
Parrot AR.Drone, Arduino,
Raspberry Pi, Intel Edison

1220 Mecaslin St NW Apt 1309
30318, Atlanta, GA

+1 (404) 797 1791

/in/mouhyemen

github.com/mouhyemen

mouhyemen.khan@gmail.com