Johnathon S. Li

Portfolio: john-s-li.github.io| **E-mail:** johnsjli@berkeley.edu | **Phone:** (+49) 15221795535

Education

University of California, Berkeley *Aug 2019 - May 2021*
Master of Science in Mechanical Engineering | Hybrid Robotics Laboratory | GPA: 3.9/4.0

Massachusetts Institute of Technology *Aug 2015 - May 2019*
Bachelor of Science in Mechanical Engineering |Gates Millennium Scholar | GPA: 4.9/5.0

Professional Experience

Software Engineer – ML + Motion Planning **Momenta** *February 2022 – Present*

- Bringing Momenta's L2+ product, Cruise Pilot, to Germany by developing new features requested by Momenta's European investors such as overtaking maneuvers compliant to German laws
- Spearheading the development of traffic light start stop control by concepting functional requirements, communicating with all relevant stakeholders, and writing new C++ features in Momenta's ADAS software stack
- Enabled machine learning and reinforcement learning pipelines for L4 autonomous driving in Germany by setting up necessary training infrastructure and adapting Python code developed by core China team
- Accelerated deployment of reinforcement learning models onto hardware by cutting down required expert training data threefold by utilizing transfer learning techniques
- Managed data mining and cleaning of all expert imitation learning data by developing rosbag processors and Python elastic search scripts on Momenta's closed-loop-automation platform

Software Engineer, Intern **Aptiv** *June – August 2019*

- Developed first proof of concept for development environment in ROS by programming software wrapper in C++ for AEB and effectively working with cross-continental controls and perception teams in Los Angeles, Sweden, and Germany
- Ensured viability of transition to development in ROS by writing verification and validation scripts in RViz and Python

Controls Engineer, Intern **General Motors** *June - August 2018*

- Enhanced performance of Chevy Bolt One-Pedal Driving on graded roads by improving existing software control architectures and implementing new control algorithms in Simulink
- Enabled usage of old IMU algorithms by debugging and parsing old IMU software, running system identification tests, and re-calibrating gains to match new vehicle dynamics of Chevy Bolt

Robotics Engineer, Intern **Medtronic** *June – August 2017*

- Created company's first proof-of-concept surgical robot remotely controlled by surgeon by prototyping relevant hardware using 3D printing and integrating relevant software modules with remote communication libraries in Simulink
- Streamlined software debugging process by integrating new CAD models of surgical tools into exiting simulations and by implementing engineering GUIs and unit tests for MATLAB functions

Research Assistant **MIT CSAIL** *May – August 2016*

- Implemented a location tracking visual feedback system for an origami robot by writing computer vision code in MATLAB that interfaced over serial data with an Arduino connected to SyRen Motor Drivers
- Enabled small origami robot to move on test platform with less resistance from magnetic coils by implementing control system to counteract non-linear behavior of coils when robot is near them

Academic Experience

MS Thesis: Quadrupedal Robot Control via Deep Reinforcement Learning *August 2020 – May 2021*

- Developed novel reinforcement learning method utilizing periodic reward composition functions to train a quadrupedal robot how to walk with various stable gaits
- Created high-fidelity quadruped simulator in PyBullet to facilitate the Proximal Policy Optimization agent training process
- Demonstrated that reward composition method is novel all-in-one training method to enable one policy to learn different walking gaits without change in network architecture

Autonomous Racecar (1:10 Scale) Controls Lead for MIT 6.141 *February – May 2019*

- Obtained path tracking with less than 0.3 meters of error by programming adaptive pure pursuit controller in ROS
- Localized our autonomous racecar to within 0.5 meters of error by developing motion model (with calibrated noise in speed, position and yaw) for Monte-Carlo particle filter
- Achieved robust traffic cone and lane following by implementing PID controller for distance and angle inputs along with color segmentation for cone identification and homography for cone distance estimation from robot in OpenCV

Skills

- **Programming:** Python (pytorch, numpy, scikit-learn, openCV, mmcv, conda); C++ (ceres, ROS, cmake, Eigen); MATLAB; Simulink; LabView; Arduino
- **Technologies:** Docker; Linux; AWS; MacOS; SolidWorks; Fusion 360; Rapid Prototyping (*3D Printing, CNC Mill and Lathe*)

Personal Projects

1. **C++ Library of Common Plath Planning Algorithms** *(https://github.com/john-s-li/cpp_path_planning)*

- A C++ library containing common path planning algorithms including Dijkstra's, A*, RRT, RRT*, and Hybrid A*

2. **C++ Library for 2D Vehicle Motion Planning using Reeds-Shepp Curves**
(https://github.com/john-s-li/reeds_shepp_cpp)

3. **Autonomous Racing via Model Predictive Control**

- Designed a linearized model predictive controller in frenet frame to generate both optimal path and optimal control inputs to go around a track as fast as possible while satisfying safety constraints