Dhruv Agrawal

Worcester, MA | +1(774)-578-4968 | dagrawal@wpi.edu | dhruv-agrawal-26mar | dhruv2603.github.io

Education

Worcester Polytechnic Institute — **August 2024 - May 2026**

Master of Science in Robotics Engineering; **GPA: 4.0/4.0** — *Worcester, MA*

Courses: **RBE502**-Robot Control, **RBE549**-Computer Vision, **RBE550**-Motion Planning

Visvesvaraya National Institute of Technology — **July 2018 - May 2022**

Bachelor of Technology in Electronics and Communication Engineering; **GPA: 8.7/10.0** — *Nagpur, India*

Courses: Digital Circuits and Microprocessors, Signal Processing, Machine Learning

Experience

Jio Platforms | *Firmware Engineer* | *Bengaluru, India* — **June 2022 – May 2024**

- Developed **state-of-charge** and **state-of-health** estimation algorithms using **model-based** and **data-driven** methods.
- Implemented the algorithms in simulation and on **edge** devices using **TensorFlow**, while collaborating within a team.
- Created custom code for **SREC** generation in order to flash the BMS onto the microcontroller through the bootloader.
- Collaborated on the design and development of firmware for an **EV charger system** using the OCPP protocol.

Publications

"Suntracker on Rocker-Bogie mechanism", *Advances in Mechanical Engineering: Select Proceedings of ICAME 2020, 719-726.*

- Engineered an **all-terrain robot** featuring a 3D-printed Rocker Bogie system stabilized with a **differential gear mechanism**.
- Mounted an adjustable solar panel on the robot and achieved **38.96%** improvement with respect to fixed mount panels.
- Achieved **15cm** step climb ability and **45°** gradient traversal success apart from smoothly maneuvering on uneven terrain.

"Improved Sign Language Recognition and Correction Using Inception Network, MediaPipe and PyEnchant", *2nd International Conference on Paradigm Shifts in Communications, Embedded Systems, Machine Learning and Signal Processing*.

- Utilized the **GoogLeNet V4** as the Neural Network for Sign Language Recognition leveraging **MediaPipe** framework.
- Integrated a **correction mechanism** using PyEnchant to improve the accuracy of the predictions by suggesting words.
- Achieved a training accuracy of **99.69%** in 47 epochs with a combined train and test dataset of 132000 images.

Projects

Polygon CBF based obstacle avoidance for cooperative manipulation of cable-suspended payload using quadrotors*

- Employ **CBF** as an obstacle avoidance method along with **non-linear MPC** for payload control in a multi-quadrotor system.
- Use **CasADi** as a problem formulation tool and **Acados** as a solver to generate the control input to the mav-payload system.
- Evaluate the effectiveness of the method while avoiding obstacles and having **safety distance constraints** on the robot.

Real Time Obstacle Avoidance & Path Planning with Kinodynamic Constraints | GitHub | Project Report

- Developed an **autonomous navigation system** for drones which have the ability to operate in unknown environments.
- The project specifically addresses the challenge of incorporating **kinodynamic constraints** in a drone's path planner.

Quadrotor Control Using LQR | GitHub | Project Report

- Implemented an **LQR controller** to track the trajectory of a quadrotor and compared it with a **PD controller** in simulation.
- The path time for the LQR controller is **15%** faster on an average than the PD controller, while having a lower distance error.

Robotic Arm Manipulation | GitHub | Project Report

- Programmed a robotic arm in ROS2 to **grasp and pick up** simple objects whose location is available to the robot.
- Implemented **velocity kinematics** for the robot to move at a constant velocity by providing incremental position values.
- Formulated a **PD controller** to control the position of the robot joints by controlling the input current to the joint actuator.

Out of Control Planning | GitHub | Project Report

- Constructed the path plan of a **pendulum** and a **non-holonomic car** system with dynamic motion constraints using RG-RRT.
- The average solution length of the RG-RRT is **half** of RRT and similar to KPIECE while having **least** node count in the tree.

Skills

- **Languages**: Python, C, C++
- **Frameworks**: PyTorch, OpenCV, OMPL, ROS2, PX4Autopilot, MATLAB, Docker, Git, OctoMap, MediaPipe
- **Tools:** RViz, NumPy, Acados, EDT3D, Arduino, S32Design Studio
- **Development Boards:** NXP's S32K144, Syntiant's Edge AI EVB, Raspberry Pi, ESP32, Arduino Uno