Aditya Varadaraj

Email : varadarajaditya@gmail.com, avaradar@umd.edu
Linkedin: https://www.linkedin.com/in/aditya-varadaraj-02272a147/
Mobile : +1-301-792-6815
Github: https://github.com/AdityaVaradaraj
3429 Tulane Drive, Apt. # 32, Hyatsville, MD - 20783

Education

- **University of Maryland - College Park** College Park, MD
  M.Eng. Robotics; GPA: 3.888/4.0 *Aug 2021 - May 2023*
  Courses: Applied Nonlinear Control of Aerospace Systems, Statistical Pattern Recognition, Rehabilitation Robotics, Perception, Planning and Control of Robots, Fundamentals of AI/DL, Robot Learning
- **National Institute of Technology Karnataka** Surathkal, Karnataka, India
  B.Tech. Mechanical Engineering; GPA: 8.13/10.0 *Jul 2016 - Apr 2020*

Skills

- **Programming Languages**: Python, MATLAB, C++, Arduino
- **Tools**: PyElastica, PyBullet, OpenAI Gym, OpenCV-Python, TensorFlow, PyTorch, Keras, ROS, Gazebo, Linux, Solidworks, Catia v5
- **Concepts**: Control Systems, Machine Learing, Deep Learning, Computer Vision, Planning

Experience

- **University of Maryland - College Park** College Park, MD
  Voluntary Work - Prof. Nikhil Chopra *Jun 2022 - Dec 2022*
  ○ **Modelling and Simulation of Soft Inverted Pendulum Robot**: **Researched** modelling methods for soft continuum robots including Cosserat Rod Theory, Piecewise Constant Curvature Approximation and FEM methods. **Developed** code to simulate: a) the soft robot inverted pendulum hinged at the start point using **Cosserat Rod Theory (PyElastica software)** b) series of 3 fixed end soft robot segments.
  ○ **Control of Soft Inverted Pendulum Robot**: **Implemented PD Joint-Space Control** for controlling curvature of fixed end 3-segment soft pneumatic robot in horizontal plane. *GitHub*
- **Bajaj Auto Ltd.** Akurdi, Pune, Maharashtra, India
  Robotics and Automation Intern *May 2018 - Jun 2018*
  ○ **Kinematics Simulation of UR5 Robot**: Wrote Python code to simulate a stick-model of a 6 dof UR-5 Collaborative Robot using Denavit-Hartenberg (DH) Parameters method of Forward Manipulator Kinematics. Developed code to estimate the joint angles given the end effector point and a rotation matrix corr. to that point, hence, allowing the user to give only robot's end effector position & orientation as input.

Academic Projects

- **Decision Making for Oncoming Traffic Overtaking Scenario using Double DQN (Deep Reinforcement Learning)**: Will **Implement and Train** a **Double DQN** (Deep Q-Learning Network) to **overtake cars** in a 2-lane 3-vehicle scenario on highways. (*Mar 2023 - May 2023*)
- **CNN-based 3D Super-Resolution of MRI Scans in IXI Dataset (Deep Learning)**: **Implemented and Trained** the **CNN-IL** algorithm in the existing IEEE journal paper on **IXI dataset**. **Proposed and Trained improved** custom networks for super-resolution in height-and-width and depth directions in **PyTorch** by **rewiring skip connections**, **adding ELU activations** and **adding more layers**. **Improved SSIM and PSNR using the above custom network.** *GitHub* (*Nov 2022 - Dec 2022*)
- **Adaptive Stiffness Impedance Control for AnkleBot (Control Systems)**: **Derived** the dynamics and **Adaptive Stiffness Impedance Control** algorithm for the **ankle rehabilitation robot** in the existing IEEE journal paper. **Implemented** the control algorithm in **Python using scipy odeint** to best reproduce the results in the journal paper. **Improved** the control algorithm by **modifying the Cost Function** and **adding Force-Feedback** *GitHub* (*Nov 2022 - Dec 2022*)
- **Monkey Species Classification (Deep Learning)**: **Implemented** a **custom 5-layer CNN** in **PyTorch** to classify Monkey species. Then used **Transfer Learning** by freezing weights of Convolutional layers of **ResNet-34** with fine-tuning using **ResNet-50**. *GitHub* (*Dec 2022*)
- **Handwritten Digit Recognition (Machine Learning)**: **Implemented Logistic Regression, Kernel SVM** with **PCA, MDA** and **LeNet-5 CNN** to solve the classification problem. *GitHub* (*Dec 2022*)
- **Face Recognition (Machine Learning)**: **Implemented Bayes Classifier, kNN, Kernel SVM and Boosted SVM** with **PCA, MDA** from scratch to solve the classification problem. *GitHub* (*Nov 2022*)
- **Clustering Algos (Machine Learning)**: **Implemented K-Means, Kernel PCA and Spectral Clustering** from scratch to solve the classification problem. *GitHub* (*Nov 2022*)
- **Robust Control of Bevel-Tipped Needle for Surgical Procedures (Control Systems)**: Derived the **SMC (Sliding Mode Control), Integral SMC (ISMC), Super Twisting Algorithm (STA)** controls algorithms and their finite-time convergence in the existing IEEE journal paper. Implemented the control algorithm in MATLAB to best reproduce the results in the journal paper. *GitHub* (*Apr 2022 - May 2022*)

- **Video Stabilization and 3D Human Pose Estimation (Computer Vision)**: Implemented Video Stabilization using VidStab and **3D Human Pose Estimation** with **OpenPose, Gesture Detection, and Depth estimation** with **hardware** implementation on Jetson Nano and Arduino based 4-wheeled mobile robot made by us. (*Apr 2022 - May 2022*)
- **Stereo Vision (Computer Vision)**: Implemented Calibration (SIFT, Essential/Fundamental Matrices), Rectification (Warping), SSD Correspondence and Depth Computation for given stereo vision image datasets in OpenCV Python. *GitHub* (*Apr 2022*)
- **LQR and LQG Control of Nonlinear Crane double pendulum system (Control Systems)**: Derived equations of motion and linearized the non-linear system using Jacobian linearization to obtain state-space equation. Checked **controllability** and implemented **LQR** control algorithm in MATLAB for the linearized and original nonlinear system. Checked **observability** of the system for various choice of output vectors and obtained best Luenberger Observer by pole placement. Plotted response of linearized and original nonlinear systems to initial conditions and unit step input for observable choices of output vector. Implemented **LQG** for linearized and original nonlinear system in MATLAB for smallest observable output vector. Plotted the initial response and unit step response of the systems. *GitHub* (*Nov 2021 - Dec 2021*)
- **Lane Detection/ Guidance (Computer Vision)**: Detected and provided turn guidance based on curvature, Sliding window approach, **warping, polynomial fitting**. *GitHub* (*Mar 2022 - Apr 2022*)
- **Holistic Adaptive Model Predictive Control of 4WID Autonomous vehicle (Control Systems)**: Derived and Understood the mechanics/dynamics equations, the controls algorithm and the cost function in the existing IEEE journal paper. Implemented the control algorithm in **MATLAB** to best reproduce the results in the paper. **Tuned some weight and constraint parameters** to make the results match the ones given in paper. *GitHub* (*Oct 2021 - Nov 2021*)
- **Path Planning**: Implemented TG-RRT*, RRT*, Dijkstra, A*, BFS, DFS algorithms