Kartik Virmani

4525 Walnut St., Apt. 213, PA,19139. | kvirmani@seas.upenn.edu | (215) 452-8308| LinkedIn • ResearchGate

EDUCATION

University of Pennsylvania, School of Engineering and Applied Science, **Philadelphia, PA** *May 2026*
Master of Science in Mechanical Engineering & Applied Mechanics: Conc. Mechatronics & Robotics.

VIT-Vellore Institute of Technology, School of Mechanical Engineering, **Vellore, TN, India** *May 2022*
Bachelor of Technology: Mechanical Engineering. **GPA: 9.73/10.00**

SKILLS

Programming & Development: C++, Python (JAX, TensorFlow, PyTorch), ROS2, Reinforcement Learning.
Robotics & Systems: Forward/Inverse Kinematics, Inverse/Forward Dynamics, Task Planning Frameworks, FSM, Behavior Trees.
Control: Path planning algorithms, Model Predictive Control (MPC), PID controllers, motor control systems, trajectory optimization.
Interpersonal Skills: Cross-functional collaboration, adaptability, project coordination, proactive communication, problem-solving.

PROFESSIONAL EXPERIENCE

Modular Robotics Lab (MODLAB) University of Pennsylvania, *Research Assistant,* **Philadelphia, PA.** ***May 2025 – Present***

- Engineered passively stable mobile humanoid robots by lowering the center of mass, connecting torso to base with spherical joints.
- Developed ROS Gazebo simulation modeling nonlinear pendulum dynamics of torso on holonomic mobile base with spherical joint.
- Implemented nonlinear control algorithms, including energy-shaping and feedback linearization, to stabilize swinging torso.
- Prototyped mechanical parts, integrated IMU and encoders, programmed embedded microcontrollers for real-time control loops.
- Tuned sensor fusion and control parameters in simulation and hardware-in-the-loop tests to optimize balance on uneven terrains.

X-Lab University of Pennsylvania, *Research Assistant,* **Philadelphia, PA.** ***February 2025 – Present***

- Developed ROS 2 motion control pipelines in C++/Python for F1-tenth, integrating MoveIt 2 and OMPL planning.
- Implemented 3D trajectory generation using system dynamics and solver optimization for uneven terrain motion tasks.
- Used 3D LiDAR to estimate robot's vertical (Z-axis) position, enabling precise control on rough surfaces.
- Formulated control equations and integrated reverse thrusters on mobile robots to test vertical motion stabilization.
- Debugged and deployed ROS2-Rviz stacks supporting vision-conditioned path planning and real-time perception-to-action policies.

Engineers India Limited, *Mechanical Engineer,* **Barmer/Delhi, India.** ***July 2022 – July 2024***

- Designed 2D/3D CAD assemblies for pressure vessels and rotary components using *SolidWorks* and *CATIA*.
- Conducted structural analysis using *ANSYS* FEA and *GD&T*, reducing installation defects and improving component reliability.
- Led *DFM/DFA* reviews and managed technical documentation (*BOMs*, *P&IDs*) for smooth manufacturing and integration.
- Diagnosed and resolved vibration-related failures through *Python*-based root-cause analysis, increasing component lifespan by 50%.

Centre of Excellence for Autonomous Vehicles Research, *Research Assistant,* **VIT-Vellore, India.** ***July 2021 – December 2021***

- Developed embedded control firmware in C++ for electric power steering and braking using Atmega and ESP32-based hardware.
- Built testbeds integrating encoders, FSRs, and vision sensors; conducted dynamic system tests and logged performance data in Python.
- Modeled and iterated on mechanical fixtures in PTC Creo for closed-loop testing, ensuring alignment of sensors and actuators.
- Prototyped electromechanical subsystems using rapid assembly methods (3D printing, laser-cutting) and performed in-lab debugging.
- Supported hands-on electronics work including PCB soldering, breadboarding, and logic-level debugging of sensor circuits.

PROJECTS

F1/10th Autonomous Vehicle Racing. ***January 2025 – Present***

- Developed ROS 2-based motion planning in C++ and Python, integrating navigation, perception, and control for racing.
- Implemented RRT, Pure Pursuit, and MPC for real-time trajectory planning and control in high-speed racing scenarios.
- Optimized vehicle state estimation using sensor fusion (LiDAR, IMU, GNSS) for precise localization and mapping.
- Designed and tested basic electric motor control circuits, improving efficiency and enhancing motion control performance.
- Implemented real-time simulation in Rviz, testing racing strategies and tuning controllers in a physics-accurate environment.
- Used NVIDIA Jetson for onboard processing, leveraging CUDA for high-speed sensor data processing and decision-making.
- Configured CI/CD pipelines with GitHub Actions, automating software testing, container builds, and deployment.
- Tuned PID and MPC controllers for high-speed race stability, minimizing lap time variations and improving trajectory tracking.

Quadrotor Control, Dynamics, and Vision-Based Localization ***April 2025 – May 2025***

- Developed and validated 3D path planning and trajectory optimization for quadrotor systems in a flight simulator.
- Tuned PID and feedback controllers, improving quadrotor stability and trajectory tracking by 20%.
- Implemented spatial vector arithmetic and complementary filters, enhancing localization and flight accuracy by 8%.
- Performed post-simulation and real-world testing, ensuring reliable quadrotor performance.

Grounded SLAM: Dynamic Object Filtering in ORB-SLAM3 Using GroundingDINO ***April 2025 – May 2025***

- Integrated GroundingDINO, SAM, Kalman Filters with ORB-SLAM3 for dynamic object filtering in RGB-D SLAM pipeline.
- Built ROS-based dual-Docker system bridging C++ SLAM and Python vision stack for real-time perception and mapping in Linux.
- Reduced localization error (ATE ↓29.5%, RPE ↓25%) and improved map density by 81% using mask-based filtering.
- Designed Kalman Filter to stabilize dynamic object tracking, improving SLAM consistency under non-static environments.
- Ported and deployed ORB-SLAM3 on ROS Noetic; fixed C++17, OpenCV, and ROS API compatibility issues.

Learning Vehicle States using Unscented Kalman Filters (UKF) ***January 2025 – Present***

- Developed a UKF-based state estimation framework for localization, tracking position, velocity, yaw angle, and yaw rate.
- Implemented sensor fusion using IMU, GPS, LIDAR, and wheel encoders, reducing localization drift by 30% over 1 km.
- Generated synthetic bicycle model vehicle dynamics data using Carsim to validate and train the algorithm.
- Modeled vehicle dynamics using a bicycle model, improving motion prediction accuracy by 25% in unstructured environments.

Learning based physics informed neural network model predictive control for autonomous vehicles. ***October 2024 – December 2024***

- Implemented Model Predictive Control (MPC) with neural networks, optimizing motion planning and trajectory tracking.
- Accelerated GPU-based reinforcement learning training, reducing motion planning convergence time by 30% using CUDA.
- Developed a Gaussian process model for motion planning algorithms, integrating reinforcement learning and AI.
- Used CarSim and MATLAB for simulating robot manipulators, generating train-test data for task planning.
- Logged high-frequency state transitions in Simulink, refining AI-based task planning strategies.

MPC-Controlled Quadruped with Custom Joint Design ***October 2024 – December 2024***

- Designed and assembled testable robotic leg joints in CAD, simulating load conditions to ensure structural robustness.
- Developed impedance control strategies using URDF-based rigid-body trees for terrain adaptability and force regulation.
- Ran motion studies to tune joint placement and actuator stroke, ensuring mechanical range and ground contact reliability.
- Validated controller performance through physics rollouts and iterated leg design to improve gait cycle repeatability.

Autonomous Mobile Robot– Penn-GRASP MEAM5100 Autonomous Robot Competition. ***October 2024 – December 2024***

- Designed and prototyped a modular autonomous robot using *Fusion 360*, laser cutting, and 3D printing within project constraints.
- Integrated *ESP32 microcontrollers*, TOF sensors, and H-Bridge motor drivers for reliable autonomous system control.
- Developed closed-loop motor control in *C++* using encoder feedback with PID algorithms for smooth, precise motion control.
- Built and soldered custom circuits, for VIVE-based localization and *I2C*, *SPI*, *UART* protocols for sensor and hardware communication.
- Implemented *WiFi-based communication* between distributed microcontrollers, modularizing sensing and control subsystems.
- Managed hardware assembly, system integration, part sourcing, and deadlines to deliver a fully functional robot within project requirements.