# Sayantani Bhattacharya

(773) 273-0625 | Evanston, IL

Portfolio: sayantani-bhattacharya.github.io | Github: github.com/Sayantani-Bhattacharya | email: sayantani@u.northwestern.edu

## EDUCATION

| | | |
|---|---|---|
| Northwestern University | Master of Science in Robotics | 2024-2025 *(expected)* |
| Indian Institute of Technology, ISM Dhanbad | Bachelor of Technology in ME | 2018-2022 |

## SKILLS

- **Robotics:** ROS 2, SLAM, Perception, DSP, Navigation, Manipulation, Motion Planning & Control, Path planning, Optimization.
- **Machine Learning:** PyTorch, Supervised, Unsupervised, Neural-Network, Deep Reinforcement Learning & Multi Agent RL.
- **Programming:** C++, C, Python, Graph Theory, Concurrency, gdb, TDD, Linux, Git, Behavior tree, Docker.
- **Libraries:** sympy, OpenCV, LP solver, PyQt, Nav2, RTABMap, g2o, MoveIt, Slam-Toolbox, Unitree GOs & Camera SDKs.
- **Software Tools:** Weights & Biases, Isaac Sim, Unity Gym, Coppelia-Sim (V-Rep), AMPL, MATLAB, LABVIEW, MongoDB.

## WORK EXPERIENCE

**SeNSE Lab — Graduate Robotics Researcher (Full Time)** | Evanston, IL, USA *Apr' 25* - Present

- Developing Bio-Inspired perception & control loop for Underwater ROV using whisker sensing and ArduSub on Pixhawk.
- Implemented 6D pose tracking of whisker ends and joystick teleop of ROV over MavLINK using BlueOS Interface and ROS.
- Analysed & iteratively refined raw sensor data to obtain robust high-fidelity inputs for perception modelling.
- Researching onboard autonomous decision-making models to detect and track underwater prey with whisker sensing alone.

**Laki Robotics — Founding Robotics Engineer (Contract)** | Hawaii, USA (Remote) *Aug' 25* - Present

- Leading the technical team to develop MVP for multi-quadruped mapping & collective intelligence system for surveillance.

**Addverb — Robotics Engineer (Full Time)** | Delhi, India *Aug' 22 - Jun' 24*

- Developed Behavior tree, Task-Scheduler, Assignment and TCP interface of warehouse material handling robot fleet in C++.
- Formulated heuristic, graph-based & linear programming based solutions for multi-agent path-finding problems of AMRs.
- Developed collision avoidance and priority based task assignment for multi-carton picking robot fleet using *C++ & lp solver.*
- Developed modules for physics emulator of the real fleet system i.e. Robot Motion, State Machine & WCS in C++.
- Built fleet operation simulation GUI for Sales Team, cutting their throughput calculation time from 2 days to 3 hrs.
- Published in IEEE Conference on Decision and Control (CDC) 2023 for Lifelong Conflict Free Node to Robot Scheduling.
- Responsible for client demos, remote site support & throughput optimization for fleets of upto 160 AGVs /AMRs.

## PROJECTS

**Collaborative Exploration and Mapping by Quadruped Fleet** *(Visual SLAM, LIDAR SLAM, Navigation, ROS 2, C++, Unitree Go1 & GO2)*

- Developed SLAM pipeline using ZED SDK & Visual-Inertial data for Unitree GO1 and RTABMap & LiDAR data for GO2.
- Developed Autonomous Navigation ROS packages using frontier exploration and high level control for GO2 SDK.
- Applied deskewing, voxel filtering, and point cloud clustering techniques to improve occupancy grid accuracy.

**Franka Arm playing Mini Golf** *(ROS 2, Motion Planning, MoveIt, Computer Vision, Python)*

- Implemented Cartesian path planning trajectories for Franka using MoveGroup API to strike a golf ball towards the hole.
- Developed ROS 2 node to synchronize ball & target detection from vision pipeline with real-time trajectory generation.

**Feedback Control of KUKA Omnidirectional Mobile Manipulator** *(V-REP, Motion Control )*

- Generated trajectory for a 5-DOF KUKA robot arm to perform pick-and-place tasks in CoppeliaSim.
- Simulated kinematics of the omnidirectional robot with odometry equations to determine its next configuration.
- Implemented and tuned a feed forward PI controller to minimize error between the current and desired robot states.

**Multi-Agent Reinforcement Learning in Table-Tennis** *(MARL, Unity Gym, PyTorch)*

- Designed Deep-Deterministic-Policy-Gradient agents to collaborate for max game time, while competing to win.

**Pen Grasping Robot** *(OpenCV, Computer Vision , Manipulation, PyTorch, Camera-Caliberation, Python)*

- PincherX100 Arm programmed to detect, calibrate, manipulate and grasp a pen using a RealSense depth camera.

**F1 Ghost Riding using Visual Pose Graph Estimation from Scratch** *(C++, Computer Vision, SLAM, ROS2, Concurrency, OpenCV, g2o)*

- Extracting RGBD features via ORB and SIFT techniques to construct optimized relative pose-graph using PnP RANSAC.

**Bee Dance Swarm Intelligence** *(MARL, PyGame simulation, PyTorch, Stable Baselines3, Python)*

- Trained PPO-based multi agent reinforcement learning model, to emulate realistic bee foraging behaviour.
- Engineered a custom simulator for differential drive robots with observation, reward, and action dynamics for each agent.