

Sayantani Bhattacharya

(773) 273-0625 | IL, Chicago

Portfolio: sayantani-bhattacharya.github.io/ | **Github:** github.com/Sayantani-Bhattacharya | **Mail:** 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, SLAM, Perception, Vision, Motion Planning & Control, Signal Processing, Path planning, Optimization.
- **Machine Learning:** Regression, Clustering, Classifying, Deep Reinforcement Learning, Multi Agent Reinforcement Learning.
- **Programming:** C++, C, Python, Graph Theory, Concurrency, gdb, TDD, Linux, Git, Behavior tree, Docker, rabbitmq, MongoDB.
- **Libraries:** PyQt, PyTorch, sympy, numpy, OpenCV, matplotlib, LP solver, Nav2, RTABMap, MoveIt, Slam-Toolbox, Unitree GOs.
- **Software Tools:** MATLAB, Simulink, Coppelia-Sim (V-Rep), AutoCAD, LABVIEW, Unity Gym Env, Weights & Biases, AMPL.

WORK EXPERIENCES

Addverb — Robotics Engineer

Aug' 22 - Jun' 24

- Formulated heuristic, graph-based & linear programming based solutions for multi-agent path-finding problems.
- Developed Behavior tree, Task-Scheduler, Assignment and TCP interface of warehouse material handling robot fleet in C++.
- Developed collision avoidance and priority based task assignment for multi-carton picking robot fleet using C++ & lp solver.
- Built the complete fleet operation simulation configurator tool, for the Sales team, using PyQt & Networkx library that reduced the throughput calculation time from 2 days to three hours.
- Developed modules for the physics emulator of the real fleet system i.e. Robot Motion & Warehouse Control System in C++.
- Published in IEEE Conference on Decision and Control (CDC) 2023 - [Link](#) for Conflict Free Node to Robot Scheduling Algo.
- Responsible for client demos and remote site support.

IIT Delhi & IIT Dhanbad — Robotics Control Intern

July 21' - Apr'22

- Designed closed-loop force control algorithms for serial and parallel robot manipulators using Sim-Mechanics.
- Implemented forward & inverse dynamics of robot manipulators using Euler-Lagrangian equations and designed GUI.

Uit-The Arctic University of Norway — Perception Intern

Jan 21' - July 21'

- Implemented and experimented with Fourier and Wavelet Transform for Denoising acoustic microscope's time-series data.
- Devised a novel method to differentiate structural damages up to 100 μm difference. (Labview, MATLAB, Python)
- Implemented multiple filters for image processing for damage detection of diseased biological samples.
- Published in MDPI sensors. - [Journal Link](#) || Presented in Symposium of Ultrasonic Electric Japan - [Proceeding Link](#)

PROJECTS

Multi-Agent Collaborative Exploration by Quadruped Fleet (SLAM, 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 packages using frontier exploration and high level control package for GO2 SDK.
- Applied deskewing, voxel filtering, and point cloud clustering techniques to improve occupancy grid accuracy.

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

-Ongoing

- Trained PPO-based multi agent reinforcement learning model, to emulate the realistic bee foraging behaviour.
- Engineered a differential drive robot environment with custom observation, reward, and action dynamics for each agent.
- The main goal is to have emergent interactions and specialisations in each bee for optimized collective nectar collection.

Bio-inspired Underwater ROV (Perception, ROS 2, C++)

-Ongoing

- Developing a system for rovers to follow a prey underwater, using the sensory information of a seal whisker-like structure.

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.
- Implemented MADDPG with experience replay and exploration noise in a Unity-based continuous control environment.

Pen Grasping Robot (OpenCV, PyTorch)

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

Linux System Monitor (C++, Linux)

- A system monitor system parallel to htop, that displays the CPU, process IDs, memory usage, etc of a Linux system.