

Sayantani Bhattacharya

(773) 273-0625 | Chicago, 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 (<i>expected</i>)
Indian Institute of Technology, ISM Dhanbad	Bachelor of Technology in ME	2018-2022

SKILLS

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

WORK EXPERIENCES

Addverb — Robotics Engineer

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.
- Developed collision avoidance and priority based task assignment for multi-carton picking robot fleet using C++ & *lp solver*.
- Built the complete fleet operation simulation tool for Sales, cutting throughput calculation time from 2 days to 3 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 for Lifelong Conflict Free Node to Robot Scheduling.
- 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'

- Devised a novel method to differentiate 100µm damage using Fourier & Wavelet transform on acoustic microscope signals.
- Implemented multiple filters for damage detection of diseased biological image samples using *Labview, MATLAB and Python*.

PROJECTS

Multi-Agent Collaborative Exploration by Quadruped Fleet (*SLAM, ROS2, C++, Perception, 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.

Franka Arm playing Mini Golf (*ROS2, Motion Planning, MoveIt, Python*)

- Implemented Cartesian path planning trajectories for Franka using MoveGroup API to strike a golf ball towards the hole.
- Developed ROS2 node to synchronize the ball & target detection from vision pipeline with real-time trajectory generation.

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

-Ongoing

- Trained PPO-based multi agent reinforcement learning model, to emulate the realistic bee foraging behaviour.
- Engineered a custom simulator for differential drive robots with 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 Perception for Underwater ROV (*Perception, Computer Vision, ROS2, C++*)

-Ongoing

- Developing a BlueOS-based system to follow a prey underwater, using sensory information of a seal whisker-like structure.
- Implementing 3D Pose tracking of whisker end arrays and joystick control of ROV over MavLINK in ROS2.

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.

Visual Pose Graph Estimation from Scratch (*C++, Computer Vision, SLAM, ROS2*)

-Ongoing

- Extracting RGBD features, estimating relative pose using feature matching and motion estimation to construct pose-graph

Pen Grasping Robot (*OpenCV, PyTorch*)

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