Assignment 2 (RI Monsoon 2025)

6 Nov 2025 - 13 Nov 2025

[20 + 20 + 30 + 30 = 100 Marks]

Bio & Neural-Inspired Architectures — Human–Robot Collaboration — AND/OR Graphs — Planning with Discrete & Continuous Quantities — PDDL+ & Hybrid Planning — ROSPlan Basics — Swarm Cognition

Instructions

- Read all questions (Q1–Q4) carefully before attempting.
- Provide answers in structured format with diagrams where required.
- Maintain clarity and brevity (avoid unnecessary length).
- Submit written content as a single compiled PDF. [Report.pdf]
- Include everything in a single zip file. [RollNo_Name_Assignment2.zip]
- Download code for Q4 from here: Sample Code.zip

Q1. Dynamic Control and Action System (DCAS): From Perception to Action (20 pts)

Goal: Apply the principles of Dynamic Control and Action Systems to analyze how perception and motor dynamics jointly give rise to intelligent behavior in robots.

Choose ONE scenario:

- A mobile robot that must balance a pole while moving toward a goal.
- A robotic arm that must pour water without spilling.
- A drone that must hover stably while tracking a moving target.

System Decomposition (1/2 page)

Describe the system as a DCAS:

- Identify perceptual variables (what is sensed continuously).
- Identify action variables (what can be controlled continuously).
- Draw or describe qualitatively the feedback couplings between them (avoid using control equations; focus on structure).

Dynamic Attractor Analysis (1/2 page)

For one selected variable (e.g., pole angle, liquid level, target distance):

- Describe what kind of attractor or stability behavior it should have (point, limit cycle, metastable).
- Explain how this attractor supports goal-directed action without discrete planning.

Perturbation Thought Experiment (1/2 page)

Imagine a small perturbation (e.g., sensor delay, noisy actuation, visual occlusion):

- Predict qualitatively how the system's dynamics would change.
- Explain whether and how the action system would recover or adapt.

DCAS vs Symbolic Reasoning (1/2 page)

Contrast your DCAS model with a symbolic planning model of the same task:

- What does the symbolic planner capture that DCAS does not?
- What does DCAS capture that symbolic planning misses?

Deliverables: A 2-page written response (with one system diagram).

Q2. Swarm Formation and Emergent Behavior (20 pts)

Goal: Model emergent swarm trail-formation via local pheromone rules.
Tasks:

1. Implement a simple swarm simulation (any language) with 10 robots following pheromone trails.
2. Introduce one perturbation midway (e.g., pheromone decay doubled, or sensor noise injected).
3. Plot collective performance over time — and explain in words why the swarm still stabilizes or collapses.
4. End with a one-paragraph comparison: "How does this swarm exhibit collective cognition even though no robot plans?"

Deliverables: Code + 2-page report with figures.

Q3. AO* Search in AND/OR Graphs (30 pts)

Goal: Implement AO* to find an optimal solution in a task-decomposition graph.
Tasks:

1. Write `ao_star.py` from scratch (no imported planners).
2. It must print the evolving partial solution graph after each iteration.
3. Annotate one non-leaf node explaining why expansion stopped there.

Reasoning Check: Submit a short (1/2 page) manual run-through on a 4-node example showing your intermediate cost estimates.

Reflection (1/2 page): Explain why AO* behaves differently from A* and BFS in your case graph, using a conceptual rather than numerical argument (e.g., how subgoal dependencies change heuristic admissibility).

Q4. Hybrid Planning with MoveIt and ROSPlan (30 pts)

Goal: Understand interleaved symbolic + motion planning (TAMP).

Experiment Design: Define one symbolic goal that is guaranteed to fail in a purely discrete planner (e.g., due to geometry) and show how the interleaved setup resolves it.

Files to Modify:

- `pddl/hybrid_problem.pddl`: Change start/goal or add new objects.
- `monitor_motion.py`: Replace random feasibility with MoveIt service call.
- `replanner.py`: Adjust replan trigger logic or thresholds.

Deliverables:

- Modified `domain.pddl` / `problem.pddl`
- Short `REPORT.md` including:
  - Number of replans observed.
  - Explanation of one infeasibility event.
  - Table of metrics (success/time/replans).
  - Reflection: When would symbolic replanning alone fail?

Analysis (1 page):

- Compare sequential vs interleaved planning: success rate, replans, and execution time.

- Then answer: If your robot had perfect motion feasibility checks, would symbolic planning still need refinement? Justify conceptually.

This starter code is intentionally minimal. Students are expected to:

- Integrate MoveIt collision checks (via /check_state_validity service).
- Experiment with continuous fluents (battery, distance, time).
- Run both sequential and interleaved modes for comparison.

Deliverables: Zip folder with domain/problem files, launch files, and REPORT.md. The report must include a 4-line summary of what surprised you about your results.