RI

Assignment 2

Q1. Scenario - A robotic arm that must pour water without spilling.

System Decomposition

Some of the perceptual variables can be -

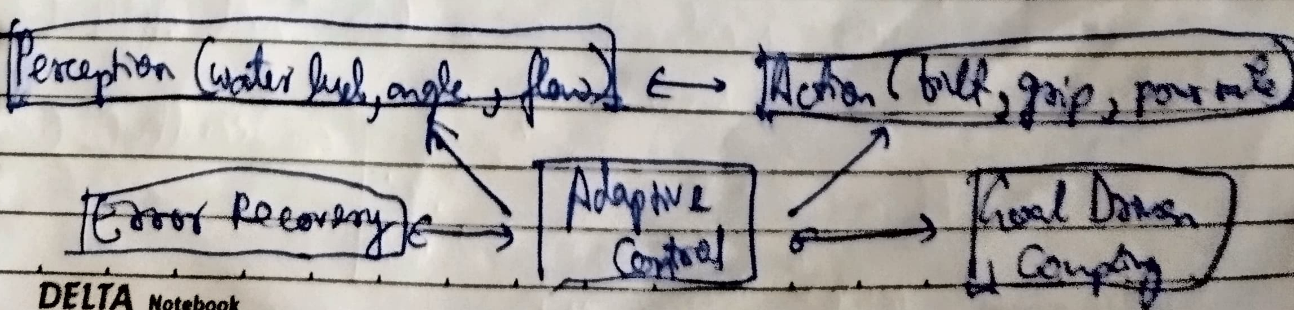
- The water level of the container
- Tilting angle of the cup
- The rate of pouring water (can be observed by the stream width in the vision)
- Distance & relative position between the pouring vessel and the target container.
- The gripping force on the vessel, to avoid dropping

Some of the action variables can be -

- The trajectory of the path & motion
- The end effector position & orientation of the arm
- Precise & variable control of the tilt speed & maximum angle.
- Rapid adjustments to abort pouring if spillage is detected
- Joint torque & ~~___~~ velocity commands for wrist.

Qualitative feedback couplings

- The continuous perception-action loop → The robot adjust its arm b/w the target & itself based on the sensory input. ex- if the spillage is detected, the robot will ~~freeze~~ freeze its motion.
- Adaptive Modulation → the speed & angle of pouring are adaptively regulated based on real-time trust & feedback from force & vision sensors



Date ____________
Page No. ____________

Dynamic Attractor Analysis

- For the variable liquid level in a robotic pouring task, a point attractor is ideal for a stable target water level that the system approaches and holds. This attractor stabilizes the continuous feedback loop as sensing confirms that the liquid is about to fill, so the pouring slows and stops naturally at the set point.
- This attractor supports goal-directed action as the actions closely track the sensed fill level and the trajectory of robot (tilt, speed, position) is continuously attracted to the desired state, so overfilling is prevented simply by intrinsic dynamics. This ~~is~~ enables adaptive pouring purely via real time coupling without the need of discrete planning.

Perturbation thought experiment

- If a small perturbation like a short sensor delay happens while measuring the liquid level, the system's dynamics would change like the robot receives slightly outdated information about the liquid level or maybe instead of approaching the target to fill smoothly, the liquid level may overshoot. for ex- the robot might start reduce pouring only after the cup is already full.
- The robot's control system is designed for continuous, dynamic correction. If it detects that its action didn't produce the expected ~~change~~ change (eg- the liquid level is too high after a tilt), it will adjust its motion & pouring rate in real time. even if some delay exists, minor errors won't persist.

DCAS vs Symbolic Reasoning

Symbolic planning models tasks like pouring by representing discrete states & action using logical symbols (like "pour", "stop", "cup full" etc) planning out explicit step by step sequences and leveraging rules for preconditions & effects.

- What Symbolic Planning captures & DCAS not?

→ Symbolic planners make it easy to verify & explain why each action is taken, check for logical consistency and debug plans. It can also reason about non-continuous conditions, global goals & dependencies (eg- don't pour if cup is already full).

- What does DCAS capture that symbolic planning misses?

→ DCAS models the pouring as a dynamic feedback process sensing liquid level and adjusting tilt where actions as a system is attracted towards the goal (ie- correct liquid level) without discrete step. Also it captures adaptive and micro-adjustments where symbolic logic struggles with representation and discretization.

Date
Page No.

Q2. • Why the swarm stabilizes or collapses?
Before the perturbation (normal pheromone decay), each robot explores randomly. When a robot finds a food source, it returns to the nest ~~with~~ while depositing pherone along its paths. Then other nearby robot senses that pheromone trail and move along it, reinforcing the same path when they also return.
This positive feedback loop strengthens the trail between the nest and that food source. Eventually, the swarm will collectively forms a stable trail to the most rewarding or accessible food sources.
This stable behavior happens without any central coordination purely due to the local interaction.

When the perturbation occurs at the halfway point the pheromone ~~[illegible]~~ evaporation doubles meaning that the pheromone trail fades faster ~~[illegible]~~ in each steps.

- So the trails that were clear start to weaken because fewer robot are reinforcing them in time.
- Some robots lose track of pheromone cues & wander randomly again.
- The swarm then becomes less ~~[illegible]~~ coordinated and collapses.

Date
Page No.

∴ It stabilizes because continuous local feedback (deposit + following) compensates for global decay. The system self-organizes around new steady-state pheromone concentrations.
and
It collapses because the positive feedback loop is broken. The pheromone information decays faster that it can be regenerated and coordination disappears.

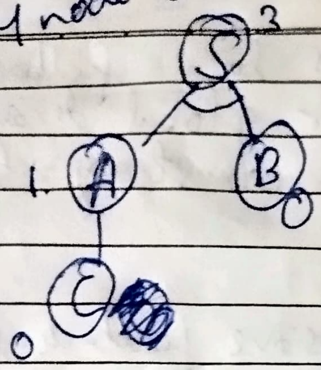
- How does this swarm exhibit collective cognition even though no robot plans?

The swarm exhibits collective cognition through the emergence of organized, goal-directed behaviours arising purely from simple local interactions rather than centralized planning.
Each robot follow minimal rules ie- exploring randomly, following stronger pheromone ~~trai~~ trails and depositing pheromones when returning with food. Yet, together they form efficient and stable trails between the nest and multiple food sources. Even when conditions change, such as increased pheromone evaporation, the swarm adapts collectively, rebuilding trails and ~~re~~ reallocating efforts toward viable food sources. Thus, the group as a whole demonstrates problem-solving intelligence despite each robot having no awareness of the overall task. This emergent coordination mirrors natural systems like ant colonies showing how complex cognition can arise from many simple, unplanned actions.

4 node example (Reasoning Check)

Date
Page No.

Q3.



each edge has weight 1

S → A+B

A → C

C & B are terminal nodes

$f(A) = f(C) + 1 = 1$

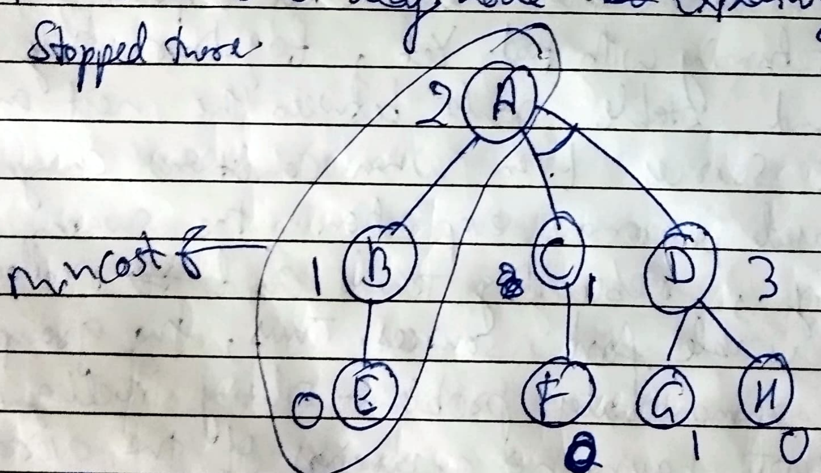
$f(S) = f(A) + 1 + f(B) + 1 = 1 + 1 + 0 + 1 = 3$

C is solved → ∴ A is also solved.

B and A are solved → ∴ S is also solved.

∴ total cost = 3 (S → ((A → C) + B))

- Annotate one non-leaf node explaining why expansion stopped there.



In this example, we won't expand the C node because we already got the minimum cost from A → B → E i.e 2. And since in AND we expand the larger cost node first (here D) and it is already larger than 2 so we will not expand the C further.

• Reflection

AO* behaves differently from A* & BFS because it operates on an AND/OR search space, not a simple path graph.

- A* assumes a singe goal path problem ie every node represents one partial path to a single goal. It expands nodes based on their total cost $(g + h)$ & finds a single least-cost path.
- BFS explores all possible paths layer by layer, without cost or heuristic guidance.

In contrast, AO* must handle subgoal dependencies:
AO* differs conceptually because it searches in a hierarchical problem decomposition space, where nodes represent tasks rather than states and costs are aggregated across inter-dependent subgoals.
This make its heuristic admissibility depend on how accurately it captures subgoal interactions, not just distance to a single target.

Date
Page No.

Q4. Analysis

Sequential planning ⇒ Success rate - fail
time taken - 5 seconds (until failure)
replans - 0
final plan - pick obj

Interleaved planning ⇒ Success rate - passed
time take - 12.5 seconds (plan + Check + replan + execute)
replans - 1
final plan - move, pick, place

If the robot had perfect motion feasibility checks, then Symbolic planning still needs refinement. Having perfect checks is different from having a perfect symbolic model. This symbolic model ie- PDDL is an abstraction of the continuous world.

PDDL model cannot and should not contain the precise coordinates of every object & all the robot's links. The Symbolic model says for example that the path is clear which is clearly its belief. The perfect check ie- MoveIt queries the ground truth of the geometric world & discovers that this belief was wrong.

∴ The symbolic planner is blind & deaf. It can generate plans and the perfect check can tell it whether the belief is correct or wrong.