Lam Research Challenge 2025

Problem Statements (PS) - CSE & Allied Branches

PS1: 100 Marks

PS1: Simulate a **Line Follower Robot in CoppeliaSim**, tailored to replicate a practical use case within a **semiconductor manufacturing environment**. While the base components (such as motors, wheels, and sensors) may be common across teams, your **robot design** and **arena layout** must be **uniquely conceptualized and implemented** to highlight your team's engineering creativity and problem-solving approach.

You are encouraged to collaborate closely with your **mechanical partner** to design the robot in a solid modelling software (e.g., SolidWorks, Fusion 360, FreeCAD) and import the model into **CoppeliaSim** for simulation.

The **arena** should be simple but industrially relevant—mirroring real-world manufacturing pathways such as those found in **Amazon warehouses** or **cleanroom semiconductor material transfer lines**. The simulation must clearly demonstrate the robot’s functionality, path accuracy, adaptability, and real-world applicability.

<table>
<tr><th>Week Number</th><th>Tasks Number</th><th>Remarks</th><th>Submissions</th><th>Date</th></tr>
<tr><td rowspan="3">W1</td><td rowspan="3">T1-1</td><td>1.Setup & Initial Implementation of CoppeliaSim</td><td rowspan="4">1.Video
2. PPT
3. Unique document</td><td rowspan="4">19-07-2025</td></tr>
<tr><td>2. Robot Design</td></tr>
<tr><td>3. Design of the Arena</td></tr>
<tr><td>W2</td><td>T1-2</td><td>Programming, Testing & Refinement</td></tr>
<tr><td colspan="5">The task descriptions and evaluation guidelines are provided in the following sections.</td></tr>
</table>

What is CoppeliaSim?

CoppeliaSim is a powerful and versatile robot simulation software developed by Coppelia Robotics. It provides a comprehensive platform to design, simulate, and test robots or automated systems in a virtual 3D environment. It is widely used in academia, research, and industry for prototyping, validating algorithms, and teaching robotics concepts.

CoppeliaSim is known for its:

- Real-time physics simulation
- Multibody dynamics
- Support for various sensors and actuators
- Extensive plugin and scripting interfaces
- Compatibility with many programming languages (Python, Lua, C++, Java, etc.)

Key Features

- 3D Simulation Environment: Create and manipulate robotic systems in a realistic 3D world.
- Integrated Physics Engines: ODE, Bullet, Vortex, and Newton for realistic motion and interactions.
- Built-in Models: Predefined robots, sensors, and actuators ready to use.
- Custom Robot Design: Build robots using primitives or import from CAD tools like SolidWorks.
- Remote API Access: Control simulations via Python, MATLAB, or ROS (Robot Operating System).
- Real-time Control and Feedback: Interact with sensors, actuators, and environments in real time.

Getting Started with CoppeliaSim

1. Download and Installation

- **Visit: https://www.coppeliarobotics.com**
- **Download the free CoppeliaSim Edu version.**
- **Extract and run the application (no complex installation needed).**

2. Explore the Interface

- **Scene Hierarchy Tree: View all objects in the scene.**
- **Model Browser: Drag-and-drop ready-to-use robots and components.**
- **Simulation Buttons: Start, pause, or stop the simulation.**
- **Scripting Editor: Modify or add Lua scripts to define behaviors.**

3. Build or Import a Robot

- **Use basic shapes to build your own robot or**
- **Import models from CAD tools or the Model Browser.**

4. Add Sensors and Logic

- **Add proximity sensors, vision cameras, encoders, etc.**
- **Write or edit Lua scripts or connect via Python API.**

5. Run Simulation

- **Press the play button to see your robot in action.**
- **Observe sensor feedback and make adjustments.**

6. External Control

- **Use Python, ROS, or MATLAB to send commands or receive data.**
- **Example: Use Python to implement a line-following algorithm remotely.**

Additional Resources

- **Official Docs: https://www.coppeliarobotics.com/helpFiles/en/index.html**
- **GitHub: https://github.com/CoppeliaRobotics**
- **Tutorials: YouTube & online courses**

T1-1: Setup & Initial Implementation

Step 1: Research and Planning

- Understand the working principles of a line follower robot.
- Study CoppeliaSim basics (if not already familiar).
- Review existing models and scripts for similar robots.

Step 2: CoppeliaSim Environment Setup

- Install and configure CoppeliaSim on your system.
- Familiarize with object creation, simulation control, and scripting (Lua or Python via remote API).

Step 3: Design the Arena

- Design a simple **2D track** consisting of a **black line on a white surface**. The evaluation will primarily focus on the **uniqueness and realism of your arena**. Your track should include a **straight segment**, at least **one curve**, and **additional elements** of your choice that reflect your creativity and problem-solving skills.
- To make your simulation more impactful, consider selecting a **real-world industrial scenario** and attempt to replicate **line-following paths** commonly used in such environments.
- Ensure proper scaling for robot sensor detection.

Step 4: Robot Modelling

- Create or import a differential drive robot and take a help of your mechanical partner to make model of it.
- Add basic components:
  - Two wheels and one caster.
  - Two or more line sensors (IR or simulated vision sensors).
- Configure motor and sensor connections.

T1-2 : Programming, Testing & Refinement: The robot must be programmed to complete the entire path efficiently, with performance evaluated based on the minimum time taken to successfully navigate the full track.

Step 1: Sensor Programming and Integration.

- Write a script to read sensor data.
- Determine sensor logic for line detection (left, right, center).

Step 2: Documentation and Video Capture

- Record simulation video showing the robot completing the track.
- Main content/guide lines for your r Videos/PPT/Unique Document :
    - Arena description and its relevance to the industry
    - Robot design
    - Sensor logic
    - Control strategy
    - Challenges faced

Guide lines for the Video and Evaluation Criterial

Stage	Focus Area	Evaluation Criteria (What to Look For)	Video (Time In Sec)	Marks
Introduction	Self & Problem Statement	Concise introduction of the presenter and a clear articulation of the problem being addressed.	5	5
Practical Relevance	Real-Life Correlation	Problem is well-connected to a real-life example or relatable scenario.	10	25
Background Research	Prior Work & References	Includes references to previous work or studies; demonstrates awareness of existing efforts.	10	10
Solution Approach	Problem-Solving Methodology	Presents a unique, innovative, and logical solution approach; well explained.	15	25
Resource Utilization	Use of Tools, Training, and Support	Effectively explains how self training and learning tools were used for your solution.	10	15
6. Publication Readiness	Documentation & Future Scope	Mentions scope for documentation, proof of concept, and publication potential.	5	5
7. Presentation & Delivery	Communication & Technical Quality	Clear speaking, confident body language, good audio quality, fluency in English, and appropriate content density for the time limit.	0	10
8. Conclusion	Learing Outcome from the PS		5	5
		Total	60	100

Guide lines for the PPT

Slide No	Slide Heading	Focus Area	Evaluation Criteria (What to Look For)	Marks
1	Title & Introduction	Problem Statement & Objective	Clear title, presenter introduction, and well-defined problem statement.	10
2	Background & Context	Relevance & Motivation	Explains why the problem is important and worth solving. Real-world connection or need is clearly shown.	10
3	Literature/Market Research	Past Work/Current Landscape	References existing solutions, research, or market gaps. Cited properly with insights.	10
4	Proposed Solution	Innovation & Uniqueness	Clearly presents the core idea/solution with innovative value and originality.	15
5	Methodology/Design	Technical Approach	Shows how the solution works — workflow, algorithm, design flow, or process diagram.	10
6	Implementation	Prototype/Execution Plan	Describes the actual development, tools used, and progress so far (or implementation plan).	10
7	Resources Used	Tools, Technologies, Training	Details the use of training, support, tools (e.g., AI, software), and how they added value.	5
8	Results/Validation	Proof of Concept or Outcomes	Presents results (if available) or expected outcomes. Includes visuals like charts/tables.	10
9	Future Scope/Publication Potential	Scalability & Next Steps	Mentions how the work could be scaled, documented, or published.	5
10	Conclusion & Delivery	Summary, Communication, Visual Appeal	Summarizes key points, good slide design, clarity of speech, confidence, and audience engagement.	15
			Total	100

Unique document Template(A Two page document)

Section 1: Understanding and Design Intent *(150–200 words)*

Key Points to Include:

- Why did you choose this specific layout/design?
- How does this simulation help automate real-world factory tasks?
- What are the functional and technical outcomes expected?
- Collaborator input: Mention if a mechanical partner supported solid modelling.

Section 2: Engineering Physiochemistry Relevance *(150–200 words)*

Highlight the link between the robot simulation and engineering chemistry/physics concepts (i.e., physiochemistry). Show your understanding of interdisciplinary learning.

Key Points to Include:

- Explain relevance of material properties (used in robot wheels/body) – friction, wear, conductivity.
- Sensor interaction with light and surface reflection (IR-based line following).
- Environmental considerations: Electrostatic Discharge (ESD) safe environments in semiconductor cleanrooms.
- Motion control and energy dissipation analysis using Newton's laws and thermodynamic principles.

Section 3: Uniqueness of the Line Follower Robot and Arena *(Open response)*

Present the distinct aspects of your design that make it original.

Key Points to Include:

- What's different in your robot design (e.g., shape, size, configuration)?
- How is your arena different from typical straight tracks?
- Mention any additional complexity (e.g., junctions, conveyor-like curves, simulated elevation).

Section 4: Geo-tagged Selfie Evidence *(Full Page)*

To verify physical participation and engagement with the team or lab environment.

Instructions:

- Include 2–3 geotagged selfies (with timestamp and location visible).
- Label each image with a small caption (what task, who is in photo, date).