CSE140: Introduction to Intelligent Systems
End-Semester Examination

Date: 03/05/2025 **Duration:** 2 hours **Total Marks:** 25

Instructions

- Attempt all questions.
- State any assumptions you have made clearly.
- Standard institute plagiarism policy holds.
- No evaluation without suitable justification. Write all required formulas and steps.
- No marks will be given for answers with only the final answer.
- Draw diagrams (if any) clearly and ensure all annotations are readable.

Q 1: Scenario: You are designing a student research project that collects mobile app usage data (session times, app names) and demographic information (age, gender) from voluntary participants. On the basis of the given scenario answer the following question.
Which of the following most accurately satisfies the obligations placed on the researchers under the **DPDP Act**? **(1 Mark)**

A. Data collection should begin after obtaining broad general consent, as specific informed consent is not mandatory if the research is academic in nature.

B. Researchers must obtain specific, informed consent from participants, ensure the option for easy consent withdrawal, and be fully transparent about data usage, sharing, and retention.

C. It is sufficient for researchers to encrypt collected data and store it securely; consent requirements are secondary for non-sensitive behavioral data.

D. Researchers should seek consent only after the data has been collected, provided no personally identifiable information is included.

Correct Answer: B

Explanation:

DPDP mandates specific, informed, and easy-to-withdraw consent — even if the research is academic and the data seems non-sensitive.

Encryption and secure storage do not replace the consent requirements.

Q 2: Scenario: A student-led research team is creating a dataset of facial images for emotional recognition research. Some participants are under 18 years old. The team wants to store all collected data indefinitely for potential future projects.

To align their project with the **DPDP Act** requirements, the researchers must: **(1 Mark)**

A. Retain all collected data indefinitely as academic datasets are exempt from retention limits.

B. Delete personal data after the specific purpose is completed unless fresh consent is obtained, and ensure verifiable parental consent for all participants under 18.

C. Only obtain consent if the facial images are linked to names or other identifiers.

D. Store collected facial images permanently as biometric data is always exempt from standard data retention requirements in research.

Correct Answer: B

Explanation:

Under DPDP:

- Retention must match the purpose — indefinite storage without purpose justification is not allowed.
- Parental consent is required for any participant under 18, regardless of identifiers.
- Biometric data is sensitive personal data — it is more regulated, not exempt.

Marks : 0.5 if correct + 0.5 for justifications

Q 3: **Scenario:**A team of students is designing a machine learning model to screen internship applications for a competitive IIIT-Delhi's internship program. The model receives several candidate features as input.
As part of fairness-aware ML practice and to ensure non-discriminatory decisions, the team must partition features into two categories:

- **Should Use:** Relevant for fair, performance-based decision-making.
- **Must Avoid:** Known legally or ethically protected attributes that must not influence decisions.

They are considering the following features:
(a) Academic GPA (b) Number of Publications (c) Age (d) Race (e) Disability Status (f) Programming Test Score (g) University Attended (h) Citizenship Status

Which of the following is the most appropriate feature partitioning based on fairness principles and known protected attributes? **(1 Mark)**

A. **Should Use:** GPA, Publications, Programming Score, University Attended
Must Avoid: Age, Race, Citizenship, Disability Status

B. **Should Use:** GPA, Race, Programming Score, Citizenship
Must Avoid: Age, Publications, Disability Status

C. **Should Use:** GPA, Age, Publications, Programming Score
Must Avoid: Race, Disability, Citizenship, University Attended

D. **Should Use:** GPA, Programming Score, Citizenship
Must Avoid: Disability, Race, Publications, Age

Correct Answer: A

Explanation:

Correct 'Should Use':

Academic GPA, Publications, Programming Test Score, University Attended — performance-related features.

Correct 'Must Avoid':

Age, Race, Disability, Citizenship → all are legally protected under U.S. law (per your image), and also align with global ML fairness practices.

Why Others Are Wrong:

B: Incorrectly includes Race and Citizenship in "Should Use".

C: Includes Age in "Should Use", which is a protected attribute.

D: Incorrectly puts Publications in "Must Avoid", which is performance-related. **Marks : 0.5 if correct + 0.5 for justifications**

Q 4: **Assertion (A):** Modern AI systems are more successful at replicating human sensorimotor coordination than abstract reasoning, because physical interaction with the environment is computationally simpler.

Reason (R): Sensorimotor skills, such as walking, grasping, or visual interpretation, evolved through millions of years of biological evolution and rely on deeply subconscious, distributed neural processes that are hard to formalize algorithmically. **(1 Mark)**

A. Both A and R are true, and R is the correct explanation of A.
B. Both A and R are true, but R is not the correct explanation of A.
C. A is true, but R is false.
D. A is false, but R is true.

Correct Answer: D

Explanation:

Assertion is false: AI systems are better at reasoning tasks like algebra, optimization, and formal logic, than at physical or perceptual tasks like walking on uneven terrain, recognizing objects under occlusion, or interpreting facial expressions — which are computationally very hard, despite seeming effortless to humans.

Reason is true: These "easy for humans" tasks are hard for machines precisely because they stem from biologically ancient, embodied intelligence, honed over evolutionary timescales.

Thus, the assertion is a common misconception — and the reason provides a correct explanation of the opposite reality.

Q 5: **Assertion (A):** Using both active and passive sensors improves the reliability of autonomous systems.

Reason (R): Active sensors perform well in high-noise or dark environments, while passive sensors provide non-invasive ambient data. **(1 Mark)**

- **(a)** Both A and R are true, and R is the correct explanation of A
- **(b)** Both A and R are true, but R is not the correct explanation of A
- **(c)** A is true, R is false
- **(d)** A is false, R is true

Answer: (a)

Q 6: Which of the following statements about Odometry and GPS in robotics are true? **(1 Mark)**

(**Note:** This question has multiple correct options. No partial marking will be there.)

(A) Odometry is the use of sensors to measure the movement of a robot's wheels or joints to estimate its position and orientation.

(B) GPS is the use of Global Positioning System (GPS) signals to determine the location of a robot.

(C) Odometry uses the equations:

$$x(t) = x(t-1) + \Delta s \cdot \cos(\theta(t-1))$$
$$y(t) = y(t-1) + \Delta s \cdot \sin(\theta(t-1))$$
$$\theta(t) = \theta(t-1) + \Delta\theta$$

(D) GPS uses signals from GPS satellites to triangulate the robot's position on the Earth's surface.

Answer:

All options (a), (b), (c), and (d) are correct. [Refer to Slide 25 in the Robotics lecture slides]

Reasoning:

(a) Odometry involves the use of sensors to measure the movement of a robot's wheels or joints to estimate its position and orientation. This technique is based on accumulating incremental changes in movement over time.

(b) GPS (Global Positioning System) indeed uses signals from satellites to determine the geographic location of a device, including a robot. The system calculates the robot's exact location on Earth by measuring the time delay between the transmission and reception of signals from a network of satellites orbiting the Earth.

(c) The equation provided a basic representation of odometry for correctly estimating a robot's current position $(x(t), y(t))$ and orientation $\theta(t)$ based on its previous position $(x(t-1), y(t-1))$, orientation $\theta(t-1)$, and the incremental movement Δs and change in orientation $\Delta\theta$ since the last measurement. This formula assumes movement on a plane, where Δs represents the linear displacement and $\Delta\theta$ represents the angular displacement.

(d) GPS uses signals from at least four GPS satellites to triangulate a receiver's (in this case, a robot's) position on Earth's surface. The process involves calculating the distance to each satellite based on the time delay between when a signal is sent and when it is received. With distances from several satellites, the system can pinpoint the receiver's location with respect to the Earth's latitude, longitude, and sometimes altitude.

NOTE: Binary Marking is done in this question. Full marks are given only when all options are correctly selected. No partial marking is done as indicated in the question itself.

Q 7: You are deploying a sensor fusion system combining two sensors: **(1 Marks)**

- **Sensor A (Active):** Accuracy 90%, consumes 5 W power
- **Sensor B (Passive):** Accuracy 70%, consumes 1 W power

If you combine them using simple weighted averaging (weight proportional to accuracy), compute:

(a) The fused accuracy of the system.
(b) The effective power consumption.

Answer:

a) The fused accuracy of the system: (1 mark)

For simple weighted averaging, the weight of each sensor is proportional to its accuracy. Sensor A has accuracy 90%, weight = 90. Sensor B has accuracy 70%, weight = 70.

The fused accuracy A_{fused} is:

$$A_{\text{fused}} = \frac{(A_1 \times W_1) + (A_2 \times W_2)}{W_1 + W_2}$$

where:

$$A_1 = 90, W_1 = 90 \quad \text{for Sensor A}$$

$$A_2 = 70, W_2 = 70 \quad \text{for Sensor B}$$

$$A_{\text{fused}} = \frac{(90 \times 90) + (70 \times 70)}{90 + 70} = \frac{8100 + 4900}{160} = \frac{13000}{160} = 81.25\%$$

b) The effective power consumption: (1 mark)

Since both sensors are actively used in the system, the total power consumption is the sum of the power consumed by each sensor.

$$\text{Power}_{\text{Sensor A}} = 5\,\text{W}$$
$$\text{Power}_{\text{Sensor B}} = 1\,\text{W}$$
$$\text{Total Power Consumption} = 5\,\text{W} + 1\,\text{W} = \boxed{6\,\text{W}}$$

Marks 0.5 for (a) if weights taken proportional and correct calculation and 0.5 for (b)

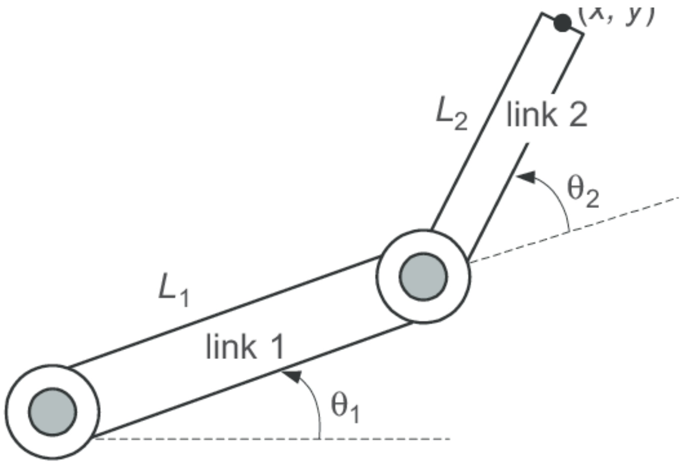

Figure 1: Robot Arm

Q 8: Consider a mobile robot moving on a horizontal surface. Suppose that the robot can execute two types of motion: **(3 Marks)**

- Rolling forward a specified distance.
- Rotating in place through a specified angle.

The state of such a robot can be characterized in terms of three parameters (x, y, ϕ), where x and y are the coordinates of the robot's center of rotation, and ϕ is the orientation angle from the positive x-axis. The action *Roll(D)* changes the state (x, y, ϕ) to $(x + D\cos(\phi), y + D\sin(\phi), \phi)$, and the action *Rotate(θ)* changes the state (x, y, ϕ) to $(x, y, \phi + \theta)$.

Suppose the robot is initially at $(0, 0, 0)$ and executes the actions *Rotate(60°)*, *Roll(1)*, *Rotate(25°)*, *Roll(2)*. Now suppose that the robot has imperfect control of its own rotation, and that, if it attempts to rotate by θ, it may actually rotate by any angle between $\theta - 10°$ and $\theta + 10°$.In that case, if the robot attempts to carry out the sequence of actions above, there is a range of possible ending states. What are the minimal and maximal values of the x-coordinate, the y-coordinate, and the orientation in the final state?

Answer:

With rotation error, the possible ranges are: - First rotation: $60° \pm 10°$ (from 50° to 70°) - Second rotation: $25° \pm 10°$ (from 15° to 35°)

Minimal x-coordinate: $1 \cdot \cos(70°) + 2 \cdot \cos(105°) = -0.176$

Achieved when the first rotation is actually 70° and the second is actually 35°.

Maximal x-coordinate: $1 \cdot \cos(50°) + 2 \cdot \cos(65°) = 1.488$

Achieved when the first rotation is actually 50° and the second is actually 15°.

Minimal y-coordinate: $1 \cdot \sin(50°) + 2 \cdot \sin(65°) = 2.579$

Achieved when the first rotation is actually 50° and the second is actually 15°.

Maximal y-coordinate: $1 \cdot \sin(70°) + 2 \cdot \sin(90°) = 2.94$

Achieved when the first rotation is actually 70° and the second is actually 20°.

Marking:

- Correct approach = 1.5 marks
- Correct answer for the subparts: 1.5 marks

Q 9: Consider the robot arm shown in Figure 1. Assume that the robot's base element is 60 cm long and that its upper arm and forearm are each 40 cm long. The inverse kinematics of a robot is often not unique. Provide an explicit closed-form solution of the inverse kinematics for this arm. **(2 Marks)**

Answer:
For the robot arm shown in Figure 1, with link lengths L1 and L2:

Let θ_1 be the angle of the first link from the horizontal axis and θ_2 be the angle of the second link relative to the first link. We need to find θ_1 and θ_2 given the end-effector position (x, y).

Step 1: The forward kinematics equations for this robot are:

$$x = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \tag{1}$$

$$y = L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \tag{2}$$

Step 2: For inverse kinematics, we first calculate θ_2:

$$x^2 + y^2 = (L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2))^2 + (L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2))^2 \tag{3}$$

$$= L_1^2 + L_2^2 + 2L_1L_2 [\cos(\theta_1)\cos(\theta_1 + \theta_2) + \sin(\theta_1)\sin(\theta_1 + \theta_2)] \tag{4}$$

$$= L_1^2 + L_2^2 + 2L_1L_2 \cos(\theta_2) \tag{5}$$

Therefore, solving for $\cos(\theta_2)$:

$$\cos(\theta_2) = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2} \tag{6}$$

And the angle:

$$\theta_2 = \pm \arccos\left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \tag{7}$$

The $\pm$ indicates two possible solutions (elbow-up and elbow-down configurations).

Step 4: Derivation of θ_1:

Let ϕ be the angle between the positive x-axis and the vector from the base to the end-effector:

$$\phi = \arctan 2(y, x) \tag{8}$$

Now, define the auxiliary angle β between the first link and the line to the end-effector. Using triangle geometry:

$$\beta = \arctan 2\left(L_2 \sin(\theta_2), L_1 + L_2 \cos(\theta_2)\right) \tag{9}$$

Then, the angle of the first joint is:

$$\begin{aligned}\theta_1 &= \phi - \beta && (10)\\ &= \arctan 2(y,x) - \arctan 2\left(L_2\sin(\theta_2), L_1 + L_2\cos(\theta_2)\right) && (11)\\ &= \arctan 2(y,x) - \arctan 2\left(\frac{L_2\sin(\theta_2)}{L_1 + L_2\cos(\theta_2)}\right) && (12)\end{aligned}$$

Note: This solution assumes the point (x, y) is reachable, which requires:

$$|L_1 - L_2| \leq \sqrt{x^2 + y^2} \leq L_1 + L_2 \tag{13}$$

Marking:

- Correct formula for θ_2: 1 mark
- Correct formula for θ_1: 1 mark

Q 10: You are working with the ROS Turtlesim simulator to build basic **navigation** and **odometry tracking** functionalities using Python. Complete the implementation of the following two ROS nodes.

(a) `turtle_navigator.py`

Create a ROS node that: **(2 Marks)**

- Subscribes to `/turtle1/pose` to obtain the turtle's current position.
- Publishes velocity commands to `/turtle1/cmd_vel` to move the turtle toward a specified target position (x, y).
- Implements a **proportional controller** to reduce the distance and orientation error.
- Stops the turtle when it reaches within a **0.1 meter threshold** of the target.

Starter Code (`turtle_navigator.py`):

```
#!/usr/bin/env python

import rospy
from geometry_msgs.msg import Twist
from turtlesim.msg import Pose
import math

target_x = 5.0
target_y = 5.0
distance_threshold = 0.1

def pose_callback(data):
    === Your code here ===
    Compute distance to target and angle difference
    Use proportional control to set linear.x and angular.z
    If distance <= threshold, stop the turtle
    Publish velocity message
    pass

if __name__ == "__main__":
    rospy.init_node("turtle_navigator")
    velocity_publisher = rospy.Publisher("/turtle1/cmd_vel", Twist, queue_size=10)
    rospy.Subscriber("/turtle1/pose", Pose, pose_callback)
    rospy.spin()
```

Answer: (a)

```
# turtle_navigator.py
import rospy
from geometry_msgs.msg import Twist
from turtlesim.msg import Pose
import math
```

```python
target_x = 5.0  # Desired x position
target_y = 5.0  # Desired y position
distance_threshold = 0.1

def pose_callback(data):
    global target_x, target_y

    # Get current position
    x, y, theta = data.x, data.y, data.theta

    # Calculate distance and angle to target
    distance = math.sqrt((target_x - x) ** 2 + (target_y - y) ** 2)
    angle_to_target = math.atan2(target_y - y, target_x - x)

    # Velocity message
    vel_msg = Twist()

    if distance > distance_threshold:
        # Simple Proportional Controller
        vel_msg.linear.x = 1.5 * distance
        vel_msg.angular.z = 4.0 * (angle_to_target - theta)
    else:
        # Stop the turtle when target is reached
        vel_msg.linear.x = 0.0
        vel_msg.angular.z = 0.0

    # Publish velocity
    velocity_publisher.publish(vel_msg)

if __name__ == "__main__":
    rospy.init_node("turtle_navigator")
    velocity_publisher = rospy.Publisher("/turtle1/cmd_vel", Twist, queue_size=10)
    rospy.Subscriber("/turtle1/pose", Pose, pose_callback)
    rospy.spin()
```

Marking:

- Correct logic and no syntax errors : 2 marks
- Correct logic with syntax errors : 1 mark

(b) `turtle_odometer.py`

Create a ROS node that: **(2 Marks)**

- Subscribes to `/turtle1/pose` to track the turtle's real-time position.
- Calculates the **total Euclidean distance** covered by the turtle.
- Logs the distance traveled every **1 second** while navigating.

Starter Code (`turtle_odometer.py`):

```python
#!/usr/bin/env python

import rospy
from turtlesim.msg import Pose
import math
import time

prev_x, prev_y = None, None
total_distance = 0.0
last_print_time = time.time()

def pose_callback(data):
    === Your code here ===
    Calculate distance between current and previous positions
    Update total_distance and print it every 1 second
```

```
    pass

if __name__ == "__main__":
    rospy.init_node("turtle_odometer")
    rospy.Subscriber("/turtle1/pose", Pose, pose_callback)
    rospy.spin()
```

Answer: (b)

```
# turtle_odometer.py
import rospy
from turtlesim.msg import Pose
import math
import time

# Variables to store distance and previous position
prev_x, prev_y = None, None
total_distance = 0.0
last_print_time = time.time()

def pose_callback(data):
    global prev_x, prev_y, total_distance, last_print_time

    # Get current position
    x, y = data.x, data.y

    # Calculate distance covered if previous position is available
    if prev_x is not None and prev_y is not None:
        distance = math.sqrt((x - prev_x) ** 2 + (y - prev_y) ** 2)
        total_distance += distance

    # Update previous position
    prev_x, prev_y = x, y

    # Print distance covered every 1 second (bonus task)
    if time.time() - last_print_time >= 1.0:
        rospy.loginfo("Total distance covered: {:.2f} meters".format(total_distance))
        last_print_time = time.time()

if __name__ == "__main__":
    rospy.init_node("turtle_odometer")
    rospy.Subscriber("/turtle1/pose", Pose, pose_callback)
    rospy.spin()
```

Marking:

- Correct logic and no syntax errors : 2 marks
- Correct logic with syntax errors : 1 mark

Q 11: (a) Calculate the **Statistical Parity Difference (SPD)** for hiring decisions using the provided sample dataset. **(1 Mark)**

(b) Interpret the SPD value in the context of fairness. What does it suggest about the model's potential bias? **(1 Mark)**

Sample Dataset:

Applicant ID	Gender	Experience (Years)	Test Score	Hired (Yes/No)
1	Male	4	85	Yes
2	Female	5	82	No
3	Male	6	90	Yes
4	Female	3	80	No
5	Female	6	88	Yes

Answer:

a) Calculate the Statistical Parity Difference (SPD) for hiring decisions: (1 mark)

Definition:

$$\text{SPD} = \Pr(Y = 1 \mid D = 0) - \Pr(Y = 1 \mid D = 1)$$

Where:

- $Y = 1$: Favorable outcome (hired)
- $D = 0$: Unprivileged group (Female)
- $D = 1$: Privileged group (Male)

Dataset Summary:

Gender	Applicant IDs	Hired (Yes)
Male (Privileged)	1, 3	2/2
Female (Unprivileged)	2, 4, 5	1/3

Step 1: Compute group-wise probabilities

$$\Pr(Y = 1 \mid D = 1) = \frac{2}{2} = 1.00$$

$$\Pr(Y = 1 \mid D = 0) = \frac{1}{3} \approx 0.33$$

Step 2: Calculate SPD

$$\text{SPD} = 0.33 - 1.00 = -0.67$$

b) Interpretation of SPD: (1 mark)

The SPD value of -0.67 indicates that the unprivileged group (females) is significantly less likely to be hired than the privileged group (males).

This suggests a strong potential **bias in favor of male applicants**, and the model may be violating fairness under the statistical parity criterion.

Marking Scheme:

(a)

- 1 mark: Correct formula and correct value found
- 0.5 marks: Only formula and incorrect value
- 0 marks: Incorrect formula and incorrect value

(b)

- 1 mark: For correct interpretation of SPD value
- 0.5 marks: Partially correct interpretation
- 0 marks: Incorrect interpretation / Interpretation without correct SPD value

Q 12: IoT-based surveillance systems using cameras (passive) and radar (active) are deployed in a smart campus. **(2 Marks)**

(a) Discuss the privacy implications of both sensor types.

(b) Which sensing mode poses greater risk and why?

(c) Propose three architectural strategies to minimize data leakage in such IoT systems.

Answer:

a) Privacy implications of both sensor types: (1 mark)

- **Cameras (passive sensors):** Collect rich visual data that can identify individuals, their behaviors, and patterns. Can capture sensitive information like faces, interactions, and activities without active participation.
- **Radar (active sensors):** Detect presence, movement, and basic positioning without capturing identifiable features. Less intrusive but still track location and movement patterns.

b) Which sensing mode poses greater risk and why: (1 mark)

- Cameras (passive sensors) pose greater privacy risks due to identifiable data. They capture detailed

visual information that can directly identify individuals, track behaviors, and potentially record sensitive activities without consent.
- Radar systems detect presence and movement without capturing identifiable features, making them inherently less invasive for privacy.

c) Architectural strategies to minimize data leakage: (1 mark)

- **Edge AI for local processing:** Process data on-device to extract only relevant information without transmitting raw footage/data to central servers.
- **Anonymization techniques:** Implement face blurring, identity stripping, and data generalization before storage or transmission.
- **End-to-End Encryption:** Ensure all data transmitted between devices and servers is encrypted to prevent unauthorized access during transmission.

Marking:

- **0.5 mark: part(a) privacy implications but not in terms of day or night or location or number of sensors**
- **0.5 mark: part(b) risk in terms of privacy or any Hazardous issue**
- **0.5 + 0.5 mark: part(c) for only architecture solution**

Q 13: Explain the following concepts: **(5 Marks)**

(a) Forward Kinematics and Inverse Kinematics

(b) Types of Actuators based on energy

(c) Open-Loop Control and Closed-Loop Control

(d) Moravec's Paradox and its implications for robotics and AI.

(e) What is SLAM in robotics, and why is it important?

Answer:

a) Forward Kinematics and Inverse Kinematics: (1 mark)

- **Forward Kinematics:** Determines end-effector position/orientation from joint variables. Maps from joint space to workspace with unique solutions.
- **Inverse Kinematics:** Determines joint variables needed for desired end-effector position/orientation. Maps from workspace to joint space with potentially multiple, zero, or infinite solutions.

b) Types of Actuators based on energy: (1 mark)

- **Electric actuators:** Motors, servos, solenoids. Precise control, common in robotics.
- **Hydraulic actuators:** Use pressurized fluid. High force output for heavy applications.
- **Pneumatic actuators:** Use compressed air. Lightweight, good for quick movements.

c) Open-Loop Control and Closed-Loop Control: (1 mark)

- **Open-Loop Control:** No feedback. Performs based on predetermined inputs without monitoring results. Simpler but less accurate with disturbances.
- **Closed-Loop Control:** Uses sensors to compare actual output with desired output. Adjusts based on error. More complex but more accurate and robust.

d) Moravec's Paradox and its implications for robotics and AI: (1 mark)

- Moravec's Paradox states that tasks humans find easy (like perception and movement) are harder for robots, while tasks we find intellectually difficult (like math or logic) are easier to automate.
- This paradox arises because sensory-motor skills evolved over millions of years and are largely subconscious, while reasoning is relatively recent and more rule-based.
- Implication: Robots struggle more with tasks involving emotional intelligence, social cues, or motor coordination than with calculations or logic.

e) What is SLAM in robotics, and why is it important? (1 mark)

- SLAM stands for Simultaneous Localization and Mapping. It allows robots to build a map of an unknown environment while determining their own position within it.
- Uses sensors like cameras, lidar, and IMU.
- Algorithms like Extended Kalman Filter or Particle Filter process the sensor data.

- SLAM is critical for autonomous navigation, especially in dynamic or unfamiliar environments, as it helps robots move intelligently without GPS.

Marking:

- Each sub-question: 1 mark
- Complete explanation: 1 mark
- Partial explanation: 0.5 mark
- incorrect : 0 marks