Name: Shamsa

Rollno.047

Class: BSDS-3A

Documentation: Model-Based Reflex Agent for Heater Control

# 1. Introduction

This project implements a Model-Based Reflex Agent in Python to simulate an intelligent heating system. Unlike a simple reflex agent that only reacts to the current environment, this agent also remembers the previous action. This prevents unnecessary switching of the heater ON/OFF, ensuring efficient and stable performance.

# 2. Objective

- To design a model-based reflex agent that regulates room temperature using a heater.  
- To avoid redundant switching by considering both the current percept (temperature) and the previous action.

# 3. Working Principle

1. The agent receives input from the environment (room temperature) through its sensor.  
2. Based on the fixed desired temperature (threshold), it decides whether to turn ON or turn OFF the heater.  
3. The agent stores the previous action so that it does not repeatedly toggle the heater unnecessarily.  
4. The actuator then performs the action (simulated here using print statements).

# 4. Rules of Operation

- If Current Temperature < Fixed Temperature (Threshold):  
 → Heater should be turned ON.  
 → If it is already ON, the heater remains ON.  
  
- If Current Temperature ≥ Fixed Temperature:  
 → Heater should be turned OFF.  
 → If it is already OFF, the heater remains OFF.

# 5. Code Explanation

class ModelBaseReflexAgent:  
 def \_\_init\_\_(self, temp):  
 self.fixed\_temp = temp # desired threshold temperature  
 self.previous\_action = None # memory of last action  
  
 def sensor(self, temp):  
 self.current\_temp = temp # perceives current room temperature  
  
 def performance(self):  
 action = None  
  
 # Rule 1: If temperature is below threshold → turn ON heater  
 if self.current\_temp < self.fixed\_temp:  
 if self.previous\_action != "Turn on the Heater":  
 action = "Turn on the Heater"  
 self.previous\_action = action  
 else:  
 action = "Heater remains ON"  
  
 # Rule 2: If temperature is equal or above threshold → turn OFF heater  
 else:  
 if self.previous\_action != "Turn off the Heater":  
 action = "Turn off the Heater"  
 self.previous\_action = action  
 else:  
 action = "Heater remains OFF"  
  
 return action  
  
 def actuator(self):  
 action = self.performance()  
 print(self.current\_temp, "°C => Action:", action)  
  
  
# --- Example Run ---  
rooms = {  
 "Living Room": 20,  
 "Drawing Room": 28,  
 "Kitchen": 34,  
 "Bedroom": 18,  
 "Balcony": 14,  
}  
  
agent = ModelBaseReflexAgent(26) # fixed desired temperature  
  
for room, temp in rooms.items():  
 print(room, end=":\t")  
 agent.sensor(temp)  
 agent.actuator()

# 6. Example Output

Living Room: 20°C => Action: Turn on the Heater  
Drawing Room: 28°C => Action: Turn off the Heater  
Kitchen: 34°C => Action: Heater remains OFF  
Bedroom: 18°C => Action: Turn on the Heater  
Balcony: 14°C => Action: Heater remains ON

# 7. Advantages of Model-Based Reflex Agent

- Prevents frequent switching of heater ON/OFF.  
- Maintains a stable indoor temperature.  
- Uses internal state (memory) to improve decision-making.  
- More efficient than a simple reflex agent.

# 8. Applications

- Smart home heating systems.  
- Energy-efficient HVAC systems.  
- Automated temperature control in offices and industries.