ZENITH STUDY HUB

Artificial Intelligence for Robotics

UNIT I | FOUNDATIONS

Intelligent agents, state spaces, AI techniques and robotic applications

Syllabus-first, visually explained and exam-ready

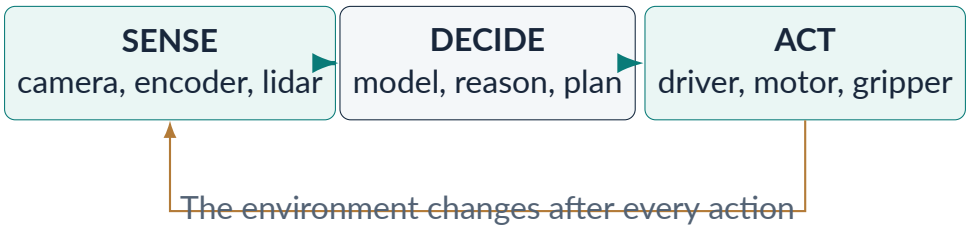

Figure 1. Original agent–environment loop for a robotics system. Sensing and acting may include non-AI components.

Official Unit I syllabus coverage

Introduction: AI versus Automation versus Robotics; intelligent agents; problem formulation; state-space representation; AI techniques (search, optimization, learning, rule-based systems); role of AI in manufacturing and robotics.

Applications: industrial robots, mobile robots, flexible manufacturing systems and real-world intelligent-system examples.

Course: PCC301-ROA | TE Robotics & Automation | SPPU 2024 Pattern

Prepared for learning review: Om Deshmukh | 21 November 2026

Your complete learning route

Exact syllabus item	Where to learn and what to practise
AI / automation / robotics	Section 1: definitions, overlapping cases, comparison and examples.
Intelligent agents	Section 2: perception, actions, agent architectures, PEAS and environment types.
Problem and state space	Section 3: initial state, actions, transitions, goal test, costs and route example.
Search / learning / rules	Section 4: principles, strengths, failure modes and application matching.
Manufacturing applications	Section 5: industrial robots, mobile robots and flexible manufacturing cells.
Real-world examples	Section 6: warehouse, inspection and adaptive production case studies.

Important syllabus boundary: Unit I establishes the foundations. Search algorithms and genetic algorithms are studied numerically in Unit II; probabilistic reasoning is developed in Unit III; detailed machine learning is Unit IV; machine vision operators belong to Unit V. This guide builds conceptual clarity without falsely claiming to teach all later units.

How to study this unit effectively

1. **Learn the words:** distinguish AI, automation, robots, agents and goals.
2. **Trace the diagrams:** explain the input–decision–output flow and the state-space graphs without memorising arrows blindly.
3. **Solve the examples:** calculate route cost, state combinations, defect precision/recall, and the two-job schedule by hand.

1 Artificial Intelligence, Automation and Robotics

In simple language
Artificial Intelligence (AI) concerns systems that use computation to perceive, reason, learn or choose actions in ways associated with intelligent behaviour. Automation is the execution of a process with reduced repeated human intervention. Robotics concerns embodied machines that sense and act in the physical world. These ideas overlap, but they are not synonyms.

1.1 The key difference, with examples

A fixed conveyor controlled by a timer is **automation**: it repeats a preset sequence. A robot arm that follows a taught path is **robotics and automation** even if no machine-learning model is present. A quality-inspection algorithm that classifies images may use **AI without being a robot**. A mobile robot that uses learned perception and searches for a route combines all three ideas.

Idea	What it primarily asks	Example
Automation	Which repeated operation should happen next?	Conveyor starts when a sensor detects an item.
Robotics	How can a physical mechanism move or interact?	Six-axis arm transfers a part.
Artificial Intelligence	Which action or interpretation is suitable under the current information?	Vision classifier identifies the part type.

1.2 A clear, original relationship diagram

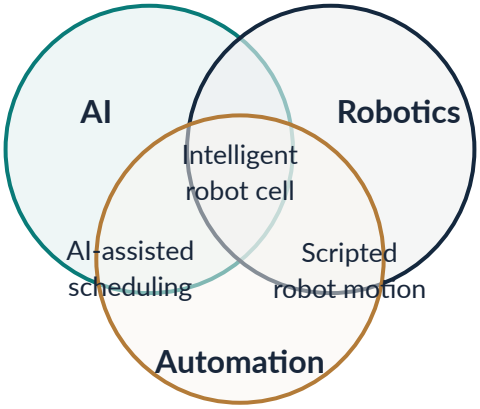

Figure 2. Overlapping domains: systems can use one, two or all three fields. The diagram is conceptual rather than a strict mathematical partition.

1.3 When does a programmed system become “intelligent”?

There is no single simple sensor or algorithm that instantly makes something AI. A common engineering test is whether the system can *select among alternatives using models, evidence, plans, rules, search or learned patterns* for a defined task. Fixed logical rules can be part of AI, although not all rule-based automation is usefully described as intelligent. The important question is the behaviour and evidence, not a marketing label.

1.4 Learning versus programming by hand

A rule-based inspection system might say, “if measured diameter is outside tolerance, reject the part.” Its rule is transparent and human-designed. A learned classifier receives examples of good and defective parts, trains model parameters and predicts a label for a new image. Both must be validated; a high training score does not guarantee reliable factory performance. An AI model can still sit inside an ordinary non-AI safety and sequencing framework.

Check your understanding
A robot arm repeats the same pickup positions all day. Is it necessarily AI? No. It is a robot doing automated work. Explain what additional planning, classification or adaptive decision function could justify an AI component.

1.5 Robotics software versus AI: a practical stack

A complete robot may have sensor drivers, signal filters, an operating system, motor controllers, a perception module, a planner, a state machine, a human interface and a protective safety system. Not all these blocks use AI. Low-level servo loops still rely on feedback control and hardware timing even when a high-level AI planner decides where to move.

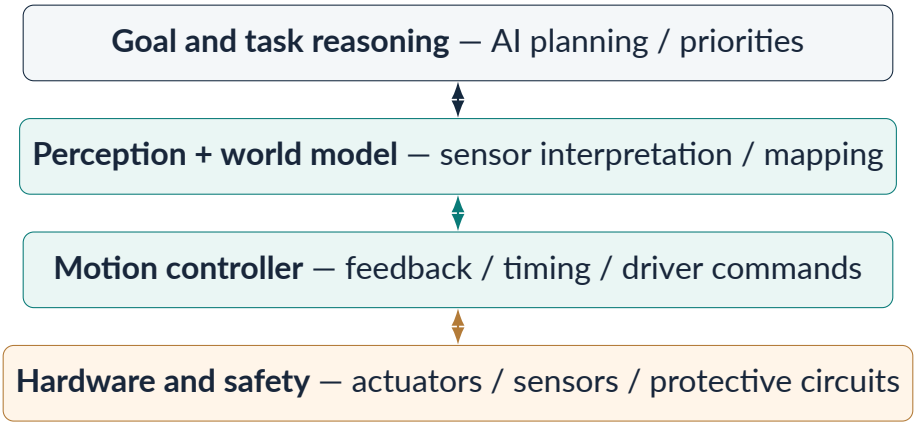

Figure 3. A layered robotics stack. AI is only some of the software; physical control and safety remain independent engineering requirements.

1.6 Where AI is useful in manufacturing

AI methods are particularly relevant when an industrial problem contains variable products, noisy measurements, multiple scheduling choices or uncertain environments. Example applications include object and defect detection, robot path planning, adaptive process monitoring, condition-based maintenance, production scheduling and warehouse transport. They are not universally needed: a simple high-speed deterministic stop circuit may be better implemented with conventional controls.

WORKED EXAMPLE: Choose AI or a fixed rule

A machine checks whether an object has reached a fixed end stop. One reliable digital limit switch and a rated control system may be sufficient. A second machine must identify one of 120 visually similar part variants under changing light. A validated vision model may be useful, but its errors, training distribution and lighting robustness must be measured.

2 Intelligent Agents: Perception, Decisions and Action

In simple language
An intelligent agent is a system that receives information about an environment through sensors, uses a policy or decision mechanism, and acts through actuators. The agent's success is measured by a chosen performance criterion, not by whether it appears humanlike.

2.1 The five important terms

Agent is the decision-maker. **Environment** is everything relevant outside the agent. **Percept** is the incoming sensor information at a particular instant. **Percept sequence** is the history of such information. **Action** is an output selected to affect the environment. **Policy** maps observations or internal states to actions. The policy may be a lookup table, rules, an optimizer or a learned model.

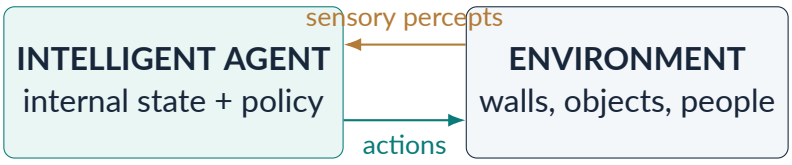

Figure 4. Two-way interaction between an agent and its task environment.

2.2 PEAS: specify the task before choosing an algorithm

PEAS stands for **Performance measure**, **Environment**, **Actuators** and **Sensors**. It is a standard framework for describing what an agent must accomplish and how it will interact with its world.

PEAS element	Warehouse-delivery mobile robot example
Performance	Deliver correct items safely; minimize unnecessary distance, delay, energy and collisions.
Environment	Warehouse lanes, shelves, doors, workers, temporary obstacles and docking stations.
Actuators	Wheel motors, brake/driver enable, signal lights and audible indicator.
Sensors	Lidar/range sensors, wheel encoders, camera, IMU, battery status and safety inputs.

Avoid a common PEAS error. A performance measure is not the same as an actuator command. “Move forward” is an action; “deliver items safely and on time” is a performance objective. A good PEAS description also identifies the environment’s hazards and constraints.

2.3 A second PEAS example: robot arm for visual sorting

Performance	Correct classification, correct bin, cycle time, product damage rate and safe handling.
Environment	Moving conveyor, parts, bins, lighting, nearby operators and tool fixtures.
Actuators	Joint drive commands, gripper, diverter, lamp and guarded stop output.
Sensors	Camera, conveyor encoder, joint encoders, gripper switch, force sensing and guarded-zone inputs.

2.4 Types of agents: know their limits

Agent approach	Mechanism and robotics illustration
Simple reflex	Responds to current percept using direct condition-action rules, such as an obstacle-avoidance threshold.
Model-based reflex	Maintains internal state to account for information not immediately visible, such as a recent obstacle location.
Goal-based	Selects actions because they lead toward a specified destination.
Utility-based	Trades off competing preferences, such as energy versus travel time.
Learning agent	Changes its predictive model or policy from experience or labelled examples.

These are conceptual descriptions, not mutually exclusive categories. A real industrial mobile robot may combine all five features in different software modules.

2.5 What does rationality mean?

A rational agent chooses an action expected to optimize a specified performance measure using the available percept history and knowledge. *Rational* does not mean omniscient or perfect: the agent may receive noisy data, have limited computation and lack a correct world model. A rational decision under uncertainty can still turn out badly. This distinction matters when evaluating whether an intelligent robot failed from a bad model, a faulty sensor or an unrealistic performance target.

2.6 Task environments and how they change algorithms

Environment property	Practical meaning
Observability	Can the agent measure all state variables needed for its decision?
Uncertainty	Does an action always have the same result, or is there randomness/noise?
Decision dependence	Does each decision stand alone, or affect later decisions?
World changes	Can the world change while the agent deliberates?
Representation	Are states/actions selected from finite steps or real-valued quantities?
Other agents	Is another agent also acting or competing/cooperating?

WORKED EXAMPLE: Classify an autonomous warehouse robot's environment

The warehouse is normally *partially observable* because shelves and people can occlude sensors; *stochastic* to the extent that human movement and wheel slip are uncertain; *sequential* because one turn changes later routes; *dynamic* because forklifts and workers move; and it mixes *discrete* job assignments with *continuous* geometry and control. These labels depend on the chosen modelling level.

2.7 Performance-metric calculation example

A route plan A takes 90 s and uses 200 J, while route B takes 105 s and uses 160 J. If the illustrative task cost is $J = t + 0.10E$, with t measured in seconds, E in joules, and a weight of 0.10 seconds per joule, then $J_A = 90 + 0.1(200) = 110$ and $J_B = 105 + 0.1(160) = 121$. Under this particular, stated cost definition, A wins. Changing the energy weight may change the decision; weights are design choices, not universal physical laws.

2.8 A practical agent design checklist

1. Specify what a good outcome means, including safety and failure cost.

2. List the sensors, operating conditions and missing information.
3. Define permitted actions and physical constraints.
4. Choose a world model and decision mechanism.
5. Define interfaces to ordinary motion-control loops and protective circuits.
6. Test against unseen environments and faults, not only familiar examples.

3 Problem Formulation and State-Space Representation

In simple language
Before a computer can search for a plan, it needs a structured description of the task: where it starts, what actions are allowed, how actions change the state, which conditions count as success, and how expensive a solution is. This description is called problem formulation.

3.1 The five elements of a search problem

1. **State space** S**:** all states the model considers possible.
2. **Initial state** s_0**:** the starting situation.
3. **Actions** $A(s)$**:** the actions permitted in each state.
4. **Transition model** $T(s, a)$**:** the successor state after action a in state s.
5. **Goal test** $G(s)$ **and path cost** $c(s, a, s')$**:** success criterion and the cost used to compare paths.

Sometimes a problem includes an explicit cost of waiting, energy or risk. Search may solve a symbolic route but a separate physical controller still has to execute it within speed, stability and collision constraints.

3.2 A state is a model, not a picture

For a grid-based warehouse robot, the simple state may be (x, y), a cell location. If turning matters, extend it to (x, y, θ). If pickup matters, add $q \in \{\text{empty}, \text{loaded}\}$. If battery constraints matter, add a battery variable. Each

addition makes the model more expressive but grows the state space. A good representation contains *just enough* information to predict allowed future behaviour.

3.3 State-space explosion

If the robot has 20 possible x positions, 20 possible y positions, 4 headings, 2 load conditions and 10 battery levels, a naive finite model has $20 \cdot 20 \cdot 4 \cdot 2 \cdot 10 = 32{,}000$ possible combinations before walls and restrictions. Not all are reachable; the calculation shows why representation design matters.

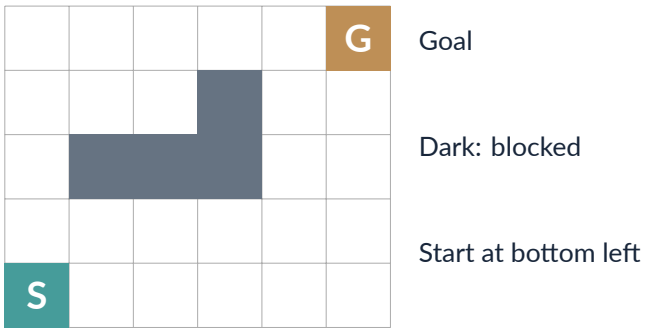

Figure 5. Original six-column, five-row state-space illustration with obstacle cells. Route costs depend on allowed moves and map assumptions.

3.4 The difference between search tree and state graph

A **state graph** represents distinct states and legal transitions. A **search tree** represents possible sequences of actions explored from the starting state, and may reach the same graph state through multiple action sequences. Failing to track explored states can cause repeated work or loops. The next syllabus unit studies specific algorithms and their properties.

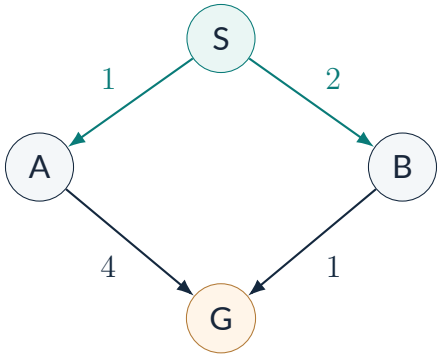

Figure 6. Small weighted state graph. S–B–G costs $2 + 1 = 3$, while S–A–G costs $1 + 4 = 5$.

WORKED EXAMPLE: Choose the cheaper of two graph paths

The graph above has $S \to A$ cost 1, $A \to G$ cost 4, $S \to B$ cost 2 and $B \to G$ cost 1. Summing edge costs gives path $S \to A \to G$ cost 5 and path $S \to B \to G$ cost 3. The second is cheaper under the stated additive cost model. Do not simply choose the cheapest first edge: a locally attractive first step need not lead to the best complete route.

3.5 Goal test versus safety constraints

Reaching the target cell is a *goal*; staying out of a restricted human working zone is a *constraint*. A planner that finds a short path through a forbidden zone has not solved the real task. Physical collision avoidance and emergency-stop safety also require properly validated execution controls, not just a graph path.

3.6 Good and poor problem formulations

Poor model	Better model
“Move quickly”	Reach defined goal, cost weighted time/energy and respected motion limits.
“Robot is at shelf”	State includes cell, heading and whether the payload is held.
“Walk anywhere”	Allowed transitions avoid walls and no-go zones.
“Stop when done”	Goal test uses a verified docking signal or geometric tolerance.

4 Four Major AI Technique Families

In simple language
AI techniques solve different engineering problems. Search chooses among sequences of actions; optimization selects good values under a cost; learning builds a model from data; rule-based systems apply explicit knowledge. Practical robots often combine them.

4.1 Search: explore a state space

Search algorithms take an initial state, permitted actions and a goal and explore candidate action sequences. **Uninformed** approaches such as breadth-first and depth-first search do not use a problem-specific estimate of the remaining distance. **Informed** methods, such as best-first or A*, use a heuristic to guide exploration. Full algorithms and numerical traces belong to Unit II, but their basic purpose must be clear now.

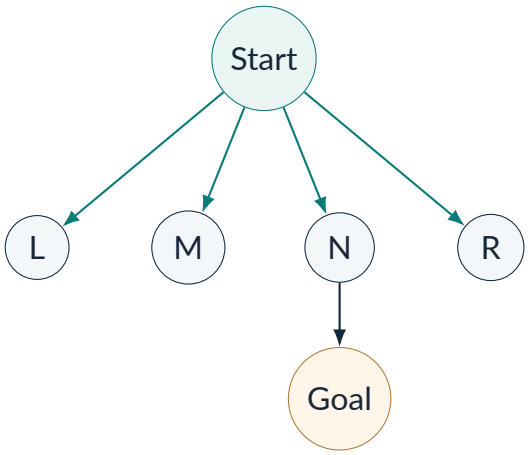

Figure 7. Conceptual frontier of a search process. A heuristic is an estimate, not proof of feasibility.

4.2 Optimization: minimize a cost or maximize utility

Optimization chooses values such as motor gains, manufacturing schedules or waypoints that best satisfy an objective while respecting limits. A candidate may be cheap but physically impossible; constraints remove such choices. Some landscapes have many local minima. General techniques include gradient-based methods when smooth models exist and metaheuristics such as genetic algorithms or simulated annealing when the structure is complex. Unit II studies metaheuristics in detail.

WORKED EXAMPLE: Weighted robot-route score

Three candidate routes have $(t, E) = (10\,\text{s}, 30\,\text{J})$, $(13\,\text{s}, 12\,\text{J})$ and $(11\,\text{s}, 20\,\text{J})$. For an illustrative objective $J = t + 0.1E$, their scores are 13, 14.2 and 13 respectively. Routes one and three tie at this weight. The decision should then use a specified tie-breaker or additional constraint rather than pretending the optimum is unique.

4.3 Learning: form a model from examples or experience

A supervised learner uses labelled input-output examples; for instance, camera images paired with “good” or “defect” labels. An unsupervised technique looks for structure in unlabelled data; for instance, clustering vibration patterns. A reinforcement-learning agent learns a policy using feedback about action outcomes, often in simulation first. These categories are not interchangeable, and real-world deployment requires distribution-shift and safety testing.

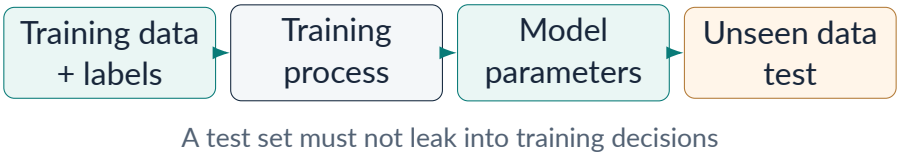

Figure 8. Training and independent evaluation are separate steps; a model must also be checked in the intended operating conditions.

4.4 Rule-based AI: reason with explicit conditions

A rule-based system stores knowledge in readable conditions and actions or conclusions. Example: “IF part is red AND destination bin is A THEN choose bin A.” Rules are easy to audit for small problems and can encode expert constraints, but large rule bases can be hard to maintain, can conflict, and may fail to generalize to new situations.

Condition	Recommended system action
Detection: high confidence	Propose the corresponding bin, subject to guard checks.
Detection: uncertain	Request inspection or route to reject bin.
Protective-zone entry	Protective control handles stop through the rated safety system.
Sensor data outdated	Reject autonomous motion command until data become valid.

4.5 Four-family comparison

Family	Useful for	Main limitation
Search	Selecting an action sequence or route	Large state space, inaccurate model or poor heuristic
Optimization	Choosing gains, schedules, paths or resource allocation	Local minima, expensive evaluation, constraint handling
Learning	Perception, fault diagnosis, model prediction	Data quality, generalization, drift, interpretability
Rule-based	Transparent policies, task sequencing, expert knowledge	Rule conflicts and difficult maintenance

4.6 Selecting an approach for a real robot

Start with the question, not the trend. A cell with fixed products and good sensors may need only a state machine and feedback controllers. A mobile robot carrying parts among many locations benefits from a route planner. A vision-driven sorter can combine learned classification, rules for rejecting uncertain

results, and a conventional robot motion controller. Search is not a substitute for motor current limits, and learning is not automatically superior to transparent threshold logic.

5 Role of AI in Manufacturing and Robotics

In simple language
Manufacturing AI is useful when the production system must recognize variation, select among multiple alternatives, estimate a future condition or replan around changing demands. Its benefit must be measured against a clear baseline.

5.1 Industrial robotic arms

An industrial robot is a reprogrammable, automatically controlled multipurpose manipulator programmable in at least three axes under the ISO/IFR industrial-robot terminology. Standard arms execute trajectories using well-established motion control; AI may add object recognition, flexible grasp selection, task planning or predictive maintenance. Reprogrammability by itself is not evidence that a robot uses machine learning.

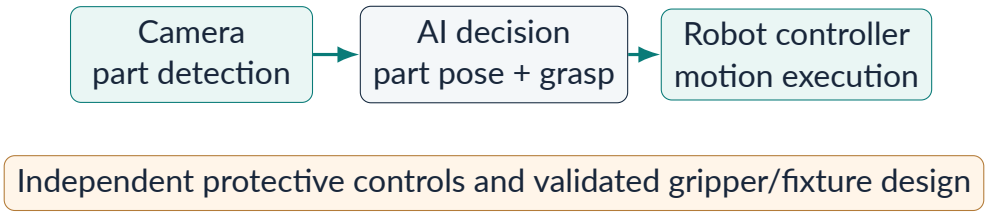

Figure 9. Machine-vision-guided picking as an AI-assisted industrial robot application.

5.2 Mobile robots

Mobile robots must estimate their pose, perceive hazards, plan motion and drive wheels or steering actuators. An autonomous mobile robot may combine geometric map-based planning with learning-based perception, while still using deterministic local feedback and monitored collision-avoidance systems. A mobile robot's performance depends on localization error, sensing, route congestion, battery and interaction with people.

5.3 Flexible manufacturing systems (FMS)

A flexible manufacturing system connects several workstations, storage and automated handling so production can change among product variants. Planning may allocate jobs, optimize operation order and route material through available stations. AI can help update schedules after disturbances, but a manufacturing execution system or dispatcher need not be machine learning.

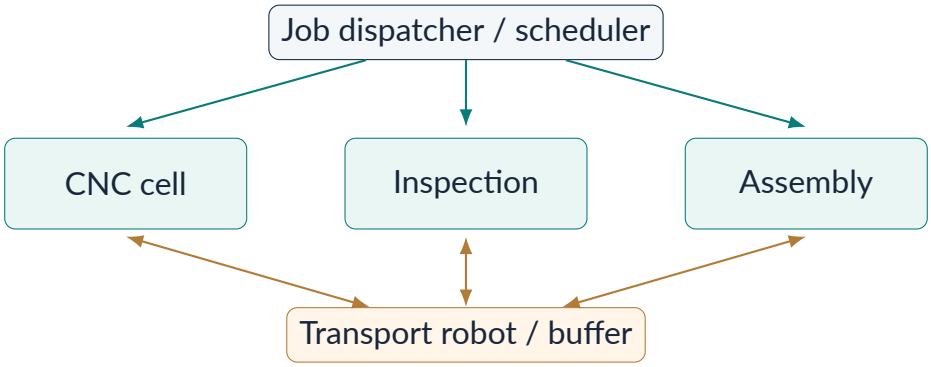

Figure 10. Conceptual FMS architecture: jobs, workstations and material handling are coordinated to support product variation.

5.4 Five realistic AI applications and what to measure

Application	Metric or review question
Visual quality inspection	Missed defects, false rejections, lighting drift, inspection rate.
Predictive maintenance	Lead time to actual failure, unnecessary maintenance alarms, data completeness.
Warehouse route planning	Time per delivery, energy, safety incidents, replanning response.
Robot grasp estimation	Pick success rate across part variants and object poses; recovery after failure.
Flexible production scheduling	Late jobs, changeover losses, machine utilization and rework.

5.5 A machine-learning classifier can be accurate and still costly

Imagine 100 parts: 20 defective and 80 good. A classifier identifies 18 defective parts correctly, misses 2, and wrongly rejects 4 good parts. Its precision for predicted defect is $18/(18+4) = 81.8\%$; recall for actual defects is $18/(18+2) = 90\%$. A factory might value missed defects and false rejects very differently. Accuracy alone can hide the business and safety cost. Detailed model evaluation is a later-unit topic; this simple example explains why performance measures must be chosen carefully.

5.6 People, reliability and safety

AI outputs should be accompanied by confidence/quality checks, fault detection, version control and monitored transitions into safe modes. A model trained under one camera or material condition may behave differently after lighting, lens or part suppliers change. A robot system needs conventional protective controls, defined operating limits, audit logs and an accountable verification process. An AI decision cannot replace a risk-assessed machine safety circuit.

6 Integrated Applications and Worked Case Studies

6.1 Case A: warehouse robot selects a delivery route

Task: carry a part from loading dock to shelf zone. **Inputs:** destination, map, battery estimate, obstacle readings and task priority. **Decision:** use planning to select a permitted path and replan when blocked. **Execution:** motor controllers convert planned motion into wheel commands. **Evaluation:** time, travel length, energy, number of interventions and no-go-zone violations. The planner's shortest path is worthless if its localization data are stale or an obstacle moves into the corridor.

6.2 Case B: AI-assisted visual quality sorting

Task: separate parts by defect category. **Inputs:** calibrated camera image, conveyor encoder and part tracking. **Decision:** a classifier predicts type and con-

fidence. **Rules:** low-confidence parts go for secondary inspection; faults stop automatic sorting. **Execution:** a controller times the gripper or diverter. **Evaluation:** false reject, missed defect, latency and throughput. The model is only one element of the complete automation sequence.

6.3 Case C: flexible cell schedules two machines

Two jobs, A and B, can run on machines M1 and M2 with the following illustrative processing times: A takes 4 min on M1 or 6 min on M2; B takes 5 min on M1 or 3 min on M2. If each machine works on one job and both jobs start at time zero, assigning A to M1 and B to M2 completes in $\max(4, 3) = 4$ min, while the swapped assignment completes in $\max(6, 5) = 6$ min. This toy example highlights scheduling as an optimization problem; real FMS scheduling also includes setup, transport, precedence, queueing and breakdowns.

6.4 Case D: inspection system faces data drift

A camera model was trained on metal parts under uniform light. New glossy parts cause reflections. An image classifier may still generate confident but wrong labels. Useful monitoring compares the incoming image distribution and sampled human-inspected outcomes with validated baselines. A deployment policy should require review before automatic threshold changes; a simple "retrain weekly" rule is not evidence of safety or correctness.

6.5 Choosing a hybrid architecture

Many useful systems combine a **learned perception module**, a **symbolic rule or search planner**, and a **conventional real-time controller**. Keeping the module boundaries explicit improves debugging. If a wrong object is picked, engineers can ask whether perception misclassified it, planning selected the wrong grasp, coordinate transforms were wrong, or motion control failed.

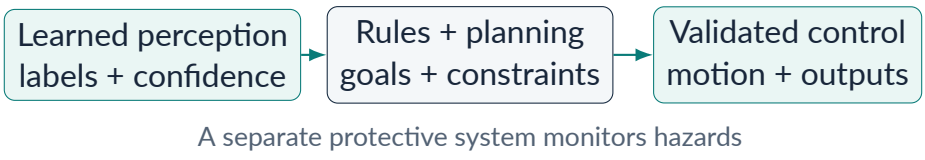

Figure 11. Hybrid intelligent architecture. Interfaces allow each module to be tested independently.

7 Exam Workshop, Glossary and Revision Questions

7.1 Six core definitions to reproduce in your answer

1. **AI:** methods for creating systems that can interpret, reason, learn or select actions using computational models.
2. **Automation:** execution of processes with less continuous human intervention.
3. **Robotics:** engineering of embodied machines that interact physically using sensing, computation and actuation.
4. **Agent:** a decision-making system that perceives an environment and takes actions.
5. **State space:** the set of modelled possible states and their relationships through actions.
6. **PEAS:** performance measure, environment, actuators and sensors—an agent-task specification.

7.2 A structured five-mark answer

For the question *"Explain intelligent agents in robotics"*: (1) define agent and environment, (2) draw a sensor-to-decision-to-actuator loop, (3) give a PEAS example, (4) explain a simple reflex and goal-based policy, and (5) mention rationality plus partial observability. This combines description, figure and engineering application rather than only memorising terms.

7.3 Common misconceptions

Misconception	Correct interpretation
Every robot uses machine learning	A programmed industrial robot can work without learned decision models.
Automation and AI are the same	Automation is about executing processes; AI refers to selected modelling/decision techniques.
A rational agent always succeeds	It chooses well given its information and objective, but may still face noise and error.
A search path is a motor trajectory	Graph planning does not ensure dynamic feasibility or safe execution.
A model's training score proves quality	Unseen operating conditions, drift and failure cost require separate tests.
A PEAS table is just a sensor list	It must include performance, environment and actuator capability.

7.4 Unit I question bank (syllabus-derived practice, not claimed PYQs)

1. Explain AI, automation and robotics with two examples of their overlap.
2. Draw the architecture of an intelligent robotic agent.
3. Define PEAS; construct PEAS for a robot vacuum or warehouse AMR.
4. Differentiate simple reflex, model-based, goal-based, utility-based and learning agents.
5. Explain the meaning of a partially observable and dynamic robot environment.
6. Describe the five elements of a state-space search problem.
7. Formulate robot navigation as a search problem, including a goal and constraints.
8. Contrast state graphs with search trees using a simple diagram.
9. Compare search, optimization, learning and rule-based approaches.

10. Explain how AI adds capability to an industrial robot without replacing low-level control.
11. Describe AI applications in a flexible manufacturing system.
12. Explain a mobile robot's perception–planning–action pipeline.
13. Propose performance measures for visual inspection and fault prediction.
14. Explain why confidence scores, drift monitoring and human review matter in factory AI.

7.5 Short answer checkpoints

Exercise	Answer
State combinations	$20 \times 20 \times 4 \times 2 \times 10 = 32{,}000$ nominal combinations.
Graph path cost	$S \to A \to G = 5$; $S \to B \to G = 3$.
Utility example	Route A score 110; route B score 121; A preferred for the given weight.
Factory scheduling	A on M1, B on M2 finishes in 4 min, better than 6 min swapped.
Defect detection	Precision $18/22 = 81.8\%$; recall $18/20 = 90\%$.

One-minute recap
Specify the task and performance measure. Write PEAS. Represent the relevant state and allowed actions. Choose a suitable AI technique. Connect its output to tested motion control and safety systems. Evaluate with unseen conditions, meaningful error costs and monitoring. This sequence explains AI in robotics more accurately than naming fashionable algorithms alone.

Source and diagram ledger

Official syllabus: SPPU, TE Robotics and Automation, PCC301-ROA *Artificial Intelligence for Robotics*, Unit I, 2024 Pattern; supplied *TE Syllabus.pdf*, printed page 14. The Unit I topic list on the cover is syllabus-derived.

Technical and academic references consulted:

1. UC Berkeley, *CS188 Introduction to Artificial Intelligence*, course topics covering search, planning and AI concepts. https://inst.eecs.berkeley.edu/~cs188/
2. University of London, *Artificial Intelligence study guide*, chapter on intelligent agents and task environments, including Russell–Norvig learning references. https://www.london.ac.uk/sites/default/files/study-guides/artificial-intelligence.pdf
3. International Federation of Robotics, *Industrial Robots*, ISO 8373-based industrial robot vocabulary. https://ifr.org/industrial-robots
4. NIST, *Intelligent Flexible Manufacturing Cell* (2023). https://www.nist.gov/publications/intelligent-flexible-manufacturing-cell
5. NIST, *2026 Roadmap on Artificial Intelligence and Machine Learning for Smart Manufacturing*. https://www.nist.gov/publications/2026-roadmap-artificial-intelligence-and-machine-learning-smart-manufacturing

Diagram provenance: All diagrams in this learning candidate are original, editable TikZ figures. No external copyrighted textbook diagram was copied; web photographs were not embedded because the raw licensed images could not be reliably downloaded for verification in this environment. These figures are conceptual rather than safety-rated machine schematics.

Validation limitation: This is a complete Unit I *review candidate*; it is not a certified engineering text, a record of past examination questions, or evidence of physically tested robot behaviour. Exact equipment, safety and performance claims need device-specific verification.

UNIT I | VERIFIED EXAM SUPPORT

Foundations of Artificial Intelligence for Robotics

Supplement to the original 23-page teaching edition. It preserves the original pages unchanged and adds exam-focused answers, question-bank traceability, and transparent verification limits.

SOURCE BOUNDARY The 2024 Pattern SPPU syllabus specifies Unit I foundations: AI vs automation vs robotics, agents, formulation, state spaces, four technique families, and robotic/manufacturing applications. Question-bank examples below are supplementary practice, not newly mandated syllabus topics.

A. Fifteen-question traceability map

Bank question	Where to find teaching content / answer cue
Q1 AI vs automation vs robotics	Source pp. 3-5; define each and use conveyor / robot arm / vision classifier.
Q2 Problem formulation	Source pp. 10-12; initial state, actions, transitions, goal test, cost.
Q3 Agent components	Source p. 6; sensors/percepts, decision policy, actuators, environment.
Q4 Five agent types	Source p. 8; simple-reflex, model-based, goal-based, utility-based, learning.
Q5 AI technique families	Source pp. 12-16; search, optimization, learning, explicit rules.
Q6 State space	Source pp. 10-12; state variables, transitions and reachable states.
Q7 Manufacturing AI	Source pp. 5, 16-19; inspection, scheduling, maintenance.
Q8 Industrial/mobile/FMS	Source pp. 16-19; each application and its limitations.
Q9 State space importance	Source pp. 10-13; graph vs search tree.
Q10 Four AI techniques	Source pp. 12-16; compare capability and failure mode.
Q11 Predictive/process/energy	Source pp. 5, 17-18; distinguish example from validated improvement.
Q12 Three application types	Source pp. 16-19; industrial arm, warehouse AMR, FMS.
Q13 Agent vs fixed automation	Source pp. 3-8; sensing, policy, action and feedback.
Q14 PEAS warehouse robot	Source pp. 6-7; performance, environment, actuators, sensors.
Q15 Benefits and challenges	Source pp. 16-19, 21; generalization, safety, cost, integration.

EXAM-READY CONCEPT ANSWERS

B. Explain five intelligent-agent types (Q4)

Type	Working principle and example
Simple reflex	Uses current percept and IF-THEN rule; stop a conveyor when a proximity input is active.
Model-based reflex	Keeps internal state to handle missing observations; an AMR remembers the last known obstacle position.
Goal-based	Selects actions to reach a target goal; robot chooses a route to a delivery shelf.
Utility-based	Scores alternatives by defined preferences/cost; choose an energy/time-balanced route.
Learning	Improves a model or policy from experience; image-based part classifier updated from labelled examples.

Diagram to draw: Environment -> Sensors/percepts -> Agent state and decision policy -> Actuators -> Environment. State that rational action depends on available evidence and the chosen performance measure.

C. Formulate an autonomous warehouse navigation task (Q2 / Q6)

Initial state: robot at (1,1), heading east, carrying a parcel. **State:** (x, y, heading, load state) with optional battery level. **Actions:** permitted forward step, turn left/right, pickup/drop; prohibit obstacle collisions. **Transition model:** maps each legal state and action to its next state. **Goal test:** parcel delivered at the target shelf and delivery confirmed. **Path cost:** sum of distance, travel time or a dimensionally consistent weighted objective. **Constraints:** no-go zones, battery minimum, turning radius. A graph search gives an abstract route, not an already safety-validated motor trajectory.

D. PEAS of warehouse robot (Q14)

PEAS	Warehouse AMR specification
Performance	Correct on-time delivery, collision avoidance, energy efficiency, low damage and recovery delay.
Environment	Shelves, aisle map, staff, forklifts, temporary obstacles, floor, charging point.
Actuators	Drive motors, brakes, gripper/lift, light and speaker.
Sensors	Lidar/camera, encoder, IMU, battery monitor, safety scanner and gripper sensor.

CHECKED NUMERICALS AND RELEASE GATES

E. Reproduce key calculations from the source

Source topic	Verified calculation
State-space combinations (p. 11)	20 x 20 x 4 x 2 x 10 = 32,000 nominal combinations.
Weighted graph (pp. 11-12)	S-A-G: 1+4=5; S-B-G: 2+1=3. The second path is cheaper.
Agent performance score (p. 9)	J=t+(0.10 s/Joule)E; route A: 90+20=110 s; route B: 105+16=121 s.
Optimization comparison (p. 14)	Routes (10,30),(13,12),(11,20): 13, 14.2, 13 s. Routes 1 and 3 tie.
Part-inspection results (p. 18)	TP=18, FN=2, FP=4, TN=76: precision=18/22=81.82%; recall=18/20=90%; accuracy=94%.
Two-machine schedule (p. 19)	A->M1, B->M2: max(4,3)=4 min; swapped: max(6,5)=6 min.

F. A short, reproducible five-mark response template

Definition (1 mark): Define the named concept in one or two lines. **Structure (1 mark):** Label a correct agent, graph or manufacturing pipeline diagram. **Working (1 mark):** State the decisions, state representation, or step sequence. **Application (1 mark):** Supply a concrete industrial-arm, AMR or FMS example. **Limitation (1 mark):** Explain uncertainty, data drift, physical feasibility or need for a separate protective system. Actual marking is set by the examiner.

G. QA and publication decision

Verified: 23 source pages and 11 figure captions inspected by page contact-sheet screening; six numerical checks reproduced; all syllabus headings and Unit I question-bank prompts mapped to source passages. Original pages unchanged. The supplement is explicitly educational, not manufacturer validation.

Remaining: enlarged figure-by-figure readability check, manual proofreading of every line, source URL/link validation, accessibility tagging and academic sign-off. The v2 file is a review candidate, NOT publication approved.

Source authority: SPPU TE Robotics and Automation Engineering 2024 Pattern PCC301-ROA, printed syllabus p. 14; AIFR Unit-Wise Question Bank, Unit I Q1-Q15; Zenith AI Unit 01 Foundations v1, pp. 1-23.