ROBOTICS

FUNDAMENTALS OF ROBOTICS

Sections

Section 1: Introduction to Robotics

1.1 What is Robotics?

Robotics is an interdisciplinary branch of engineering and science that integrates mechanical engineering, electrical and electronics engineering, computer science, and control systems to design, construct, operate, and apply robots.

A robot is a programmable electromechanical system capable of sensing its environment, processing data, and performing actions to achieve a goal, either autonomously or under human supervision.

Key Concept:

Robotics = Mechanics (body) + Electronics (nerves) + Computing (brain)

1.2 Why Robotics?

Robots are designed to:

- Perform repetitive, dangerous, or precise tasks with consistency.
- Operate in environments unsafe for humans (nuclear, deep sea, space).
- Increase productivity and efficiency in industries.
- Assist in healthcare, education, and exploration.

Examples:

- Industrial robot arms assembling cars.
- Surgical robots performing delicate operations.
- Autonomous drones for surveillance or mapping.
- Educational robots like Otto or LEGO Mindstorms for learning.

1.3 Characteristics of a Robot

Characteristic	Description
Autonomy	Performs actions with minimal or no human intervention.
Sensing	Perceives environment through sensors.
Reprogrammability	Can be programmed for multiple tasks.
Manipulation	Physically interacts with environment via actuators.
Intelligence	Processes information and makes decisions.

1.4 Basic Components of a Robot

1. Mechanical System – body, joints, links, structure
2. Sensors – gather data (IR, ultrasonic, camera)
3. Actuators – create motion (motors, servos, pneumatics)
4. Controller – processes sensor data, generates commands (microcontroller, processor)
5. Power Source – battery or supply unit
6. Software/Algorithm – defines robot behavior and logic

Block Diagram:

```
[Sensors] → [Controller] → [Actuators]
     ↑                    ↓
 Environment ←--- Feedback ----
```

Workshop Note : Identify Components

🧰 *Objective:*
Familiarize with the parts that make up a simple robot.

Activity:

1. Observe an educational robot (e.g., line follower or Otto biped).
2. Identify:
    - Sensors (input devices)
    - Actuators (motors, servos)
    - Controller (Arduino, Raspberry Pi, etc.)
    - Power supply
3. Draw its block diagram in your notebook.

Section 2: History and Evolution of Robotics

2.1 Early Concepts

The idea of robots is ancient. Myths, stories, and designs for artificial beings date back centuries:

- Ancient Greece: The word *robot* originates indirectly from Greek "automaton" meaning self-moving.
- 15th Century: Leonardo da Vinci designed a mechanical knight (1495).
- 18th Century: Clockwork automatons could play music or write.

2.2 Modern Robotics Evolution

Year/Period	Development	Description
1921	Term "Robot" introduced	From play *R.U.R.* (Rossum's Universal Robots) by Karel Čapek.
1954	First industrial robot concept	George Devol designed *Unimate*, first programmable manipulator.
1960s	Industrial use begins	Unimate installed in General Motors factory for welding.
1970s–80s	Robotics research expands	Use in medicine, space, and autonomous control.
1990s–2000s	Mobile & humanoid robots	Sony AIBO, Honda ASIMO, educational robots emerge.
2010s–2020s	AI & collaborative robots (Cobots)	Integration with vision, ML, and cloud computing.

2.3 Generations of Robots

Generation	Key Features
1st Generation	Fixed sequence robots (hard-wired movements)
2nd Generation	Variable sequence, programmable robots
3rd Generation	Sensor-based, adaptive robots
4th Generation	Intelligent robots with AI, computer vision
5th Generation	Collaborative & learning robots, cloud-connected systems

---

Workshop Note : Robotics Timeline

Objective:
Understand the evolution of robotics through visual mapping.

Activity:
Create a timeline poster showing the major robotic milestones:

- First robot (Unimate)
- Industrial robot arms
- Humanoid robots (ASIMO, Atlas)
- Modern AI-driven robots (Boston Dynamics, Tesla Bot)

Section 3: Definition and Classification of Robots

3.1 Definition by Standards

According to ISO 8373:

"A robot is an actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment to perform intended tasks."

3.2 Classification of Robots

Robots can be classified based on structure, motion, autonomy, and application.

3.2.1 Based on Mechanical Structure

Type	Description	Example	Workspace
Cartesian Robot	Linear motion along X, Y, Z axes	CNC machine	Cuboidal
Cylindrical Robot	Rotates about base + vertical motion	Loading robots	Cylindrical
SCARA Robot	Selective Compliance Assembly Robot Arm; horizontal flexibility	PCB assembly robot	Planar
Articulated Robot	Multiple rotary joints, like human arm	Industrial arm (ABB, KUKA)	Spherical
Delta Robot	Parallel linkage, high-speed pick & place	Packaging robot	Dome-shaped
Polar Robot	Spherical coordinates	Welding, painting	Spherical

---

3.2.2 Based on Degree of Autonomy

Type	Description	Example
Manual	Controlled directly by human (wired)	Teleoperated bomb disposal unit
Semi-Autonomous	Human supervision + partial control	Surgical robot
Fully Autonomous	Operates independently	Self-driving car, drone

---

3.2.3 Based on Application

Application	Example
Industrial Robots	Welding, assembly, painting
Service Robots	Cleaning, delivery, surgery
Mobile Robots	Drones, autonomous vehicles
Humanoid Robots	ASIMO, Atlas
Educational Robots	Otto DIY, LEGO EV3
Space Robots	Mars rover, Canadarm
Military Robots	Bomb disposal, surveillance drones

Workshop Note : Robot Classification

Objective:
Visually classify and compare different robots.

Activity:

1. Watch short clips or images of robots performing tasks (industrial, humanoid, mobile).
2. Classify each robot under:
   - Structure type
   - Level of autonomy
   - Application
3. Record observations in a table.

3.3 Characteristics Comparison

Feature	Industrial Robot	Mobile Robot	Humanoid Robot
Structure	Fixed base, articulated arm	Wheeled/legged platform	Bipedal body
Mobility	Limited (fixed)	High	High
Precision	Very high	Moderate	Moderate
Application	Assembly, manufacturing	Delivery, mapping	Interaction, learning
Example	FANUC Arm	Autonomous Rover	ASIMO, Pepper

Workshop Reflection

- What makes a robot intelligent?
- Can a mobile robot be industrial?
- What features make humanoid robots closer to humans?

Section 4 : Robot Components and Subsystems

Every robot—no matter how simple or advanced—is built from five fundamental subsystems.
These together form the "Robot System Architecture."

```
+------------------------+
| CONTROL SYSTEM         | ← Brain
+------------------------+
| SENSOR SYSTEM          | ← Senses
+------------------------+
| ACTUATOR SYSTEM        | ← Muscles
+------------------------+
| POWER SUPPLY           | ← Energy Source
+------------------------+
| MECHANICAL STRUCTURE   | ← Body
+------------------------+
```

4.1 Mechanical Structure

- Framework of links, joints, gears, chassis, or frames.
- Determines the workspace, payload, and stability.
- Materials: aluminium, carbon fibre, ABS plastic, steel.

Examples:

- Robot arm → Links & rotary joints.
- Mobile robot → Chassis + wheel assembly.

4.2 Actuator System

Actuators are the muscles of a robot—they convert energy into motion.

Type	Energy Used	Example Use	Remarks
Electric Motors (DC, Stepper, Servo)	Electrical	Most mobile/arm robots	Precise, easy control
Hydraulic Actuators	Pressurised fluid	Heavy industrial robots	High power, complex
Pneumatic Actuators	Compressed air	Pick-and-place robots	Fast, less accurate
Shape Memory Alloys / Piezo	Thermal / Crystal	Micro-robots	Small motion, experimental

---

4.3 Sensor System

Sensors give feedback—the ability to sense position, distance, orientation, etc.

Type	Example Sensors	Purpose
Position / Motion	Encoders, potentiometers	Joint angles, wheel rotation
Proximity / Range	Ultrasonic, IR, LiDAR	Detect obstacles
Vision	Camera, depth sensors	Object recognition

Force / Torque	Strain gauge	Gripping control
Environmental	Temperature, gas, light	Adapt to surroundings

4.4 Control System

The brain that interprets sensor data and commands actuators.

- Implements algorithms for movement and decision-making.
- Often built using embedded systems (microcontrollers, microprocessors, or SBCs like Raspberry Pi).
- Uses control loops (open-loop or closed-loop).

4.5 Power Supply

- Provides energy to all components.
- Types: Li-ion batteries, DC supply, pneumatic compressors, hydraulic pumps.
- Must meet current & voltage requirements safely.

4.6 End Effector

- The “tool” attached to the robot arm’s end for performing tasks.
- Examples: gripper, welder, suction cup, camera mount, scalpel, 3D printer nozzle.

Section 5 : Robotic Arm Fundamentals

5.1 Links and Joints

A robot manipulator consists of:

- Links → Rigid bodies connecting joints.
- Joints → Allow relative motion between links.

Joint Type	Motion	Example
Revolute (R)	Rotation	Servo joint
Prismatic (P)	Linear	Telescopic arm
Spherical (S)	3-axis rotation	Humanoid shoulder

5.2 Degrees of Freedom (DoF)

Definition:
The number of independent motions a robot (or any mechanical system) can perform.

- Each revolute or prismatic joint adds one DoF.
- A human arm ≈ 7 DoF (3 shoulder + 1 elbow + 3 wrist).

Example	No. of DoF	Motions Possible
SCARA arm	4	X, Y, Z + rotation
6-Axis arm	6	Full 3D manipulation
Mobile robot	3	X, Y, orientation (θ)

Formula (for serial manipulators):

$$\text{Total DoF} = \sum_{i=1}^{n} f_i$$

where f_i = DoF contributed by each joint.

5.3 Workspace

Workspace (or *Reach Envelope*) is the total region a robot's end-effector can reach.

- Depends on link lengths and joint limits.
- For an arm: usually a *hemispherical* or *cylindrical* region.

Diagram (Conceptual):

```
        ↑ Z
       /|
      / |
     /  |    <- Reach limit
------O---→ X
    /
   /
  ↓ Y
(Shaded area = workspace)
```

Types:

1. Reachable workspace – all possible positions the arm can extend to.
2. Dexterous workspace – subset where the tool can orient freely.

5.4 Kinematic Chain

A series of links and joints forming a manipulator.

- Open chain: free end effector (typical arm).
- Closed chain: links form a loop (Delta, parallel robots).

Forward Kinematics:
Calculates end-effector position from joint angles.
Inverse Kinematics:
Finds joint angles required for desired position.

5.5 Coordinate Systems

- Joint space: Angles or positions of joints (θ_1, θ_2 …).
- Task space / Cartesian space: X-Y-Z coordinates of the end-effector.

Transformations between these are done using homogeneous matrices or Denavit-Hartenberg (DH) parameters in advanced robotics.

Section 6 : Embedded Systems in Robotics

6.1 What Is an Embedded System?

An embedded system is a dedicated computing system designed to perform a specific control function within a larger mechanical or electrical system.

Feature	Description
Hardware	Microcontroller + memory + I/O interfaces
Software	Embedded firmware (C, MicroPython, etc.)
Real-time	Must respond to events quickly and deterministically
Integration	Tight coupling of hardware, sensors, and actuators

Examples:

- Arduino controlling line follower motors
- STM32 running PID for robotic arm
- ESP32 streaming sensor data to web UI

6.2 Why Embedded Systems in Robotics?

Reason	Explanation
Real-Time Control	Robot decisions must be immediate (motor speed, sensor feedback).
Compact & Low-Power	Microcontrollers are small and efficient.
Direct Hardware Access	Can read sensors, drive motors, generate PWM, etc.
Customizability	Reprogrammable for any robotic task.

6.3 Robotics vs Embedded Systems

Aspect	Robotics	Embedded Systems
Focus	Mechanical motion, autonomy, and control integration	Real-time data processing & device control
Scope	Multidisciplinary: mechanical + electrical + software	Primarily electronics & firmware
Hardware	Motors, sensors, chassis, controllers	Microcontrollers, timers, ADC, communication
Output	Physical movement / task execution	Logical or electrical signals
Example	A line-follower robot	The Arduino board inside it

Conclusion:
👉 Robotics is the system, while the embedded controller is its nervous system.

6.4 System Approach to Building a Robot

To design a robot systematically, follow the Robot Development Lifecycle:

Step	Description
1. Define Objective	What task should the robot perform?
2. Mechanical Design	Choose configuration (SCARA, wheeled, arm, biped).
3. Select Actuators & Sensors	Based on precision, torque, and sensing needs.
4. Choose Controller Platform	Arduino, ESP32, STM32, Raspberry Pi, etc.
5. Develop Control Algorithm	Open/closed loop, PID, path-planning.
6. Integration & Testing	Connect, calibrate, debug hardware.
7. Optimization	Tune performance and efficiency.

Section 7 : Open-Loop and Closed-Loop Control Systems in Robotics

Control systems are the decision-making core of a robot. They determine how sensor data influences motion and how accurately the robot performs tasks.

7.1 Open-Loop Control System

Definition:
A system where the output is not measured or fed back to the controller.

Example:
You set a motor to rotate for 5 seconds — it moves, but if the wheel slips, the controller doesn't know.

Diagram (conceptual):

[Controller] → [Actuator] → [Plant/Robot Arm] → [Output]

(No feedback path)

Advantages:

- Simple and low-cost
- Fast, less computation

Disadvantages:

- No error correction
- Inaccurate under variable load or environment

Real-world Example:
Basic line follower robot without sensors, using fixed motor speeds.

7.2 Closed-Loop Control System

Definition:
A system where the output is continuously measured, compared with the desired value, and corrections are made automatically.

Diagram (conceptual):

[Controller] → [Actuator] → [Plant/Robot Arm] → [Sensor Feedback] ↺

Advantages:

- High accuracy and stability
- Compensates for disturbances

Disadvantages:

- More complex, requires sensors and processing

Real-world Example:
A self-balancing robot or servo-controlled robotic arm maintaining precise positions.

7.3 Mathematical Model (Basic Feedback Loop)

$$E(t) = R(t) - C(t)$$
$$C(t) = G(s)H(s)E(t)$$

where:

- $R(t)$: Reference input (desired value)
- $C(t)$: Controlled output
- $E(t)$: Error signal
- $G(s)$: Forward gain (controller + plant)
- $H(s)$: Feedback path gain

This feedback concept is the foundation for PID (Proportional–Integral–Derivative) controllers in robotics.

Workshop Note:

1. Run a DC motor with fixed delay (open-loop).
2. Add an encoder to measure speed (closed-loop).
3. Observe how speed regulation improves.

Real-World Robots and Case Studies

8.1 Industrial Robots

Company	Robot	Features
FANUC	LR Mate 200iD	6-axis articulated arm for assembly
ABB	IRB 1200	Compact industrial manipulator
KUKA	KR AGILUS	High-speed packaging and welding robot
Yaskawa	Motoman GP12	Precision handling robot
Universal Robots (UR)	UR5, UR10	Collaborative robot (cobot) – works safely near humans

Applications: welding, painting, material handling, packaging, CNC loading/unloading.

8.2 Service and Research Robots

Type	Example	Application
Humanoid	Honda ASIMO	Walking, assistance
Educational	Otto DIY, Lego Mindstorms	STEM learning
Domestic	iRobot Roomba	Home cleaning
Medical	Da Vinci Surgical System	Remote surgery
Space	NASA Curiosity Rover	Mars exploration

8.3 Modern Trends

1. Collaborative Robots (Cobots): Safe, human-friendly arms.
2. Swarm Robotics: Many small robots cooperating (inspired by ants).
3. Soft Robotics: Flexible materials for delicate tasks.
4. AI + Vision Integration: Object recognition and path planning.
5. Edge Computing in Robotics: Real-time decision-making onboard.

Section 9 : Workshop Activities, Review, and Summary

9.1 Workshop 1: Identify Robot Subsystems

Objective: Recognize all five major subsystems of a robot.
Materials: Small robot kit (or images).

Subsystem	Example Component
Mechanical	Chassis, frame, arm links
Actuator	DC motor, servo
Sensor	IR sensor, ultrasonic
Controller	Arduino, STM32
Power	Battery pack

9.2 Workshop 2: Build a Conceptual Robot

Task:
Design (on paper or CAD) a robot that can pick an object and place it 50 cm away.

Steps:

1. Define task (Pick-and-place).
2. Choose robot type (SCARA or 4-DOF arm).
3. Select actuators (Servos).
4. Select sensors (IR or limit switch).
5. Choose controller (Arduino Uno).
6. Draw block diagram showing signal flow.

9.3 Workshop 3: Mini Simulation

Use Tinkercad Circuits or Proteus to:

- Connect an IR sensor and DC motor to Arduino.
- Program the motor to start when IR detects an obstacle.
- Observe open-loop vs closed-loop behaviour.

9.4 Self-Study Questions

1. Define a robot and explain its key characteristics.
2. List and explain five major robot subsystems.
3. Differentiate between SCARA and Cartesian robots.
4. What is a degree of freedom (DoF)?
5. Explain the difference between open-loop and closed-loop control with examples.
6. How do embedded systems contribute to robotics?
7. Describe the workspace of a robotic arm.
8. What is the significance of feedback in control systems?

9.5 Summary Table: Day 1 Concepts

Concept	Key Points
Robot Definition	Programmable mechanical system with sensing, computation, and action.
Classification	By structure, autonomy, motion, and application.
Subsystems	Mechanical, actuator, sensor, controller, power.
DoF	Independent motion axes.
Workspace	Reachable region of the robot arm.
Embedded Role	Real-time control and signal processing.
Control Systems	Open vs closed loop, feedback mechanisms.
Design Approach	Define task → Design → Select components → Control → Test.

9.6 Quick Quiz

Question	Options	Answer
The first industrial robot was ___	a) ASIMO b) Unimate c) PUMA	b)
SCARA robot is mainly used for ___	a) Painting b) Assembly	b)
The brain of the robot is the ___	a) Controller b) Actuator	a)
Degree of freedom means ___	a) Number of motors b) Independent motions	b)
Closed-loop system uses ___	a) Feedback b) Timer	a)

9.7 Reflection & Discussion

- What mechanical system in nature inspires robotics design the most?
- Discuss why feedback and sensors are essential for autonomy.
- What could go wrong if we use open-loop control in a humanoid robot?
