OR					
11	a)	Discuss how MATLAB Robotics Toolbox supports kinematics and path planning.	L2	CO5	5 M
	b)	Explain the concept of human–robot collaboration with suitable examples.	L2	CO5	5 M

Code: 23ME6501

III B.Tech - I Semester - Honors Examinations - NOVEMBER 2025

INDUSTRIAL ROBOTICS AND AUTOMATION
(HONORS in MECHANICAL ENGINEERING)

Duration: 3 hours   Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level   CO – Course Outcome

PART – A

		BL	CO
1.a)	Differentiate between fixed and programmable automation.	L2	CO1
1.b)	What is meant by collaborative robots (cobots)?	L2	CO1
1.c)	What is a homogeneous transformation matrix?	L2	CO2
1.d)	Define forward kinematics.	L2	CO2
1.e)	List any two differences between Lagrangian and Newton–Euler methods.	L2	CO3
1.f)	Define trajectory planning.	L2	CO3
1.g)	Write any two applications of vision sensors in robotics.	L2	CO4
1.h)	What is ladder logic?	L2	CO4
1.i)	Give two advantages of offline programming.	L2	CO5
1.j)	Define teach pendant programming.	L2	CO5

PART – B

			BL	CO	Max. Marks
		UNIT-I			
2	a)	What is the work envelope of a robot? Sketch and explain two views to indicate the work envelope of a cylindrical robot.	L2	CO1	5 M
	b)	Describe the various considerations taken into account while using robots in material handling applications.	L2	CO1	5 M
		OR			
3	a)	Describe the function of the four basic components of robotic systems.	L2	CO1	5 M
	b)	Evaluate the benefits and limitations of using robots in automobile assembly lines.	L2	CO1	5 M
		UNIT-II			
4	a)	Explain the difference between forward kinematics and inverse kinematics.	L2	CO2	5 M
	b)	Determine the rotation matrix for a rotation of 45^0 about the y-axis, followed by a rotation of 120^0 about the z-axis and a final rotation of 90^0 about the x-axis.	L2	CO2	5 M
		OR			
5		For a 2-link planar manipulator with link lengths L1 and L2, formulate the DH parameters and transformation matrices.	L3	CO2	10 M

		UNIT-III			
6		Derive the dynamic equation of a simple 2-link planar robot using Newton–Euler formulation.	L3	CO3	10 M
		OR			
7	a)	Differentiate between Joint space trajectory planning and Cartesian space trajectory planning.	L3	CO3	5 M
	b)	Discuss the concept of compliance in robotic systems with an example.	L3	CO3	5 M
		UNIT-IV			
8	a)	Explain the working principle of a stepper motor and its advantages in robotics.	L2	CO4	5 M
	b)	Discuss in detail the working of the Position Encoders in robots.	L2	CO4	5 M
		OR			
9	a)	Write a simple ladder logic program to control a motor using start and stop push buttons.	L2	CO4	5 M
	b)	Differentiate between Modbus and Profibus communication protocols.	L2	CO4	5 M
		UNIT-V			
10	a)	Explain the advantages and disadvantages of offline programming.	L2	CO5	5 M
	b)	Write short notes on ABB RAPID language and its key features.	L2	CO5	5 M

Code: 23ME5501

PVP 23

III B.Tech - I Semester - Honors Examinations - NOVEMBER 2025
INDUSTRIAL ROBOTICS AND AUTOMATION
(HONORS IN MECHANICAL ENGINEERING)

Duration: 3 hours **Max. Marks: 70**

PART – A

1\. a) Fixed automation - 1M
Programmable automation -1M
1\. b) Meaning of collaborative robots (Cobots) - 2M
1\. c) Representation of 4×4 homogeneous transformation matrix- 2M
1\. d) Definition of forward kinematics - 2M
1\. e) Any two differences between Lagrangian and Newton–Euler methods- 2M
1\. f) Definition of trajectory planning- 2M
1\. g) Any two major applications of vision systems in robotics- 2M
1\. h) Definition of ladder logic - 2M
1\. i) Any two advantages of offline programming- 2M
1.j) Definition of teach pendant programming- 2M

PART – B

2 a) What is the work envelope of a robot? Sketch and explain two views to indicate the work envelope of a cylindrical robot. **[5M]**

- Definition of work envelope of a robot- 1M
- FIGURE- 2M
- Explain two views-2M

2 b) Describe the various considerations taken into account while using robots in material handling applications. **[5M]**

- FIGURE- 2M
- considerations taken into account while using robots in material handling applications-3M

3 a) Describe the function of the four basic components of robotic systems. **[5M]**

- FIGURE- 1M
- Four basic components of robotic systems- 4*1=4M

3 b) Evaluate the benefits and limitations of using robots in automobile assembly lines.

- Benefits of using robots in automobile assembly lines -2.5M
- Limitations of using robots in automobile assembly lines -2.5M

4 a) Explain the difference between forward kinematics and inverse kinematics. **[5M]**

- Forward Kinematics (FK): 2.5M
- Inverse Kinematics (IK): 2.5M

4b

- Rotation of 45^0 about z-axis- 1M
- Rotation of 120^0 about y-axis-1M
- Rotation of 90^0 about x-axis- 1M
- Final Rotation Matrix - 2M

5)

- 2-Link planar manipulator figure- 2M
- D-H parameter Table- 3M
- Link 1 – 2M
- Link 1 – 2M
- Final Answer- 1M

6) Derive the dynamic equation of a simple 2-link planar robot using Newton–Euler formulation. **10M**

- Step 1- 2M
- Step 2- 2M
- Step 3- 2M
- Step 4- 2M
- Step 5- 2M

7 a) Differentiate between Joint space trajectory planning and Cartesian space trajectory planning. **5M**

- Joint Space Trajectory Planning – **2.5M**
- Cartesian Space Trajectory Planning. – **2.5M**

7 b) Discuss the concept of compliance in robotic systems with an example. **5M**

- Compliance: 2M
- Need for Compliance in Robotics: 2M
- Importance of Compliance in Robotics: 1M

8 a) Explain the working principle of a stepper motor and its advantages in robotics. **5M**

- Construction of Stepper Motor: 2M
- Working Principle of Stepper Motor: 2M
- Advantages: 1M

8 b) Discuss in detail the working of Position Encoders in robots. **5M**

- Working Principle of Position Encoders: 3M
- Advantages of Position Encoders: 2M

9) Write a simple ladder logic program to control a motor using start and stop push buttons. 5M

- Ladder Logic Program - 2M
- Explanation of logic - 3M

9 b) Differentiate between Modbus and Profibus communication protocols. **5M**

Modbus - 2.5M

- Profibus -2.5M

10 a) Explain the advantages and disadvantages of offline programming. **5M**

- Advantages of Offline Programming - 2.5M
- Disadvantages of Offline Programming - 2.5M

10 b)Write short notes on ABB RAPID language and its key features. **5M**

- Any 5 Features of ABB RAPID – 5*=5M

11 a)Discuss how MATLAB Robotics Toolbox supports kinematics and path planning. **5M**

- Support for Kinematics- 2.5M
- Support for Path Planning - 2.5M

11 b) Explain the concept of human-robot collaboration with suitable examples. **5M**

- Characteristics of Human–Robot Collaboration - 2.5M
- Examples of Human–Robot Collaboration - 2.5M

Code: 23ME6501

PVP 23

III B.Tech - I Semester - Honors Examinations - NOVEMBER 2025
INDUSTRIAL ROBOTICS AND AUTOMATION
(HONORS IN MECHANICAL ENGINEERING)

Duration: 3 hours

Max. Marks: 70

Note:

1. This question paper contains two parts: A and B.
2. Part-A contains 10 short answer questions. Each question carries 2 marks.
3. Part-B contains 5 essay questions with internal choice; choose either or. Each question carries 10 marks.
4. Assume any missing data suitably.
5. Use of Non-Programmable Scientific Calculator is allowed.

BL – Bloom's Level **CO – Course Outcome**

PART – A

1. a) Differentiate between fixed and programmable automation. [2M]

Fixed automation uses specialized equipment for high-volume, repetitive production with little flexibility.

Programmable automation allows reprogramming of machines to produce different products, offering medium flexibility.

1. b) What is meant by collaborative robots (Cobots)? [2M]

Collaborative robots are robots designed to safely work alongside human workers without physical barriers, using sensors and force-limiting mechanisms.

1. c) What is a homogeneous transformation matrix? [2M]

A homogeneous transformation matrix is a 4×4 matrix used in robotics to represent both rotation and translation of a coordinate frame in a single matrix.

$$\begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation → (upper-left 3×3); Translation → (d_x, d_y, d_z column)

1. d) Define forward kinematics. [2M]

Forward kinematics calculates the position and orientation of the robot end-effector using the given joint parameters.

1. e) List any two differences between Lagrangian and Newton–Euler methods. [2M]

S.No	Lagrangian–Euler method	Newton–Euler method
1	Lagrangian method is energy-based and uses kinetic and potential energy	Newton–Euler is force-based and uses forces and moments
2	Lagrangian is computationally heavier	Newton–Euler is faster for real-time applications

1. f) Define trajectory planning. [2M]

Trajectory planning refers to determining a smooth and feasible path for the robot to move from start to goal while satisfying velocity and acceleration limits.

1. g) What are major applications of vision systems in robotics? [2M]

Major applications include object detection and recognition, inspection and quality control, assembly guidance, and robot navigation.

1. h) What is ladder logic? [2M]

Ladder logic is a graphical programming language used in PLCs that represents control logic in the form of electrical relay circuit diagrams.

1. i) Write any two advantages of offline programming. [2M]

- Reduces robot downtime since programming is done without stopping production
- Allows simulation and error-checking before implementation.

1.j) Define teach pendant programming. [2M]

Teach pendant programming is a method where the robot is manually guided using a hand-held device to record positions and create programs.

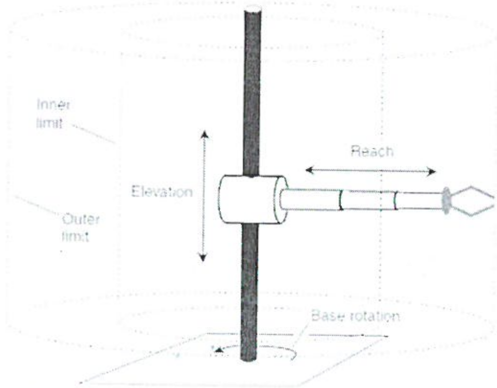
PART – B
UNIT – I

2 a) What is the work envelope of a robot? Sketch and explain two views to indicate the work envelope of a cylindrical robot. **[5M]**

The work envelope of a robot is the three-dimensional space within which its end-effector can move and perform tasks.

It is determined by the robot's configuration, joint limits, link lengths, and degrees of freedom.

A cylindrical robot combines a rotating base, a linear vertical motion, and a radial linear motion. Hence its work envelope resembles a cylinder.



Work envelope

Top View Explanation:

- The base rotation sweeps a circular area.
- The radial arm extends and retracts, forming a set of concentric circles.
- Thus, the top view shows a circular ring-shaped region.

Side View Explanation

- The vertical prismatic joint moves the arm up and down.
- The radial joint provides horizontal extension.
- The side view shows a rectangular profile that extends vertically.

Together, these views form a hollow cylindrical work volume.

Applications

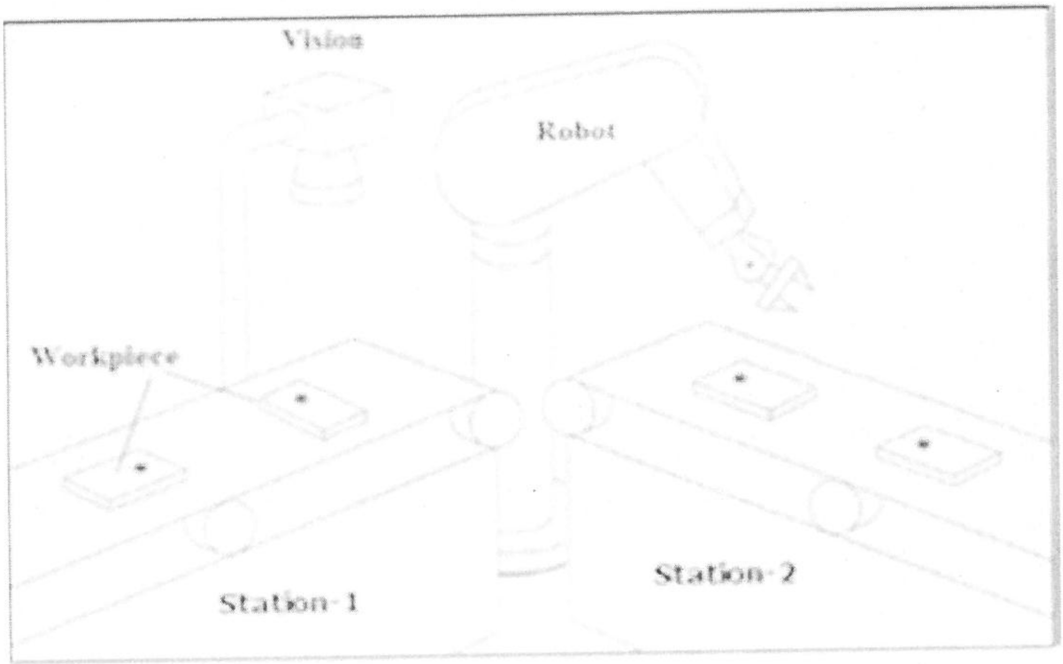
Material handling

Machine loading/unloading

Simple pick-and-place automation

---

2 b) Describe the various considerations taken into account while using robots in material handling applications. **[5M]**



When using robots for material handling, several factors must be considered:

1. **Payload Capacity**: Robot must support the weight of objects, including gripper weight.

2. **Reach and Work Envelope**: Must cover entire handling region without singularities or collisions.

3. **Speed and Cycle Time**: High-speed operations improve productivity while ensuring safety.

4. **Type of End-Effector**: Grippers must match object shape, size, fragility, and surface properties.

5. **Precision and Repeatability**: Critical for accurate placement of parts.

6. **Safety Requirements**: Includes physical barriers, vision monitoring, and emergency stops.

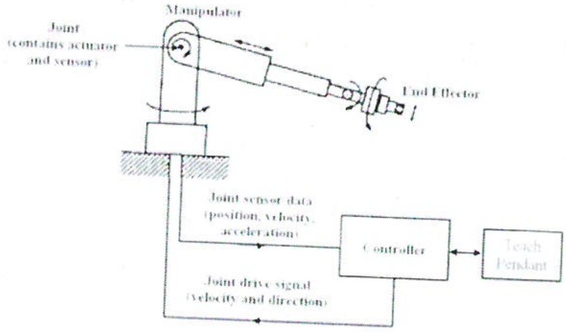
7. **Environmental Conditions**: Dust, heat, humidity, or chemical exposure may require protective covers.

8. **Integration with Other Systems**: Must communicate with conveyors, PLCs, sensors, and vision systems.

9. **Floor Space and Layout**: Robot position must optimize material flow and minimize bottlenecks.

10. **Cost and ROI**: Evaluates installation cost vs. performance improvements.

3 a) Describe the function of the four basic components of robotic systems. [5M]
A typical robotic system consists of four major components:



1. **Mechanical Structure**: Includes links, joints, actuators, and body. Provides physical framework for robot motion.
2. **Sensors**: Provide internal (joint position, temperature) and external (vision, proximity) feedback. Enable closed-loop control, safety, and accuracy.
3. **Controller:** Acts as the brain of the robot.
   - ➢ Executes programs
   - ➢ Processes sensor data
   - ➢ Generates control commands
   - ➢ Ensures stable and accurate motion
4. **End-Effector:** Tool attached to robot wrist for interacting with environment. Examples: grippers, welding torch, spray gun, cutters.

Together, these components allow the robot to sense, decide, and act.

---

3 b) Evaluate the benefits and limitations of using robots in automobile assembly lines. [5M]

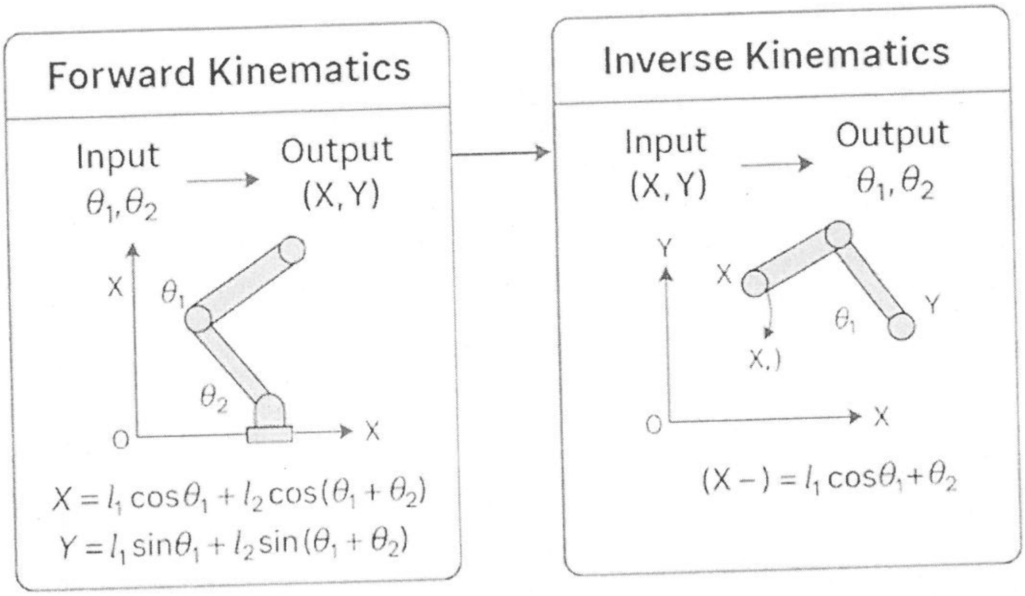
Benefits:
1. High speed and improved production rate
2. Excellent repeatability and quality
3. Increased worker safety (handling dangerous tasks)
4. Ability to work continuously (24×7)
5. Enhanced precision in welding, painting, assembly
6. Reduction of labor cost in long term
7. Better utilization of space and workflow automation

Limitations:
1. High initial investment
2. Requires skilled personnel for programming/maintenance
3. Limited flexibility for rapidly changing models
4. Possible job displacement concerns
5. Requires robust safety systems
6. Downtime during breakdowns may affect entire line

Robots significantly improve productivity and quality but demand high initial cost and skilled management.

---

4 a) Explain the difference between forward kinematics and inverse kinematics. [5M]



Forward Kinematics (FK):

- Computes the end-effector position and orientation from given joint angles.
- Direct and simple; uses transformation matrices.
- Always produces a unique result.

Inverse Kinematics (IK):

- Computes joint angles required for a desired end-effector pose.
- Complex and nonlinear; may have multiple or no solutions.
- Requires numerical or analytical methods.

S.No	Forward Kinematics	Inverse Kinematics
1	Forward Kinematics is straightforward	Inverse Kinematics is mathematically complex
2	Forward Kinematics has unique solutions	Inverse Kinematics often has multiple solutions
3	Forward Kinematics used for simulation	Inverse Kinematics used for planning actual robot motion
4	Forward Kinematics uses DH parameters	Inverse Kinematics solves nonlinear equations

1b) Determine the rotation matrix for a rotation of 45° about the Z-axis, followed by 120° about y-axis & a final rotation of 90° about the x-axis. [5M]

Sol * Rotation 45° about Z-axis

$$R_z(45^\circ) = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.707 & -0.707 & 0 \\ 0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

* Rotation of 120° about y-axis

$$R_y(120^\circ) = \begin{bmatrix} \cos 120^\circ & 0 & \sin 120^\circ \\ 0 & 1 & 0 \\ -\sin 120^\circ & 0 & \cos 120^\circ \end{bmatrix}$$

$$= \begin{bmatrix} -0.5 & 0 & 0.866 \\ 0 & 1 & 0 \\ -0.866 & 0 & -0.5 \end{bmatrix}$$

* Rotation of 90° about x-axis

$$R_x(90^\circ) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c90^\circ & -s90^\circ \\ 0 & s90^\circ & c90^\circ \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Final Rotation Matrix,

$$R = R_z(45^\circ) \times R_y(120^\circ) \times R_x(90^\circ)$$

$$= \begin{bmatrix} 0.707 & -0.707 & 0 \\ 0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.5 & 0 & 0.866 \\ 0 & 1 & 0 \\ -0.866 & 0 & -0.5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

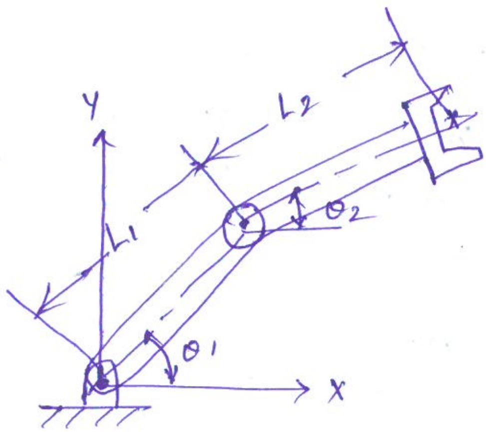
$$= \begin{bmatrix} -0.354 & 0.612 & 0.707 \\ -0.354 & 0.612 & -0.707 \\ -0.866 & -0.500 & 0 \end{bmatrix}$$

—x—

5) For a 2-link planar manipulator with link lengths L_1 & L_2 formulate the D-H parameters & transformation matrices.

[10M]

sol.



Link	a_i	α_i	d_i	θ_i
1	L_1	0	0	θ_1
2	L_2	0	0	θ_2

D-H Parameter Table:

$$^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For link 1

$$^{1-1}T_1 \text{ or } {}^{0}T_1 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & L_1 C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & L_1 S\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For link 2:

$$^{2-1}T_2 \text{ (or) } {}^{1}T_2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & L_2 C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & L_2 S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Overall Transformation Matrix,

$$^{0}T_2 \text{ (or) } T = {}^{0}T_1 \times {}^{1}T_2$$

$$= \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & L_1 C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & L_1 S\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & L_2 C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & L_2 S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow {}^{0}T_2 = \begin{bmatrix} C_1C_2 - S_1S_2 & -(C_1S_2 + C_2S_1) & 0 & L_1C_1 + L_2(C_1C_2 - S_1S_2) \\ C_1S_2 + C_2S_1 & C_1C_2 - S_1S_2 & 0 & L_1S_1 + L_2(C_1S_2 + C_2S_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(ie) $C_1C_2 - S_1S_2 = \cos(\theta_1 + \theta_2)$

$C_1S_2 + C_2S_1 = \sin(\theta_1 + \theta_2)$

(e)

$$T = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & L_1\sin\theta_1 + L_2\sin(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This is the final transformation matrix for a 2-link planar manipulator.

——— x ———

UNIT – III

6) Derive the dynamic equation of a simple 2-link planar robot using Newton–Euler formulation. **10M**

The dynamic equation for a simple 2-link planar robot using the Newton-Euler formulation can be represented in the standard matrix form: $\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) = \boldsymbol{\tau}$, where $\boldsymbol{\tau}$ is the vector of joint torques (τ_1, τ_2), $\boldsymbol{\theta}$ is the vector of joint angles (θ_1, θ_2), $\mathbf{M}(\boldsymbol{\theta})$ is the **mass matrix**, $\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}$ represents the **Coriolis and centrifugal forces** vector, and $\mathbf{G}(\boldsymbol{\theta})$ is the **gravity vector**. The specific elements of these matrices are derived algebraically by following the iterative Newton-Euler procedure described below, resulting in complex trigonometric expressions relating all physical parameters of the links to the applied torques.

Step 1: Define Parameters and Frames

Establish coordinate frames for the base (0), link 1 (1), and link 2 (2), typically following Denavit-Hartenberg conventions. Key parameters include link masses m_i, centers of mass r_{Ci}, moments of inertia I_i, and joint angles θ_i, velocities $\dot{\theta}_i$, and accelerations $\ddot{\theta}_i$.

Step 2: Forward Iteration (Velocities and Accelerations)

Iteratively calculate the angular and linear velocities and accelerations from the base frame to the tip. Assume the base frame has zero velocity and acceleration ($\boldsymbol{\omega}_0 = \mathbf{0}$, $\dot{\boldsymbol{\omega}}_0 = \mathbf{0}$, $\mathbf{v}_0 = \mathbf{0}$, $\dot{\mathbf{v}}_0 = -g\hat{\mathbf{z}}$).

The equations for link i relative to the previous link $i-1$ are:

Angular velocity:

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \dot{\theta}_i \mathbf{z}_{i-1}$$

Angular acceleration:

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1} + \ddot{\theta}_i \mathbf{z}_{i-1} + \boldsymbol{\omega}_{i-1} \times (\dot{\theta}_i \mathbf{z}_{i-1})$$

Linear acceleration of origin of frame i:

$$\dot{\mathbf{v}}_i = \dot{\mathbf{v}}_{i-1} + \dot{\boldsymbol{\omega}}_{i-1} \times \mathbf{p}_i + \boldsymbol{\omega}_{i-1} \times (\boldsymbol{\omega}_{i-1} \times \mathbf{p}_i)$$

Linear acceleration of center of mass of link i:

$$\dot{\mathbf{v}}_{Ci} = \dot{\mathbf{v}}_i + \dot{\boldsymbol{\omega}}_i \times \mathbf{r}_{Ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{r}_{Ci})$$

Step 3: Forward Iteration (Forces and Moments)

Calculate the forces and moments acting on each link due to gravity and inertia.

Net force on link i:

$$\mathbf{F}_i = m_i \dot{\mathbf{v}}_{Ci}$$

Net moment on link i:

$$\mathbf{N}_i = \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i)$$

Step 4: Backward Iteration (Forces, Moments, and Torques)

Iteratively calculate the forces and moments transmitted between links, from the end-effector back to the base, to find the required joint torques τ_i. Assume zero external forces and moments at the end-effector.

Force transmitted from link $i+1$ to link i:

$$\mathbf{f}_i = \mathbf{F}_i + \mathbf{R}_{i+1} \mathbf{f}_{i+1}$$

Moment transmitted from link $i+1$ to link i:

$$\mathbf{n}_i = \mathbf{N}_i + \mathbf{R}_{i+1} \mathbf{n}_{i+1} + \mathbf{p}_i \times \mathbf{f}_i + \mathbf{r}_{Ci} \times \mathbf{F}_i$$

Joint torque for link i:

$$\tau_i = (\mathbf{n}_i)^T \mathbf{z}_{i-1}$$

Step 5: Final Dynamic Equation

By performing the full algorithm for the 2-link robot, the inverse dynamics equation is obtained, which can be expressed in the generalized matrix form. The forward dynamics (finding accelerations from torques) requires inverting the mass matrix.

The final dynamic equation for a simple 2-link planar robot derived using the Newton–Euler formulation is generally expressed as a second-order differential equation in joint space: $\mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) = \boldsymbol{\tau}$.

OR

7 a) Differentiate between Joint space trajectory planning and Cartesian space trajectory planning. **5M**

Trajectory planning in robotics deals with generating a smooth, feasible, and collision-free motion for a robot from an initial to a final configuration.
Two major approaches are

- **Joint Space Trajectory Planning** and
- **Cartesian Space Trajectory Planning**.

Both methods differ in motion representation, computational complexity, and execution behavior.

1. Joint Space Trajectory Planning

Joint space trajectory planning defines the path in terms of **joint variables** (θ, d), i.e., each joint's motion is planned independently.

Characteristics

- Trajectory defined in robot joint coordinates (e.g., $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$).
- Simple computation – uses polynomial or spline interpolation.
- Ensures smooth motion because trajectory is planned directly in joint space.
- Robot moves through a sequence of intermediate joint configurations.
- Path in Cartesian space is not explicitly controlled and may be curved.

Advantages

- Fast and computationally efficient.
- Guarantees smooth velocity, acceleration, and jerk at joint level.
- Easy to avoid joint limits and singularities.
- Suitable for point-to-point motions (pick-and-place tasks).

Limitations

- End-effector path in workspace may be unpredictable.
- Cannot ensure straight-line motion in task space.

2. Cartesian Space Trajectory Planning

In Cartesian space planning, the trajectory is defined directly in the workspace (x, y, z, orientation).

Characteristics

- Motion is expressed as end-effector trajectory: $(x(t), y(t), z(t), \alpha(t), \beta(t), \gamma(t))$.
- Ensures precise path geometry (e.g., straight line, circular arc).
- Joint variables are calculated using inverse kinematics.
- Smoothness of motion depends on IK feasibility and joint constraints.

Advantages

- Provides maximum control over end-effector motion.
- Ideal for tasks needing accuracy (welding, cutting, painting).
- Maintains linear, circular, or complex paths in workspace.

Limitations

- Computationally heavy due to inverse kinematics.
- May result in discontinuities in joint velocities at singularities.
- Difficult to satisfy joint constraints while maintaining Cartesian smoothness.

---

7 b)Discuss the concept of compliance in robotic systems with an example. **5M**

Compliance:

Compliance in robotic systems refers to the ability of a robot or its end-effector to yield, deform, or adapt when interacting with its environment. It allows the robot to absorb external forces rather

than resisting them rigidly. Compliance is crucial for safe, smooth, and intelligent interaction with objects, humans, and uncertain environments.
A compliant robot behaves like a spring-damper system, allowing controlled flexibility in one or more directions while maintaining rigidity in others.

Need for Compliance in Robotics
Robotic tasks often involve interaction with the environment where small errors in position or force can cause:

- Damage to the robot or workpiece
- Excessive contact forces
- Misalignment in assembly operations
- Unsafe human–robot collaboration

Types of Compliance

(a) Passive Compliance

- Achieved through mechanical elements such as springs, compliant joints, flexible couplings, elastic materials, or shock absorbers.
- No active sensing or control.
- Simple and low-cost.

Examples:
Flexible grippers, shock-absorbing end-effectors, compliant wrists.

(b) Active Compliance

- Achieved through sensors (force/torque sensors) and control algorithms.
- Robot detects external forces and adjusts motion dynamically.
- Provides precise force control.

Examples:
Impedance control, admittance control, hybrid position/force control.

Importance of Compliance in Robotics
Compliance is essential for:

1. **Assembly operations**
   Handling tolerances, misalignments, and press-fit operations.
2. **Human–robot collaboration (Cobots)**
   Ensuring safety by limiting forces.
3. **Surface finishing**
   Polishing, grinding, and cleaning tasks benefit from flexible force control.
4. **Manipulation of fragile objects**
   Prevents damage to delicate items.
5. **Contact-rich tasks**
   Such as peg-in-hole insertion, welding, and sanding.

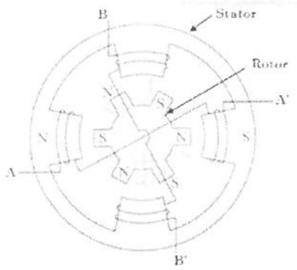
Example: Peg-in-Hole Assembly
One of the best examples to explain compliance is the classical **peg-in-hole insertion** task.

UNIT – IV

8 a) Explain the working principle of a stepper motor and its advantages in robotics. 5M

A **stepper motor** is an electromechanical device that converts electrical pulse signals into **precise, discrete angular movements** called *steps*. Each pulse moves the motor shaft by a fixed

angle (step angle), making it ideal for applications requiring accurate positioning and repeatability.
Common step angles: **0.9°, 1.8°, 7.5°, 15°**.



Construction of Stepper Motor

A stepper motor consists of:

1. Stator
- Made of multiple electromagnetic windings (phases: A, B, C…)
- When energized in a sequence, they form a rotating magnetic field.

2. Rotor
- Either a permanent magnet rotor, variable reluctance rotor, **or** hybrid rotor.
- Aligns itself with the energized stator poles.

3. Shaft
- Rotates in discrete steps depending on the excitation pattern.

Working Principle of Stepper Motor

The working principle is based on the alignment of rotor magnetic field with the rotating magnetic field of the stator windings.

Step-by-Step Working
1. When a particular stator coil is energized, it creates a magnetic pole.
2. The rotor aligns itself with that pole.
3. Energizing the next coil shifts the magnetic field position.
4. The rotor follows this shift, moving by one "step."
5. Continuous sequential energizing produces rotary motion in fixed increments.

The direction of rotation depends on the order of excitation:
- **A → B → C** → clockwise
- **C → B → A** → counter-clockwise

Advantages of Stepper Motors
- Precise Positioning
- Excellent Repeatability
- Open-Loop Control
- High Holding Torque
- Low-Speed High Torque
- Good for Start-Stop Applications
- High Reliability and Rugged Construction
- Cost-Effective

Applications of Stepper Motors
- 3D printers and CNC machines
- Robotic arm joint control
- Automated guided vehicles (AGVs)
- Pick-and-place robots
- Camera positioning and pan-tilt units

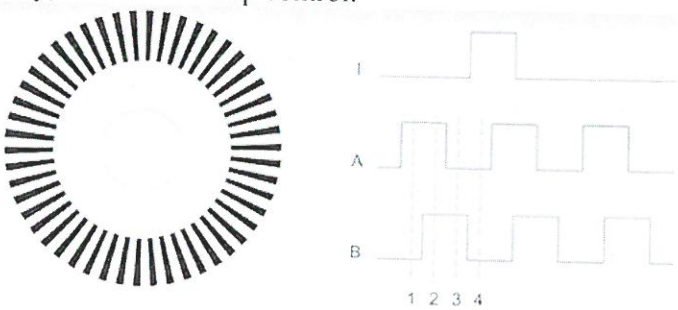
- ➢ Medical robotic equipment
- ➢ Conveyor indexing systems

Stepper motors are widely used in robotics due to their simplicity, precision, reliability, and excellent controllability. Their ability to convert digital pulse commands into accurate mechanical movements makes them essential for robotic systems requiring precise positioning, low-speed torque, and cost-effective motion control.

---

8 b) Discuss in detail the working of Position Encoders in robots. **5M**

Position encoders are essential feedback devices used in robotic systems to measure the position, speed, and direction of robot joints or end-effectors. They convert mechanical motion into electrical signals and provide accurate feedback to the robot controller, ensuring precise movement, repeatability, and closed-loop control.



A position encoder is an electromechanical sensor that detects the angular or linear displacement of a mechanical component.

In robots, these encoders are typically mounted on:

- Servo motors
- Stepper motors
- Robot joints
- Linear actuator shafts

Encoders play a critical role in *robot motion control*, *trajectory tracking*, and *error correction.*

Types of Position Encoders

- ➢ Incremental Encoders
- ➢ Absolute Encoders
- ➢

<u>Working Principle of Position Encoders</u>

Although different encoder types have variations, their basic working principle involves:

Mechanical to Electrical Conversion:
Movement of the robotic joint or motor shaft is mechanically linked to the encoder disc or strip.

Sensing Element:
Encoders use either:

- Optical sensing
- Magnetic sensing
- Capacitive sensing
- Resistive/potentiometric sensing

Signal Generation:
1. Optical Encoders
These are the most widely used.

- Consist of a rotating disc with alternating transparent and opaque sectors.
- A **LED** shines light through the disc.
- A **photo-detector** receives the interrupted light.
- As the disc rotates, the light is chopped into pulses.

Each pulse represents one step of rotation → position change.

2. Magnetic Encoders

- Use a rotating magnet and a Hall-effect sensor.
- Magnetic field changes as the shaft rotates, generating analog or digital signals.

3. Capacitive Encoders

- Changes in capacitance occur as patterned rotor and stator plates move.
- Used for high precision in compact spaces.

Signal Processing

The raw signals from sensors are processed by the robot controller:

For Incremental Encoders

- Count pulses to measure displacement.
- Quadrature signals determine direction.
- Pulse frequency gives speed.
- Index pulse marks a reference position.
-

For Absolute Encoders

- Each position corresponds to a unique binary or Gray code.
- Controller reads the digital code to get exact position instantly.

Advantages of Position Encoders

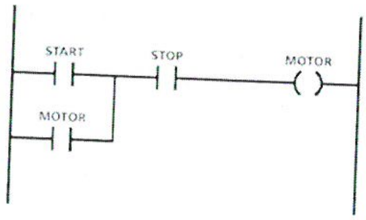
- High accuracy and resolution
- Precise speed and position feedback
- Reliable for closed-loop control
- Fast response suitable for dynamic robotics
- Absolute encoders give position even after power loss
- Compact and durable

OR

9) a) Write a simple ladder logic program to control a motor using start and stop push buttons. 5M

A simple motor control in ladder logic uses a "seal-in" (or latching) circuit to keep the motor running after the start button is released. The logic is typically contained in a single rung.

Here is the ladder logic diagram and an explanation of how it works:



Ladder Logic Program

The ladder diagram consists of one main rung of logic:

- The rung has three contacts in series with the output coil Q:1 .
- A parallel branch with a contact related to the output (Q:1 auxiliary contact) is used for latching.

Rung	**Logic Description**
Rung 1	A normally open (NO) I:1 (Start) contact and a normally closed (NC) I:2 (Stop) contact are in series. In parallel with I:1 is an NO auxiliary contact from the Q:1 (Motor) output. The result energizes the Q:1 (Motor Coil) output.

Operation (Seal-In Logic)

1. **Stop Condition:** Initially, the circuit is de-energized. The physical Stop button is NC, so the I:2 contact in the ladder logic appears closed (true) to the PLC. The physical Start button is NO, so the I:1 contact is open (false). The Motor Q:1 output is OFF.
2. **Starting the Motor:** When the **Start push button (I:1)** is pressed, it closes the I:1 contact in the PLC program. This completes the circuit path through the now-closed I:2 contact, energizing the Q:1 Motor Coil output.
3. **Latching (Holding):** Once Q:1 is energized, its own auxiliary NO contact (Q:1 contact in parallel with I:1) closes. This creates a parallel path for the current/logic flow (the "seal-in" path).
4. **Releasing the Start Button:** When the physical Start button is released, the I:1 contact opens, but the seal-in path through the Q:1 auxiliary contact remains closed, keeping the motor running continuously.
5. **Stopping the Motor:** When the **Stop push button (I:2)** is pressed, it momentarily opens the physical NC contact, breaking the circuit path in the ladder logic. The Q:1 output de-energizes, which in turn opens its own seal-in contact, and the motor stops. The circuit remains stopped until the Start button is pressed again.

---

9 b) Differentiate between Modbus and Profibus communication protocols. 5M

Modbus and Profibus are two widely used industrial communication protocols, but they differ significantly in architecture, speed, application, and data handling. Their comparison is given below:

Parameter	**Modbus**	**Profibus**
Origin & Standard	Developed by *Modicon (Schneider Electric)*; one of the earliest	Developed by *Siemens and the German government* under

	industrial communication protocols.	PROFIBUS International
Communication Type	Master–Slave communication	Multi-Master communication in addition to Master–Slave
Physical Layer & Variants	Works on RS-232, RS-485, TCP/IP (Modbus RTU, Modbus ASCII, Modbus TCP). Simple wiring and low-cost hardware.	Uses RS-485, MBP (Manchester Bus Powered), and fiber optic networks. Popular versions: Profibus-DP, Profibus-PA.
Data Transmission Speed	Lower speed, typically 1.2 kbps – 115 kbps, Modbus TCP up to 100 Mbps. Suitable for slow or moderate-speed devices.	Very high communication speed: 9.6 kbps to 12 Mbps. Suitable for high-performance automation systems.
Network Size	Supports up to 247 devices per network.	Supports up to 126 devices, but with much faster data exchange.
Complexity & Configuration	Very simple, easy to configure, install, and troubleshoot.	More complex, requires special configuration tools and trained personnel.
Application Areas	-SCADA systems -Remote monitoring -Small and medium process control -Simple field devices (sensors, meters, drives)	-High-speed automation -Factory automation (Profibus-DP) -Process industries (Profibus-PA) -Large industrial networks
Cost	Low cost, economical, ideal for basic communication	Higher cost due to hardware complexity and speed capabilities

---

10 a)Explain the advantages and disadvantages of offline programming. **5M**

Offline programming (OLP) is the process of programming a robot using simulation software on a computer, without stopping the actual robot or production line. Programs are created, tested, and optimized virtually before being transferred to the real robot.

Advantages of Offline Programming

1. No Production Downtime

- Robots do not need to be stopped for teaching.
- Production continues while programming is done in parallel, improving productivity.

2. Safe Testing Environment

- Programs are tested in a simulated environment.
- Prevents collisions, damage to equipment, and safety risks to operators.

3. High Accuracy and Optimization

- Software provides precise modelling of robot reach, speeds, and path planning.
- Complex paths and motions (e.g., welding, painting, machining) are easier to optimize.

4. Faster Deployment

- Most of the programming work is completed before installation.
- Reduces start-up time during commissioning.

5. **Better Planning for Layout and Fixtures**
   - Simulation helps visualize cell layout, robot placement, and tooling design.
   - Helps detect reachability issues early.
6. **Ideal for Complex Systems**
   - Multi-robot coordination, conveyor tracking, and advanced applications can be tested virtually.
   - Saves time compared to manual teaching.

Disadvantages of Offline Programming

1. **Requires Accurate Robot and Cell Model**
   - Any difference between the virtual model and real environment creates errors.
   - Requires high-quality CAD models of tools, fixtures, and cell layout.
2. **Higher Initial Cost**
   - OLP software is expensive.
   - Requires powerful computers and trained personnel.
3. **Need for Skilled Programmers**
   - Operators must understand simulation, CAD, and robot kinematics.
   - Not suitable for quick, simple tasks where online teaching is faster.
4. **Calibration Errors**
   - Robot base, tool center point (TCP), and work objects must be calibrated accurately.
   - Poor calibration leads to mismatches between simulation and real execution.
5. **Limited Real-World Feedback**
   - Simulations cannot fully replicate real-world conditions such as:
     - tool wear
     - sensor feedback
     - variations in workpiece
     - environmental constraints
   - Final fine-tuning on the real robot is still required.

---

10 b)Write short notes on ABB RAPID language and its key features. 5M

ABB RAPID is a high-level, structured programming language used for controlling ABB industrial robots. It is designed to be simple, flexible, and powerful, enabling programmers to write motion instructions, handling routines, I/O operations, error control, and communication tasks. RAPID is used in ABB's RobotStudio (offline programming) and on the robot's teach pendant (FlexPendant).

Features of ABB RAPID

1. **High-Level and Easy-to-Read Syntax**
   - RAPID uses simple English-like commands (e.g., MoveJ, MoveL, WaitTime), making it easy to understand and debug.
   - Structure is similar to modern programming languages with procedures, functions, loops, and variables.
2. **Powerful Motion Control**
   - Supports different types of robot movements:
     - **MoveJ** (Joint interpolation)
     - **MoveL** (Linear interpolation)
     - **MoveC** (Circular interpolation)
   - Allows control of speed, accuracy, orientation, and task frames.
3. **Modular Programming**
   - Programs can be divided into **modules**, **procedures**, and **functions**, improving readability and reusability.
   - Makes large automation projects easier to manage and update.

4. Support for I/O and Device Communication

- RAPID can read and control digital and analog signals.
- Supports communication protocols such as Ethernet/IP, DeviceNet, and serial communication, enabling integration with PLCs and sensors.

5. Error and Exception Handling

- Provides structured error handling using TRAP routines and ERROR functions.
- Allows safe recovery from unexpected events such as collisions or signal failures.

6. Multitasking Capability

- Allows multiple tasks to run simultaneously.
- Useful for applications requiring background monitoring, conveyor tracking, or communication while robot motions are executed.

7. Integration with RobotStudio

- RAPID programs can be created, simulated, tested, and debugged in ABB RobotStudio.
- Enables offline programming without interrupting production, improving productivity.

ABB RAPID is a powerful and user-friendly robot programming language that supports advanced motion control, modular design, multitasking, and integration with external devices. Its simplicity and flexibility make it widely used in industrial automation.

---

11 a)Discuss how MATLAB Robotics Toolbox supports kinematics and path planning. 5M

The **MATLAB Robotics Toolbox** provides a powerful set of functions and simulation tools that help in modeling robots, computing kinematics, and generating optimal path-planning solutions. It is widely used in teaching, research, and robotic system development.

1. Support for Kinematics

a) Forward Kinematics

- The toolbox allows creation of robot models using DH parameters, URDF files, or predefined robot libraries.
- Functions like rigidBodyTree, getTransform, and show compute and visualize joint positions, end-effector pose, and robot configurations.
- Users can determine the position and orientation of the end-effector for any set of joint angles.

b) Inverse Kinematics

- The toolbox provides solvers such as inverse Kinematics and generalized Inverse Kinematics.
- These solvers compute the joint angles needed to reach a desired end-effector pose while respecting joint limits.
- Task-space constraints (position, orientation, joint constraints) can be incorporated easily.

c) Kinematic Simulation & Visualization

- Tools like Robot Visualizer and Simulink 3D Animation allow 3D simulation of robot motion.
- Users can animate kinematic results and verify reachability and collision-free motion.

2. Support for Path Planning

a) Motion Planning Algorithms

The toolbox supports multiple advanced path-planning algorithms, including:

- RRT (Rapidly-Exploring Random Tree)
- RRT*
- PRM (Probabilistic Roadmap)
- A* and grid-based planners

These planners help in generating collision-free trajectories in complex environments.

b) Trajectory Generation

- Functions like trapveltraj, cubicpolytraj, and bsplinepolytraj generate smooth joint-space and task-space trajectories.
- These trajectories ensure continuous velocity and acceleration profiles for robot motion.

c) Environment and Collision Modeling

- Users can model obstacles using collision objects, meshes, and occupancy maps.
- The planners automatically compute safe paths avoiding obstacles.

d) Simulation with Dynamics (Optional)

- Integration with Simscape Multibody enables dynamic simulation of path-planned trajectories for realistic behavior.

MATLAB Robotics Toolbox provides comprehensive tools for both **kinematics** and **path planning**, enabling users to model robots, compute complex motions, and generate collision-free trajectories efficiently. It supports research, teaching, and industrial robotic system development.

---

11 b) Explain the concept of human-robot collaboration with suitable examples. 5M

Human–Robot Collaboration (HRC) refers to a work environment where humans and robots operate together in a shared workspace, performing complementary tasks to achieve higher productivity, safety, and efficiency. Unlike traditional industrial robots that work inside cages, collaborative robots (cobots) are designed to work directly alongside humans with built-in safety features.

HRC combines the strength, precision, and repeatability of robots with the judgment, flexibility, and decision-making ability of humans.

<u>Characteristics of Human–Robot Collaboration</u>

1. Shared Workspace

Humans and robots work close to each other, sharing tools, components, or work surfaces.

2. Safety Features

Collaborative robots include:

- Force and torque sensing
- Speed and separation monitoring
- Emergency stop behavior
- Rounded edges and lightweight designs

3. Task Complementarity

Humans handle tasks requiring creativity, inspection, complex decision-making, while robots perform repetitive, heavy, or precision tasks.

4. High Flexibility

HRC systems can be easily reprogrammed and adapted for small-batch manufacturing, assembly, and packaging.

<u>Types of Human–Robot Collaboration</u>

1. Coexistence

Humans and robots work near each other but without direct interaction.

2. Sequential Collaboration

Human and robot perform tasks one after another at the same workstation.

3. Cooperative Collaboration

Both work simultaneously on the same part or task.

4. Responsive Collaboration

Robot adjusts speed or motion based on human presence and action.

Examples of Human–Robot Collaboration

Example 1: Assembly Line Support

Cobot assist a human worker in automobile assembly by:

- holding heavy components
- performing precise screwing operations
- allowing the human to focus on inspection and alignment

This reduces fatigue and improves accuracy.

Example 2: Packaging and Palletizing

Robots handle:

- repetitive lifting
- box stacking
- labeling tasks
  Humans supervise quality, manage product flow, and handle exceptions.
  This improves productivity and reduces ergonomic injuries.
-

Example 3: Logistics and Warehousing

Mobile robots collaborate with human pickers:

- robots transport bins and materials
- humans pick items requiring judgment
  This increases speed, reduces walking distance, and improves workflow.

Human–Robot Collaboration integrates human intelligence with robotic precision, enabling safer, more efficient, and more flexible manufacturing. With advancements in sensing, AI, and safety technologies, HRC is becoming essential in modern industrial automation.
