ZENITH STUDY HUB

SPPU • TE ROBOTICS & AUTOMATION • SEMESTER V

Robot Kinematics & Dynamics

Unit II | Forward and Inverse Kinematics

PCC303-ROA • 2024 Pattern • 8 teaching hours

The Concept-to-Calculation Guide

SIX FULL TOPIC CHAPTERS • ORIGINAL VECTOR DIAGRAMS
EXAM ANSWERS • NUMERICAL COMPANION • FORMULA SHEET

SOURCE AND SCOPE

Study design: Every syllabus topic is explained from first principles, followed by diagrams, formula derivations, checks, common errors and exam-writing guidance. Problems with substantial calculations are also in a separate worked-numericals PDF. The six-topic structure follows the supplied official SPPU 2024 Pattern syllabus, printed page 19. All diagrams here are newly drawn.

Prepared by

Om Deshmukh

Zenith Study Hub • 21 November 2026 • Edition 3 pilot

zenith.ironlabs.tech

How to study Unit II without memorizing blindly

IN SIMPLE WORDS

A robot's **forward kinematics** tells you where its tool will be when you know its joint positions. **Inverse kinematics** does the reverse: it tells you which joint positions can place the tool at a chosen pose. Both describe geometry, not the forces required to move.

A repeatable problem-solving method: (1) draw the arm and mark the fixed base; (2) define joint angles and positive directions; (3) identify link lengths and units; (4) choose geometry, transformations or D–H; (5) calculate; (6) check physical reach, joint limits, and both branches where they exist.

SOURCE AND SCOPE

Official basis: SPPU, TE Robotics and Automation, 2024 Pattern, effective 2026–27, syllabus printed page 19. Unit II consists of six printed topic groups. ESE is 70 marks across five units (general scheme: 14 per unit); no specific 7+7 exam question pattern is guaranteed.

Supporting references: Lynch and Park, *Modern Robotics*, Chs. 3, 4, 6 and Appendix C (official online chapters); John J. Craig, *Introduction to Robotics: Mechanics and Control*, and supplied earlier Zenith RKD notes. Explanations and figures below are freshly authored rather than reproduced.

Previous papers: named questions from 2019 Pattern papers 6583-177, 6262-175 and 6403-244 are used as topic practice only. Questions depending on unclear scanned robot figures are not numerically solved by guessing the missing geometry.

Notation used throughout: $q_1, q_2, \ldots$ are joint coordinates; L_1, L_2 link lengths; **p** position; R orientation; T a 4×4 pose transform. Angles in trigonometric formulas are in radians unless a degree mark is shown. atan2(y, x) is the quadrant-aware angle; it is not the same as $\tan^{-1}(y/x)$ in every quadrant. A positive revolute rotation follows the right-hand rule.

Syllabus item	Best way to learn it
Forward kinematics	Draw links, derive coordinates, multiply transformations and check a numerical example
D–H convention	Assign frames, fill a four-parameter table and verify a transformation product
Inverse kinematics	Build a triangle, show both branches, check reach and substitute angles into forward equations
Solution strategies	Compare geometric, algebraic and iterative methods; trace the algorithm
Redundancy and solutions	Visualize multiple robot postures, singularity and joint-limit constraints
2-link / 6-DOF example	Solve a complete 2R case and understand an explicitly specified 6-joint example

Contents

1 Forward kinematics: joint positions to tool pose **6**

1.1 Start with the robot's joints, links and coordinate frames . . . 6
1.2 Deriving forward position for a single rotary link . . . 6
1.3 Deriving the complete 2R equations step by step . . . 6
1.4 Extending the pattern to a planar 3R arm . . . 7
1.5 From two coordinates to a complete 3D tool pose . . . 7
1.6 Build a planar transform without memorizing . . . 8
1.7 A reusable forward-kinematics flowchart . . . 8
1.8 Six quick conceptual checks . . . 8

2 Denavit–Hartenberg (D–H) convention **9**

2.1 Why four parameters, and what each means . . . 9
2.2 Frame assignment: slow, practical procedure . . . 10
2.3 Derive the standard 4×4 D–H matrix . . . 11
2.4 D–H table for the planar 2R robot . . . 11
2.5 Prismatic and spatial joints: what changes? . . . 11
2.6 D–H versus geometric vectors versus PoE . . . 12

3 Inverse kinematics: tool target to joint angles 12

3.1 Exactly what is given and what is unknown? . . . 13
3.2 Reachability: decide whether a solution can exist first . . . 13
3.3 Derive the elbow angle from the cosine rule . . . 13
3.4 Derive the shoulder angle with two smaller angles . . . 14
3.5 What happens at the boundaries? . . . 14
3.6 IK with a desired orientation for a 3R planar arm . . . 15
3.7 Why a closed-form IK answer may not exist . . . 15

4 How to choose and apply an inverse-kinematics strategy 16

4.1 The main families of solution methods . . . 16
4.2 Geometrical strategy: visualizing before solving . . . 16
4.3 Algebraic strategy: start from forward equations . . . 16
4.4 Numerical strategy: improve a guess repeatedly . . . 17
4.5 Damped least squares in one understandable formula . . . 17
4.6 Joint limits and collision checks are not optional . . . 18

5 Redundancy, multiple solutions and singular cases 18

5.1 Multiple discrete answers are not the same as redundancy . . . 18
5.2 Define task dimension and redundancy clearly . . . 19
5.3 Why would an engineer intentionally add redundant joints? . . . 19
5.4 Self-motion and the Jacobian null space . . . 19
5.5 Singular configurations: a robot may lose a direction of motion . . . 20

6 Full cases: 2-link planar and 6-DOF industrial arm 20

6.1 Case A: a two-link planar robot from start to finish . . . 20
6.2 Case B: what "6-DOF industrial manipulator" means . . . 21
6.3 An explicitly specified six-axis teaching model . . . 21
6.4 Check this model at the home configuration . . . 22
6.5 Position and orientation decoupling: when you may use it . . . 22
6.6 What full six-axis inverse kinematics needs . . . 22

7 Verified older-paper questions with model answers 23

7.1 2019 Pattern paper 6583-177, November 2025, Q1(a), physical p. 1, 9 marks . . . 23

7.2 2019 Pattern paper 6583-177, November 2025, Q2(a), physical p. 1, 9 marks . . 24
7.3 2019 Pattern paper 6403-244, May 2025, Q2(a), physical p. 1, 8 marks 24
7.4 2019 Pattern paper 6262-175, April 2024, Q1(b), physical p. 1, 9 marks 25

8 Practice bank and answer-check prompts 25
8.1 Definitions and diagram questions 25
8.2 Calculations and derivations 26
8.3 Analytical reasoning and exam-length answers 26

9 One-night revision: formulas, flow and mistakes 27
9.1 Formula sheet — use the declared conventions 27
9.2 Six common exam slips and fast corrections 28
9.3 How to write a clear long-form exam answer 28

10 Sources and independent verification 28

ZENITH

1 Forward kinematics: joint positions to tool pose

TOPIC TYPE: Geometric + mathematical + diagram based **LEARNING GOAL: Given joints, find the tool position and orientation.**

IN SIMPLE WORDS

Imagine entering two joint angles into a robot controller. Forward kinematics is the calculation that predicts the robot's hand position before the motors even start moving. You already know the link lengths and angles; you calculate the final location.

1.1 Start with the robot's joints, links and coordinate frames

A **serial manipulator** has rigid links joined in a chain. The first link is attached to a fixed base; each new joint changes the orientation or position of every link beyond it. A **revolute joint** contributes an angular coordinate (often q_i measured in radians or degrees); a **prismatic joint** contributes a linear displacement (usually measured in metres). Thus a joint-space vector for a two-joint rotary arm is $\mathbf{q} = [q_1\ q_2]^T$. The **end-effector pose** is more informative than position alone: it includes both location $\mathbf{p}$ and orientation R relative to a stated reference frame.

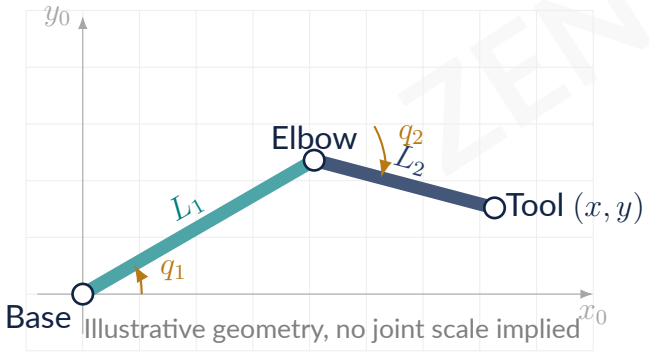

Figure. A planar two-revolute-joint (2R) robot. Joint 1 is measured from $+x_0$; joint 2 is measured relative to link 1. A negative q_2 bends the elbow clockwise.

1.2 Deriving forward position for a single rotary link

For a length L turning by angle q about the origin, draw a right triangle with hypotenuse L. The horizontal component is $x = L\cos q$; the vertical component is $y = L\sin q$. These relations come from the definitions of cosine and sine. At $q = 0$, the tip must lie at $(L, 0)$; at $q = 90°$, it must lie at $(0, L)$. Testing these cases immediately catches a swapped sine/cosine.

1.3 Deriving the complete 2R equations step by step

For the 2R arm in the figure, link 1 points at an *absolute* angle q_1 measured from the base x-axis. Link 2 is rotated by q_2 *relative to link 1*, so its absolute angle is $q_1 + q_2$. The tip is the vector sum

of the two link directions:

$$\mathbf{p}_{\text{tip}} = L_1 \begin{bmatrix} \cos q_1 \\ \sin q_1 \end{bmatrix} + L_2 \begin{bmatrix} \cos(q_1+q_2) \\ \sin(q_1+q_2) \end{bmatrix}, \tag{1}$$

$$\boxed{x = L_1 \cos q_1 + L_2 \cos(q_1+q_2)}, \tag{2}$$

$$\boxed{y = L_1 \sin q_1 + L_2 \sin(q_1+q_2)}. \tag{3}$$

The orientation of a tool aligned with link 2 is $\phi = q_1 + q_2$ in the plane. For a different tool mounting angle ϕ_0, use $\phi = q_1 + q_2 + \phi_0$. Thus two different joint configurations can sometimes reach the same position but with different elbow postures.

STEP-BY-STEP WORKED EXAMPLE

Original teaching example: A planar 2R arm has $L_1 = 0.40\,\text{m}$, $L_2 = 0.30\,\text{m}$, $q_1 = 30^\circ$, $q_2 = -45^\circ$. Find position and orientation.

1. Sketch signs: The second link's global angle is $30 - 45 = -15^\circ$.

2. Substitute:

$$x = 0.40\cos 30^\circ + 0.30\cos(-15^\circ), \quad y = 0.40\sin 30^\circ + 0.30\sin(-15^\circ).$$

3. Compute: $x \approx 0.6362\,\text{m}$, $y \approx 0.1224\,\text{m}$. The tool points at $\phi = -15^\circ$.

4. Check: The distance $\sqrt{x^2+y^2} \approx 0.6479\,\text{m}$ is below the maximum reach $0.70\,\text{m}$, so the result is physically plausible.

1.4 Extending the pattern to a planar 3R arm

Add a third link L_3 and relative angle q_3. The third link has absolute orientation $q_1 + q_2 + q_3$, so

$$x = L_1 \cos q_1 + L_2 \cos(q_1+q_2) + L_3 \cos(q_1+q_2+q_3), \tag{4}$$

$$y = L_1 \sin q_1 + L_2 \sin(q_1+q_2) + L_3 \sin(q_1+q_2+q_3), \tag{5}$$

$$\phi = q_1 + q_2 + q_3. \tag{6}$$

The rule is easier to remember than the formula: **each link contributes its length in the direction of the sum of joint angles up to that link.** This also reveals why adding a joint requires another transform in the chain.

1.5 From two coordinates to a complete 3D tool pose

A real manipulator needs a 3D pose. Position has components $\mathbf{p} = [x, y, z]^T$; orientation is a 3×3 rotation matrix R. Combine them in a **homogeneous transformation**:

$$\boxed{T = \begin{bmatrix} R & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}}, \quad T \in \mathbb{R}^{4\times 4}.$$

The fourth row makes rotations and translations multiply together in one consistent operation. The transform ${}^0T_1$ maps coordinates expressed in frame 1 into frame 0; ${}^0T_2 = {}^0T_1\,{}^1T_2$ composes successive moves. **Order matters**: matrix multiplication is generally not commutative.

EXAM TIP / COMMON MISTAKE

For a point written in homogeneous coordinates $\tilde{p} = [x\ y\ z\ 1]^T$, translation affects the point; for a direction $\tilde{v} = [v_x\ v_y\ v_z\ 0]^T$, the translation column has no effect. Never describe T as "just a 3D rotation."

1.6 Build a planar transform without memorizing

A 2D pose with angle ϕ and translation (x, y) can be stored as

$$T_{2D} = \begin{bmatrix} \cos\phi & -\sin\phi & x \\ \sin\phi & \cos\phi & y \\ 0 & 0 & 1 \end{bmatrix}.$$

For the 2R arm, multiply the elementary transformations

$$T_0^2 = R_z(q_1)\ \text{Trans}_x(L_1)\ R_z(q_2)\ \text{Trans}_x(L_2),$$

where the notation means the corresponding *homogeneous* rotation and translation matrices. This composition yields exactly the trigonometric equations derived from the vector diagram. The first method helps intuition; the second method scales more naturally to many joints.

1.7 A reusable forward-kinematics flowchart

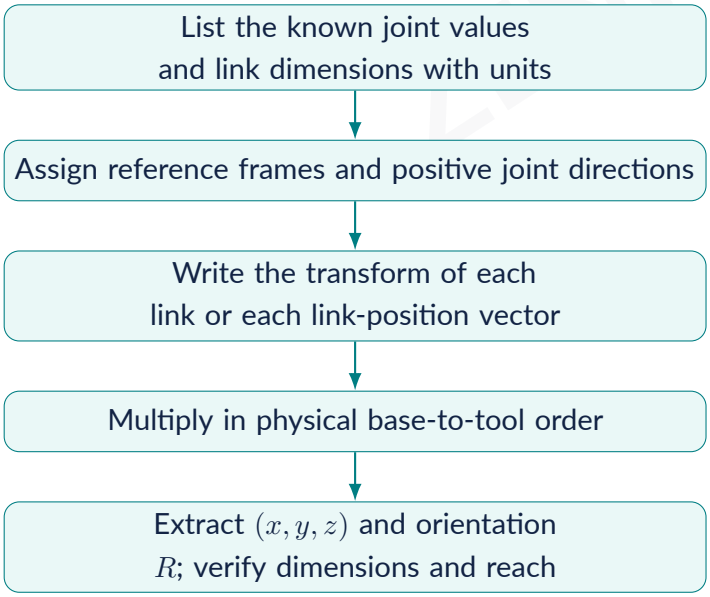

Figure. Forward-kinematics calculation pipeline.

1.8 Six quick conceptual checks

1. If q_2 is relative to link 1, why is the second angle $q_1 + q_2$? Because both rotations contribute to the link's global orientation.
2. If a revolute joint moves but all link lengths are fixed, does the workspace move? The *reachable set* is fixed by the geometry and joint limits; the current point moves within it.

3. Does knowing position determine orientation? No. Multiple configurations and orientations can share the same position.
4. Can you add angles when their axes are different in 3D? Not generally. Use rotation matrices and their specified order.
5. Is $T_0^1 T_1^2$ equal to $T_1^2 T_0^1$? Generally no; the frame subscripts also show why the latter is not a valid adjacent-frame chain.
6. When must you convert degrees to radians? Before using formulas or software routines that expect radians; the trigonometric values themselves depend on correct units.

WRITE THIS IN THE EXAM

Model answer outline: Define forward kinematics and derive the two-link planar equations. First define inputs (q_1, q_2, L_1, L_2) and outputs (tip position and angle). Draw the two-link figure with both positive angles. Resolve link 1 into horizontal/vertical components, then resolve link 2 at angle $q_1 + q_2$. Add the components to obtain the two boxed equations. State tool orientation $\phi = q_1 + q_2$ and finish with one practical use: predicting the tool's location from encoder readings.

2 Denavit–Hartenberg (D–H) convention

TOPIC TYPE: Frame-assignment + transformation + table + diagram **LEARNING GOAL: Convert a chain of mechanical links into a reusable mathematical model.**

IN SIMPLE WORDS

The D–H convention is a disciplined way of choosing coordinate frames at robot joints. Instead of inventing a different set of measurements for every robot, you describe the change from one link frame to the next with just four numbers. Put those four numbers into a table, generate a matrix for each joint, then multiply the matrices to find the final tool pose.

2.1 Why four parameters, and what each means

For a *standard/classical* D–H frame assignment, the successive coordinate frames are related through four operations in this specific order:

$$A_i = R_z(\theta_i)\, T_z(d_i)\, T_x(a_i)\, R_x(\alpha_i)$$

These represent rotation about the old z-axis by θ_i, translation along that axis by d_i, translation along the new common normal by a_i, and rotation about the new x-axis by α_i. Standard D–H frame-indexing conventions vary across textbooks; this guide uses z_{i-1} as the axis of joint i and shows the transform from frame i to $i-1$ as $A_i = {}^{i-1}T_i$. Do not mix this convention with the modified D–H one in the same table.

Parameter	Geometric meaning	What often changes?
θ_i	Rotation about z_{i-1}	Joint variable for a revolute joint
d_i	Offset along z_{i-1}	Joint variable for a prismatic joint
a_i	Common-normal distance between successive joint axes	Usually fixed by link geometry
α_i	Twist about x_i from one joint axis to the next	Usually fixed by link geometry

EXAM TIP / COMMON MISTAKE

Common confusion: Some diagrams place a_{i-1} or α_{i-1} in their tables because they use a *modified* convention. That is not necessarily an error, but you must use the corresponding transform formula. Exam answers should state the convention explicitly.

2.2 Frame assignment: slow, practical procedure

1. **Identify each joint axis.** For revolute joint i, z_{i-1} points along the shaft about which that joint rotates; for prismatic joint i, it points along the linear sliding direction.
2. **Choose each x_i.** Point it along the common normal from axis z_{i-1} to axis z_i when there is a unique common normal. If axes intersect, choose a convenient perpendicular to their plane. If axes are parallel, choose the shortest line between them. Record any frame ambiguity.
3. **Place frame origins.** Place the origins at the common-normal intersections or convenient points consistent with the chosen convention. Keep the link dimensions measured along the assigned axes.
4. **Choose each y_i.** Use the right-hand rule $y_i = z_i \times x_i$ to make a right-handed frame. Check arrows before writing any signs.
5. **Measure the four numbers.** Set θ_i and α_i as signed rotations; d_i and a_i are signed distances along the selected axes.
6. **Validate the zero posture.** At a known pose (often all $\theta_i = 0$), the product of your matrices should put the links where the drawing puts them.

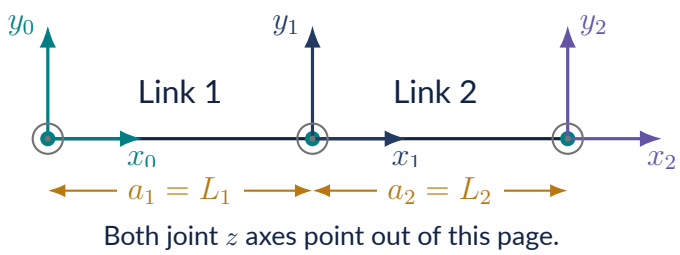

Figure. A planar 2R zero posture; all consecutive revolute z axes are parallel, so $\alpha_1 = \alpha_2 = 0$. All frame origins lie in the plane.

2.3 Derive the standard 4×4 D–H matrix

Substitute the elementary matrices into A_i and multiply. The resulting transform is

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The left 3×3 block describes orientation; the rightmost three entries give translation. The last row is always $[0\ 0\ 0\ 1]$. **A useful test:** if $\theta_i = 0$ and $\alpha_i = 0$, the matrix must represent pure translation a_i along x and d_i along z.

2.4 D–H table for the planar 2R robot

Assume two revolute joints with lengths L_1, L_2, all axes perpendicular to the plane, and a tool frame at the end of link 2. A valid standard D–H table for the chosen frames is

i	θ_i	d_i	a_i	α_i
1	q_1	0	L_1	0
2	q_2	0	L_2	0

$$A_1 = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & L_1 \cos q_1 \\ \sin q_1 & \cos q_1 & 0 & L_1 \sin q_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_2 = A_1\big|_{q_1 \mapsto q_2, L_1 \mapsto L_2}.$$

Multiply $T_0^2 = A_1 A_2$; its position column becomes $[L_1 \cos q_1 + L_2 \cos(q_1+q_2),\ L_1 \sin q_1 + L_2 \sin(q_1+q_2),\ 0]^T$. This agrees with the diagram-derived equations in Section 1. That agreement is a **validation**, not a coincidence: both describe the same chain.

STEP-BY-STEP WORKED EXAMPLE

Original D–H example: A 2R planar robot has $L_1 = 0.5$ m, $L_2 = 0.2$ m, $q_1 = 90°$, $q_2 = -90°$.
Step 1: With $d_i = \alpha_i = 0$, the transformations are rotations about z followed by translations along local x.
Step 2: Link 1 ends at $(0, 0.5)$; link 2's global angle is $0°$, so it adds $(0.2, 0)$.
Result: $\mathbf{p} = [0.2\ 0.5\ 0]^T$ m, and tool orientation $q_1 + q_2 = 0°$.
Check: The transform has $R = I$ for this orientation, which matches a tool pointing along base $+x$.

2.5 Prismatic and spatial joints: what changes?

For a prismatic joint the displacement d_i varies while θ_i is typically fixed. Suppose a Cartesian slide moves vertically: choose $a_i = \alpha_i = \theta_i = 0$ and $d_i = q_i$. Then $A_i = \text{Trans}_z(q_i)$; it moves a

point without changing orientation. For spatial arms with nonparallel axes, $\alpha_i \neq 0$ accounts for the angle between successive joint axes. Do not try to force all $\alpha_i = 0$ on a genuine 3D six-axis arm.

2.6 D–H versus geometric vectors versus PoE

Method	Strong for	Caution
Direct vector geometry	Simple planar arms with visibly known lengths and angles	Harder to manage many rotated spatial axes
Standard D–H	Repeated, tabular link-by-link manipulator modelling	Frame choices and convention mixing introduce sign errors
Product of exponentials (PoE)	Alternative axis-based spatial kinematics using screw motions	Requires understanding twist coordinates; beyond the minimal unit description

EXAM TIP / COMMON MISTAKE

High-value exam check: Write "standard D–H" at the start; show your labeled frame sketch; give the four-column table; write A_i; then compute $T_0^n = \prod_i A_i$. If your resulting transformation differs from a simple geometric case, revisit axis directions and signs before proceeding.

WRITE THIS IN THE EXAM

Model answer: Explain standard D–H convention and its four parameters. Start with why link-to-link coordinate frames are useful. Draw two adjacent frames and name θ, d, a, α with their axes. State $A_i = R_z(\theta_i)T_z(d_i)T_x(a_i)R_x(\alpha_i)$, display the standard 4×4 matrix, and demonstrate a two-link table. Explain revolute versus prismatic variables and mention the standard/modified convention distinction.

3 Inverse kinematics: tool target to joint angles

TOPIC TYPE: Geometry + equations + two-branch numerical solutions **LEARNING GOAL: Find joints that place a robot tool at a specified target.**

IN SIMPLE WORDS

In forward kinematics, you turn the joints and calculate where the hand goes. In inverse kinematics, you choose where the hand should go and calculate how the joints must turn. This reverse calculation can have two answers, one answer, no answer, or (for redundant robots) very many answers.

3.1 Exactly what is given and what is unknown?

The desired tool position $\mathbf{p}_d$ and orientation R_d are given in a *known base frame*, along with link geometry and sometimes joint limits. The unknown is $\mathbf{q}$ such that $T(\mathbf{q}) = T_d$. For a two-link planar positioning problem with no separate tool orientation constraint, the target is only (x, y) and the unknowns are (q_1, q_2). For a generic spatial pose, one must satisfy both three position and three orientation coordinates; a six-DOF arm often has up to six independent joint coordinates, but "six joints" by itself does not guarantee every pose is reachable.

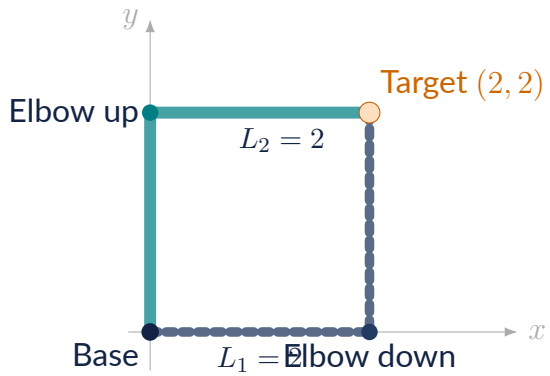

Figure. Two exact geometric postures for equal links $L_1 = L_2 = 2$ and the same target $(2, 2)$. The elbow-up path is solid; the elbow-down path is dashed. Both satisfy link lengths and forward kinematics.

3.2 Reachability: decide whether a solution can exist first

For two rigid links of lengths L_1 and L_2, let $r = \sqrt{x^2 + y^2}$ be the distance from the base to the target. The arm can reach a point in the unconstrained plane if

$$\boxed{|L_1 - L_2| \le r \le L_1 + L_2.}$$

Why? The longest reach occurs when the links point in the same direction: $r_{\text{max}} = L_1 + L_2$. The shortest reach occurs when one folds back against the other: $r_{\text{min}} = |L_1 - L_2|$. Joint limits, self-collision and obstacles can reduce the *usable* workspace further. A point within the annulus may still be forbidden by a particular real robot.

STEP-BY-STEP WORKED EXAMPLE

Reachability test: With $L_1 = 0.40\,\text{m}$ and $L_2 = 0.30\,\text{m}$, points are geometrically reachable only for $0.10 \le r \le 0.70\,\text{m}$. Target $(0.50, 0.20)$ has $r = \sqrt{0.29} \approx 0.5385\,\text{m}$ and passes. Target $(0.8, 0.1)$ has $r \approx 0.8062\,\text{m}$ and fails: no choice of joint angles will make the rigid links longer.

3.3 Derive the elbow angle from the cosine rule

Draw a triangle with sides L_1, L_2 and r. The exterior joint angle q_2 is related to the interior triangle angle by a supplement. Applying the cosine rule yields

$$r^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos q_2.$$

Rearrange, and introduce c_2 for clarity:

$$\boxed{c_2 = \cos q_2 = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}}.$$

If $|c_2| > 1$ beyond small rounding error, the target is unreachable in this ideal model. For the two elbow branches, $s_2 = \pm\sqrt{1 - c_2^2}$ and $q_2 = \text{atan2}(s_2, c_2)$; the positive and negative values correspond to the two choices of elbow bend under the chosen sign convention. Both should be calculated unless joint limits or the question excludes a branch.

3.4 Derive the shoulder angle with two smaller angles

Let $\beta = \text{atan2}(y, x)$ be the direction of the target ray from the base. Let γ be the angle between the first link and that target ray. From the right triangle formed by the link-vector sum,

$$\gamma = \text{atan2}(L_2 \sin q_2, L_1 + L_2 \cos q_2).$$

Therefore the base-joint angle is

$$\boxed{q_1 = \text{atan2}(y, x) - \text{atan2}(L_2 \sin q_2, L_1 + L_2 \cos q_2)}.$$

This is the cleanest 2R planar inverse-kinematics formula to memorize *after understanding the triangle*. For $x < 0$ or $y < 0$, using quadrant-aware atan2 avoids an incorrect $180°$ branch.

STEP-BY-STEP WORKED EXAMPLE

Fully solved original numerical: $L_1 = 0.40$ m, $L_2 = 0.30$ m, target $(x, y) = (0.50, 0.20)$ m.
Step 1 — reach: $r^2 = 0.50^2 + 0.20^2 = 0.29$; $r \approx 0.5385$ is inside $[0.10, 0.70]$.
Step 2 — elbow cosine:

$$c_2 = \frac{0.29 - 0.16 - 0.09}{2(0.4)(0.3)} = \frac{1}{6}.$$

Step 3 — both elbow angles: $q_2 = \pm\cos^{-1}(1/6) \approx \pm 80.4059°$.
Step 4 — shoulder: $\beta = \text{atan2}(0.20, 0.50) \approx 21.8014°$. Then compute the correction γ for each signed q_2 and use $q_1 = \beta - \gamma$.
Step 5 — state both branches: One branch bends the elbow positive and the other negative. Compute both values with a quadrant-aware calculator; they give different q_1 but the same target coordinates when checked by forward kinematics.
Step 6 — validation: Substitute each pair into $x = L_1 \cos q_1 + L_2 \cos(q_1 + q_2)$ and $y = L_1 \sin q_1 + L_2 \sin(q_1 + q_2)$. The reconstructed target must agree within the requested precision.

3.5 What happens at the boundaries?

At $r = L_1 + L_2$, the links become collinear and extended: $q_2 = 0$. The two elbow branches merge. At $r = |L_1 - L_2|$, they are fully folded: $q_2 = \pm 180°$ in this convention; again the branches merge. These configurations are important because the Jacobian of a 2R planar arm loses rank when $\sin q_2 = 0$. A small change in the desired tool motion may then require very large changes in joint velocities. Singularities are covered formally in Unit III; here the intuitive warning is enough.

3.6 IK with a desired orientation for a 3R planar arm

For a three-revolute-joint arm in the plane, suppose the desired tool pose is (x, y, ϕ). If the final link length L_3 is known, first compute the **wrist centre** (the joint before the last link):

$$\boxed{x_w = x - L_3 \cos\phi, \qquad y_w = y - L_3 \sin\phi.}$$

Now solve the *2R inverse kinematics problem* for (x_w, y_w) using the first two lengths L_1, L_2. Finally use $q_3 = \phi - q_1 - q_2$. This removes one joint from the position problem by using the known tool orientation. The result may again have two branches. Remember that ϕ is a total orientation; q_3 is relative to link 2.

STEP-BY-STEP WORKED EXAMPLE

3R wrist-centre example: Take $L_1 = 0.4\,\text{m}$, $L_2 = 0.3\,\text{m}$, $L_3 = 0.1\,\text{m}$, and target $(x, y, \phi) = (0.60, 0.20, 0^\circ)$. Then

$$x_w = 0.60 - 0.10\cos 0 = 0.50, \quad y_w = 0.20 - 0.10\sin 0 = 0.20.$$

The first two joints solve the same 2R problem as above; once a branch is chosen, $q_3 = -(q_1 + q_2)$. The last link points along the global $+x$ direction, so the tool position returns to $(0.60, 0.20)$ and orientation zero. This is a model of how a six-axis arm also often separates wrist-position from wrist-orientation tasks when its wrist axes intersect.

3.7 Why a closed-form IK answer may not exist

A robot can have joint limits, mechanical offsets, intersecting or parallel axes, and collision restrictions. Algebraic IK must solve nonlinear trigonometric equations and may have many discrete branches or none. Some complex manipulators use numerical search rather than one universal triangle formula. Even if the mathematics gives a solution, the controller must reject configurations violating mechanical limits or safety constraints.

EXAM TIP / COMMON MISTAKE

Five common mistakes: (1) using $\tan^{-1}(y/x)$ without quadrant checking; (2) computing just one elbow branch; (3) forgetting the reach test; (4) confusing absolute and relative joint angles; (5) solving a scanned past-paper diagram by guessing the missing frame or axis directions. Draw, define and verify before writing the final angles.

WRITE THIS IN THE EXAM

Model answer outline: Geometric inverse kinematics of a 2-DOF planar manipulator. Define IK and target (x, y). Draw the arm as a triangle, mark L_1, L_2, r, q_1, q_2, give the cosine-rule derivation for $\cos q_2$, state the $\pm$ branches, derive q_1 using the target direction minus the link-vector correction, test reach and joint limits, and finish by substituting each solution into the FK equations. A complete answer includes the *diagram*, *both formulas*, *branch explanation and validation*; a two-line memorized formula is not sufficient.

4 How to choose and apply an inverse-kinematics strategy

TOPIC TYPE: Algorithm + method comparison + practical workflow **LEARNING GOAL: Select a suitable IK method instead of applying one formula to every robot.**

IN SIMPLE WORDS

There is no single formula that solves every robot. For a simple two-link arm, draw a triangle. For a model with neatly separable position and orientation, solve parts algebraically. For a more complicated robot, repeatedly improve a guess until the tool gets close enough to the target.

4.1 The main families of solution methods

Method	How it works	Advantages and limitations
Geometrical	Uses triangles, link lengths, cosine rule and orientation decomposition	Intuitive, often exact, excellent for 2R/3R; difficult for irregular 3D geometry
Algebraic / analytic	Solves equations from a forward-kinematics model using trigonometric identities and elimination	Finds symbolic branches for suitable mechanisms; can become long and sensitive to geometry
Numerical iterative	Begins with a joint guess and updates it based on tool-position/orientation error	General and flexible; must handle initial guess, local convergence, limits and singularities
Search / optimization	Minimizes a cost function based on position error and constraints	Accommodates obstacles or joint preferences; computational cost and local minima matter

4.2 Geometrical strategy: visualizing before solving

First look for a triangle: for a planar 2R arm, the sides are L_1, L_2, and r. Next look for a *wrist centre*: in a 3R planar arm or a suitable six-axis arm with intersecting wrist axes, subtract the tool offset from the desired tool position. Once the position subproblem is solved, the remaining joints satisfy an orientation subproblem. **This is not universally valid**: spherical-wrist decoupling depends on actual robot geometry, not just on there being six joints.

4.3 Algebraic strategy: start from forward equations

Suppose you know

$$x = L_1 c_1 + L_2(c_1 c_2 - s_1 s_2), \qquad y = L_1 s_1 + L_2(s_1 c_2 + c_1 s_2),$$

where $c_i = \cos q_i$ and $s_i = \sin q_i$. Square and add $x^2 + y^2$; many cross terms cancel, giving $r^2 = L_1^2 + L_2^2 + 2L_1L_2c_2$. This elimination step is the algebraic reason the cosine-rule formula works. Then use the direction of the resultant vector to recover q_1. Algebra is helpful for checking the geometry and deriving consistent formulas for more complex mechanisms.

4.4 Numerical strategy: improve a guess repeatedly

When the desired joint angles are not easy to obtain in closed form, define a task-space error $\mathbf{e} = \mathbf{p}_d - \mathbf{p}(\mathbf{q})$. Near a current guess $\mathbf{q}_k$, a first-order approximation is

$$\mathbf{p}(\mathbf{q}_k + \Delta\mathbf{q}) \approx \mathbf{p}(\mathbf{q}_k) + J(\mathbf{q}_k)\Delta\mathbf{q},$$

where J is the Jacobian. For a square nonsingular Jacobian, choose $\Delta\mathbf{q} = J^{-1}\mathbf{e}$ and update $\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta\mathbf{q}$. If J is rectangular or ill-conditioned, a pseudoinverse or damped-least-squares method is preferred. The Jacobian is studied systematically in Unit III; here it is an *algorithmic tool* for explaining how a computer searches for the IK solution.

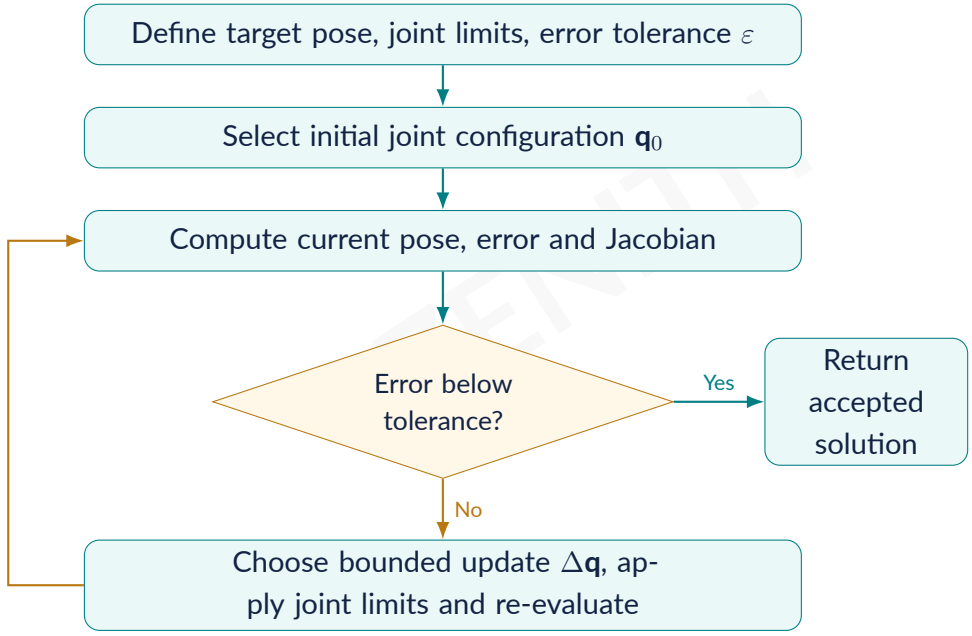

Figure. Iterative inverse-kinematics workflow, with a stopping test. Implementations also need iteration limits and failure reporting.

4.5 Damped least squares in one understandable formula

Near a singular configuration, direct inversion of the Jacobian can make a tiny desired tool movement produce a huge joint update. Damped least squares regularizes the update:

$$\boxed{\Delta\mathbf{q} = J^T\left(JJ^T + \lambda^2 I\right)^{-1}\mathbf{e}}, \qquad \lambda > 0.$$

The damping λ trades off exact correction for a smaller, more stable step. It is not an assurance of global convergence; different starting guesses can still find different postures or fail to reach a constrained target. For exam purposes, explain it as a controlled correction that avoids extreme joint motion near ill-conditioned configurations, rather than as magic "always works" software.

4.6 Joint limits and collision checks are not optional

A mathematically correct IK solution might require $q_2 = 130^\circ$ while the physical joint can only move to 90°. It may also collide with the robot's own base, a workpiece or a protective enclosure. Therefore practical IK follows this order: **geometry and equations** $\rightarrow$ **candidate solutions** $\rightarrow$ **joint range check** $\rightarrow$ **collision/clearance check** $\rightarrow$ **choose best valid solution**. When choosing between two valid branches, criteria include shortest joint travel, avoiding singularities, energy, clearance and smooth motion from the current posture.

WRITE THIS IN THE EXAM

Model answer: Compare analytical and numerical IK approaches. Define IK and why nonlinear equations arise. Explain geometric and algebraic methods with their advantage of exact multiple-branch solutions for suitable arms. Explain the numerical method: pose error, initial guess, Jacobian-based iteration and convergence tolerance. State drawbacks: initial-guess sensitivity, singularities, joint limits and local minima. Include the iterative flowchart and conclude with when each method is chosen.

QUICK RECALL

Choice rule: 2R planar $\rightarrow$ geometry. Clear wrist-centre decomposition $\rightarrow$ partly analytic. Irregular many-joint system $\rightarrow$ iterative IK with a good seed and feasibility checks. Keep the problem's geometric assumptions visible.

5 Redundancy, multiple solutions and singular cases

TOPIC TYPE: Conceptual + geometric + comparison based **LEARNING GOAL: Explain why one target may correspond to many, one or no joint configurations.**

IN SIMPLE WORDS

Two people can reach the same shelf in different ways: one keeps the elbow high, the other low. A robot can also use different joint postures to reach one target. If the robot has extra joints beyond the task's independent requirements, it may have infinitely many ways to achieve the same task while changing the unused joint motions.

5.1 Multiple discrete answers are not the same as redundancy

A 2R arm reaching a general (x, y) point often has two *isolated* solutions: elbow up and elbow down. The number of joint coordinates (two) equals the number of independent planar position coordinates (two); so the arm is not generically redundant for that task. Those two answers arise because nonlinear geometry can have several branches. By contrast, a planar 3R arm required only to hit (x, y) has three joint variables for a two-dimensional task. Away from degeneracies

and subject to limits, one joint may be varied and the others adjusted to keep the tip at the same point; the solution family is continuous.

5.2 Define task dimension and redundancy clearly

Let n be the number of independently actuated joint coordinates and m the number of independent task coordinates. For a regular task Jacobian of full row rank, if $n > m$ the arm is kinematically redundant for that *particular task*. Example: a 7R arm used for a general six-dimensional 3D end-effector pose often has one extra joint degree of freedom locally; a 6R arm used for only 3D tip position may have additional freedom because tool orientation is not specified. This statement depends on joint-axis geometry and Jacobian rank. **Do not confuse** number of motors, total geometric joint count and effective independent task motion.

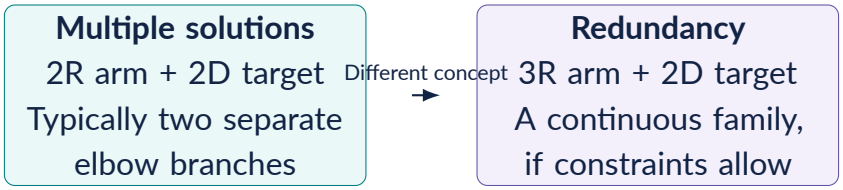

Figure. Separate geometric branches versus an extra continuous joint degree of freedom.

5.3 Why would an engineer intentionally add redundant joints?

1. **Obstacle avoidance:** move an elbow around an obstacle while holding the tool pose.
2. **Joint-limit avoidance:** choose a comfortable posture that keeps joint angles away from hard stops.
3. **Singularity avoidance:** prefer configurations with better manipulability and smaller joint-speed amplification.
4. **Secondary tasks:** keep a camera pointed toward a target, avoid a cable twist or optimize ergonomics without moving the tool.
5. **Collision clearance:** preserve tool contact while shifting the intermediate links away from the workcell boundary.

Redundancy is useful, but it does not automatically guarantee avoidance of all obstacles or singularities. Choosing a redundant motion requires an additional objective or optimization policy.

5.4 Self-motion and the Jacobian null space

For local differential motion, $\dot{\mathbf{p}} = J(\mathbf{q})\dot{\mathbf{q}}$. If $\mathbf{z}$ lies in the null space, $J\mathbf{z} = 0$, then moving the joints infinitesimally along $\mathbf{z}$ does not change the task coordinates to first order. One common local redundancy solution is

$$\boxed{\dot{\mathbf{q}} = J^{+}\dot{\mathbf{p}} + (I - J^{+}J)\mathbf{z}},$$

where J^{+} is a pseudoinverse. The first term accomplishes the tool motion, and the second term provides freedom for a secondary objective. This equation is an *extension for conceptual*

understanding; detailed Jacobian and velocity mapping appear in Unit III. The null-space idea is easier to remember by visualizing elbow repositioning while the hand stays at the chosen point.

5.5 Singular configurations: a robot may lose a direction of motion

For the 2R planar arm, the Jacobian determinant is $\det J = L_1 L_2 \sin q_2$. At $q_2 = 0$ (fully extended) or $q_2 = \pi$ (folded), $\det J = 0$. This means the local linear mapping cannot be inverted uniquely in every task-space direction. It does not mean the robot freezes: it means its instantaneous tool-motion capabilities are locally restricted and inverse velocity calculations can become ill-conditioned. At these postures, the two ordinary IK elbow branches merge.

EXAM TIP / COMMON MISTAKE

Terminology: *Unreachable* target: no feasible joint configuration exists. *Multiple solutions*: several configurations achieve a target. *Redundancy*: extra joint freedom relative to a particular task. *Singular*: instantaneous velocity map loses rank. These are related but not interchangeable words.

WRITE THIS IN THE EXAM

Model answer: Explain redundancy and multiple solutions in inverse kinematics. Define IK, draw a two-link elbow-up/elbow-down example, state why it gives two isolated branches without redundancy, then define task dimension m and joint dimension n. Explain a redundant $n > m$ example, applications (obstacle avoidance, joint limits, secondary objectives), the meaning of a null-space motion, and the need to consider physical constraints. Mention that singularities can merge discrete branches.

6 Full cases: 2-link planar and 6-DOF industrial arm

TOPIC TYPE: Full worked case studies + diagrams **LEARNING GOAL: Connect the formula to a complete robot model rather than unrelated equations.**

6.1 Case A: a two-link planar robot from start to finish

IN SIMPLE WORDS

When you meet a new 2R exam problem, do not jump straight to an inverse-angle formula. First draw the arm, write its forward model, identify the target and joint limits, check reach, calculate both elbow branches, and use forward kinematics to verify your answers.

Known physical parameters: $L_1 = 0.40\,\text{m}$; $L_2 = 0.30\,\text{m}$. Revolute joints q_1, q_2 rotate about parallel axes pointing out of the page. The end-effector is a point at the distal end of link 2, and its orientation is not separately constrained. **(i) Forward model:**

$$x = 0.40\cos q_1 + 0.30\cos(q_1+q_2), \qquad y = 0.40\sin q_1 + 0.30\sin(q_1+q_2).$$

(ii) Standard D–H table:

i	θ_i	d_i	a_i	α_i
1	q_1	0	0.40	0
2	q_2	0	0.30	0

(iii) Inverse task: desired tip at $(0.50, 0.20)$ m. First verify $r = \sqrt{0.29} \approx 0.5385$ m lies within $0.10 \le r \le 0.70$. Then calculate

$$c_2 = \frac{0.50^2 + 0.20^2 - 0.40^2 - 0.30^2}{2(0.40)(0.30)} = \frac{1}{6}, \quad s_2 = \pm\sqrt{1-(1/6)^2}.$$

(iv) Elbow branches: $q_2 = \text{atan2}(s_2, c_2)$, $q_1 = \text{atan2}(0.20, 0.50) - \text{atan2}(0.30s_2, 0.40 + 0.30c_2)$. Use the positive s_2 for one branch and the negative for the other. **(v) Check**: substitute the angles into the two forward formulas and verify that both reproduce $(0.50, 0.20)$ within the rounding tolerance. **(vi) Interpretation:** Because the desired task has only two Cartesian position coordinates, a 2R arm normally has two discrete branches rather than an extra continuous degree of freedom.

6.2 Case B: what "6-DOF industrial manipulator" means

Most articulated industrial arms contain a sequence of rotary joints at base, shoulder, elbow and wrist, although exact axes and offsets differ between manufacturers. A generic six-axis robot can independently control three position and three orientation degrees of freedom *in many regular configurations*, subject to geometry and mechanical limits. It is crucial to distinguish a **generic teaching model** from a specific manufacturer's robot: without a documented D–H or URDF model, you cannot honestly calculate the real robot's exact position or joint angles.

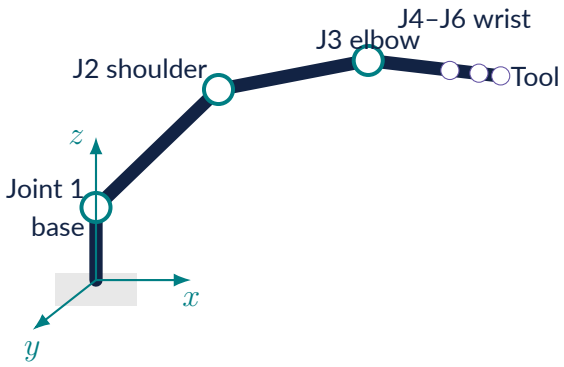

Figure. Schematic six-joint industrial arm showing a typical base–shoulder–elbow–wrist division. The joint axes are intentionally not presented as a manufacturer's D–H assignment.

6.3 An explicitly specified six-axis teaching model

To make an honest calculation, *define the exact joint transformations first*. Let the robot have base height h, upper-arm length L_a, forearm length L_b and a tool offset d. Define its illustrative model

as

$$T_0^6 = T_z(h)\, R_z(q_1)\, R_y(q_2)\, T_x(L_a)\, R_y(q_3)\, T_x(L_b) \cdot R_x(q_4)\, R_y(q_5)\, R_x(q_6)\, T_z(d).$$

Every symbol represents a homogeneous 4×4 transform. This is a **defined pedagogical kinematic chain**; it is not a claim about a named industrial robot or its standard D–H table. Joint 1 is base yaw, joints 2 and 3 bend the shoulder/elbow about local y, and joints 4–6 orient the tool. All joint transformations affect the tool position because of the final offset d.

6.4 Check this model at the home configuration

Set all six angles to zero. Each rotation becomes identity, so the chain reduces to translations $T_z(h)T_x(L_a)T_x(L_b)T_z(d)$. Consequently,

$$\boxed{\mathbf{p}_0 = \begin{bmatrix} L_a + L_b \\ 0 \\ h + d \end{bmatrix}, \qquad R_0^6 = I.}$$

Now set $q_1 = 90^\circ$ and keep all other joint angles zero. The base yaw rotates the horizontal link projections from $+x$ to $+y$, but vertical offsets remain vertical:

$$\boxed{\mathbf{p} = \begin{bmatrix} 0 \\ L_a + L_b \\ h + d \end{bmatrix}, \qquad R_0^6 = R_z(90^\circ).}$$

If $L_a = 0.35\,\text{m}$, $L_b = 0.25\,\text{m}$, $h = 0.40\,\text{m}$ and $d = 0.10\,\text{m}$, the predicted point is $(0, 0.60, 0.50)\,\text{m}$. This simple posture is a robust sanity check for a longer matrix product.

6.5 Position and orientation decoupling: when you may use it

For a robot with three intersecting wrist axes at a common point, the wrist-centre position can often be solved using the first three joints. The desired wrist-centre position is $\mathbf{p}_w = \mathbf{p}_d - R_d \mathbf{p}_{\text{tool}\leftarrow\text{wrist}}$, where the final vector is expressed in the tool orientation frame (notation must be defined for a real robot). Solve the arm for $\mathbf{p}_w$, then compute the remaining wrist rotation from $R_3^6 = (R_0^3)^T R_d$. Finally determine the wrist angles using the robot's actual rotation sequence. **Warning:** the formula and number of independent solutions depend on the specific axes. The generic six-axis teaching model above has its own defined sequence and should not be replaced by an unrelated textbook geometry.

6.6 What full six-axis inverse kinematics needs

A complete numerical inverse solution for a real 6R robot needs (1) actual link and joint frames or a URDF; (2) exact tool/base calibration; (3) target pose, not position alone unless orientation is free; (4) joint limits and branch policy; (5) collision and singularity handling; and (6) forward

verification. If a past paper shows a robot figure but the figure is missing, then neither the joint ordering nor correct angles can be recovered from link lengths alone. Mark it as *requires original figure* instead of inventing them.

WRITE THIS IN THE EXAM

Model answer: Describe a 6-DOF industrial manipulator and its kinematic analysis. Draw a labeled base–shoulder–elbow–wrist schematic. Define joint variables and the six-DOF pose. Explain coordinate frames and that FK multiplies link transforms from base to tool. Show a clearly labeled *illustrative* six-joint transformation chain. For IK, explain finding joint configurations matching desired position and orientation, possible wrist-centre decoupling, multiple branches and geometric/joint-limit constraints. State that exact solutions need the actual robot geometry.

7 Verified older-paper questions with model answers

TOPIC TYPE: Exam-writing + evidence based **LEARNING GOAL: Write the complete method, not only a final equation.**

SOURCE AND SCOPE

These are **verified topic-matched questions from earlier 2019 Pattern SPPU papers**, not a prediction of the 2024 Pattern question arrangement. The original documents, question numbers and physical pages were checked. Model answers are fresh educational explanations, not official university solutions. When the paper requires a robot diagram that is absent from extracted text, this guide does not invent it.

7.1 2019 Pattern paper 6583-177, November 2025, Q1(a), physical p. 1, 9 marks

Question summary: Explain inverse kinematics for a two-degree-of-freedom robot by the geometric method. Open the original 2019 Pattern paper.

Exam-ready answer:

1. Define *inverse kinematics*: finding joint coordinates from a desired tool configuration using known robot geometry, without computing forces/torques.
2. Draw a planar two-link RR robot. Mark base O, elbow A, tool $P = (x, y)$, lengths L_1, L_2, relative joints q_1, q_2 and distance $r = \sqrt{x^2 + y^2}$.
3. State reach condition $|L_1 - L_2| \leq r \leq L_1 + L_2$ and explain that a target failing it has no geometric solution.
4. Use the cosine rule $r^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos q_2$ to obtain $c_2 = (r^2 - L_1^2 - L_2^2)/(2L_1L_2)$.

5. Both branches are $q_2 = \text{atan2}(\pm\sqrt{1-c_2^2}, c_2)$, except where the branches merge at a boundary.
6. Let $\beta = \text{atan2}(y, x)$ and $\gamma = \text{atan2}(L_2 \sin q_2, L_1 + L_2 \cos q_2)$; then $q_1 = \beta - \gamma$.
7. Verify by substituting both pairs into the FK formulas; reject any branch violating joint limits or other physical constraints.

Final diagram instruction: Include the triangle and two elbow postures. The derivation shows *why* two solutions exist, rather than merely stating a memorized formula.

7.2 2019 Pattern paper 6583-177, November 2025, Q2(a), physical p. 1, 9 marks

Question summary: Distinguish forward and inverse kinematics in robotic motion. Original paper.

Point	Forward kinematics (FK)	Inverse kinematics (IK)
Given	Joint coordinates and link geometry	Desired tool pose and link geometry
Find	Tool position and orientation	Joint coordinates that achieve target
Direction	Joints $\longrightarrow$ pose	Pose $\longrightarrow$ joints
Typical difficulty	Straightforward chain of matrices	Nonlinear; can have zero, one, many solutions
2R illustration	$x = L_1c_1 + L_2c_{12}$	$c_2 = (x^2 + y^2 - L_1^2 - L_2^2)/(2L_1L_2)$
Use	Feedback position estimation, simulation, visualizing a posture	Pick-and-place positioning, robot path execution, task-space commands
Verification	Confirm physical dimensions and frame signs	Substitute IK output back into FK; check reach and joint limits

Concluding explanation: Both methods use the same geometry. FK is usually unique for a fully specified joint configuration and model, whereas IK is generally not unique. Neither method on its own determines drive forces or actuator torque, which belong to dynamics.

7.3 2019 Pattern paper 6403-244, May 2025, Q2(a), physical p. 1, 8 marks

Question summary: Define the inverse-kinematics problem and explain approaches to solve it. Original paper.

Model answer: State $T(\mathbf{q}) = T_d$ with unknown $\mathbf{q}$, explain why trigonometric constraints and joint limits make the problem difficult, then describe: (1) geometric triangles for planar/decoupled models, (2) algebraic elimination of FK equations, (3) numerical iterations using pose error and Jacobian, and (4) optimization when constraints or posture preferences matter. For each,

state a benefit and limitation. Finish with a small method-selection flowchart and the need for validation. **Do not solve the figure-dependent Q2(b)** unless the original robot image and frames have been inspected.

7.4 2019 Pattern paper 6262-175, April 2024, Q1(b), physical p. 1, 9 marks

Question summary: Explain the fundamental difference between FK and IK. Original paper.

Model answer: Give the input/output definitions with a labeled manipulator sketch, show the FK expressions for a 2R robot, then derive the inverse cosine rule and indicate both elbow branches. Include differences in uniqueness, computational complexity and use in robot control. State that real solutions must respect geometry and joint limits. A comparison table plus an example is stronger than two definitions alone.

8 Practice bank and answer-check prompts

How this section works: The prompts below are original practice, not claimed previous-paper quotations. The numbers are organizational tags; they are not promises of marks or SPPU section numbers. For theory answers, use the topic chapters for full detail. For calculations, the separate companion gives more examples and independent checks.

8.1 Definitions and diagram questions

T1. Define forward kinematics. Give inputs, outputs and one application. *Check: joint vector, link geometry, tool pose, example.*

T2. Define inverse kinematics. Why can two robot postures give the same target? *Check: nonlinear geometry, elbow branches, figure.*

T3. Draw a planar 2R robot and distinguish q_2 from the global orientation $q_1 + q_2$.

T4. Explain the position/orientation distinction using a 3D homogeneous pose matrix.

T5. List and explain four standard D–H parameters with directions and units.

T6. Draw successive D–H frames and state the six frame-assignment steps.

T7. Write the classical D–H transform formula and its 4×4 matrix. Why must convention be specified?

T8. Contrast revolute and prismatic D–H variables.

T9. Compare direct vector geometry, D–H and product-of-exponentials representation.

T10. Explain the difference between IK branches and redundant IK solutions.

T11. Describe when a robot's geometry allows wrist-centre separation.

T12. Draw a six-joint industrial arm; identify base, shoulder, elbow, wrist and tool.

8.2 Calculations and derivations

N1. Derive x, y for a planar 2R robot by vector addition.

N2. Derive the 3R planar x, y, ϕ relations.

N3. For $L_1 = 0.5$ m, $L_2 = 0.2$ m, $q_1 = 90°$, $q_2 = -90°$, find the tool position. *Check:* $(0.2, 0.5)$ *m.*

N4. A 2R arm has $L_1 = 0.4$ m, $L_2 = 0.3$ m. Determine whether the targets at radial distances 0.09, 0.5 and 0.71 m are reachable. *Check: no, yes, no.*

N5. Derive $\cos q_2$ using the cosine rule, stating the length units.

N6. Derive the base angle formula using β and γ and explain quadrant-awareness.

N7. Solve IK for $(x, y) = (0.50, 0.20)$ m with $L_1 = 0.40$ m, $L_2 = 0.30$ m, including both branches and FK verification.

N8. For a planar 3R arm with $L_3 = 0.1$ m and $\phi = 0$, derive the wrist centre for target $(0.6, 0.2)$ m.

N9. Write the standard D–H table for a planar 2R arm and show the product $A_1 A_2$ gives the same FK model.

N10. Explain how $T_z(h) R_z(90°) T_x(L)$ affects a point at the origin.

N11. Differentiate the 2R position equations to obtain the 2×2 velocity Jacobian; identify the condition that makes its determinant zero (preview of Unit III).

N12. Given a near-singular Jacobian, explain why a damped update is safer than a direct inverse.

8.3 Analytical reasoning and exam-length answers

E1. Explain the full algorithm from target-pose definition to joint-limit and collision check.

E2. Compare geometric, algebraic and iterative IK strategies using examples.

E3. Show why a 2R robot has two discrete solutions but is not generically redundant for a 2D positioning task.

E4. Explain how redundancy can avoid obstacles while maintaining a tool position.

E5. Describe two ways an IK algorithm should report failure, distinguishing unreachable targets from iteration timeouts.

E6. With a six-axis robot diagram, explain FK, IK, tool pose and why real machine geometry must be verified.

EXAM TIP / COMMON MISTAKE

Authenticity rule: Do not claim the original 2019 Pattern papers above were set under the 2024 Pattern. Do not assign page numbers to a textbook or question unless the original source and its physical PDF location have been checked. The six axis example in this guide is explicitly original, not a disguised manufacturer model.

9 One-night revision: formulas, flow and mistakes

QUICK RECALL

Ten-minute retrieval drill: (1) Draw a 2R arm from memory. (2) Write forward x, y. (3) Write its standard D–H table. (4) Derive c_2 using the cosine rule. (5) Explain elbow up/down. (6) Write the atan2 shoulder equation. (7) Name two IK solution methods. (8) Explain redundancy in one sentence. (9) Write the reach condition. (10) State the check to reject impossible results.

9.1 Formula sheet — use the declared conventions

Planar 2R FK:
$$x = L_1 \cos q_1 + L_2 \cos(q_1 + q_2),$$
$$y = L_1 \sin q_1 + L_2 \sin(q_1 + q_2),$$
$$\phi = q_1 + q_2.$$

Planar 3R FK:
$$x = \sum_{i=1}^{3} L_i \cos\Big(\sum_{j=1}^{i} q_j\Big),$$
$$y = \sum_{i=1}^{3} L_i \sin\Big(\sum_{j=1}^{i} q_j\Big),$$
$$\phi = q_1 + q_2 + q_3.$$

2R IK:
$$r^2 = x^2 + y^2, \quad |L_1 - L_2| \le r \le L_1 + L_2,$$
$$c_2 = \frac{r^2 - L_1^2 - L_2^2}{2L_1L_2}, \quad s_2 = \pm\sqrt{1 - c_2^2},$$
$$q_2 = \text{atan2}(s_2, c_2),$$
$$q_1 = \text{atan2}(y, x) - \text{atan2}(L_2 s_2, L_1 + L_2 c_2),$$
$$x_w = x - L_3 \cos\phi, \quad y_w = y - L_3 \sin\phi, \quad q_3 = \phi - q_1 - q_2.$$

Standard D–H:
$$A_i = R_z(\theta_i) T_z(d_i) T_x(a_i) R_x(\alpha_i),$$
$$T_0^n = A_1 A_2 \cdots A_n, \qquad T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}.$$

9.2 Six common exam slips and fast corrections

Mistake	Correction
Treating q_2 as global link angle	Second-link direction is $q_1 + q_2$
Using $\tan^{-1}(y/x)$ blindly	Use quadrant-aware $\text{atan2}(y, x)$
Taking one value of $\cos^{-1}$ only	State $s_2 = \pm\sqrt{1 - c_2^2}$ and both branches
Ignoring reach or joint limits	Test feasibility before accepting any result
Mixing standard/modified D–H	State the convention and use one consistent transform
Assuming six joints means any pose is possible	Check joint geometry, limits, singularity and collision

9.3 How to write a clear long-form exam answer

Write in five stages: **definition** (one or two lines), **labeled diagram** (axis, angles, lengths), **derivation** (not just the final boxed formula), **worked check or practical application**, and **limitations** (reach, branches, limits, convention). The expected answer length depends on the real question; do not assume a fixed mark split from older papers.

10 Sources and independent verification

1. SPPU, TE Robotics & Automation, 2024 Pattern syllabus, course PCC303-ROA, printed page 19: authoritative list of Unit II topics and course details.
2. Lynch, K. M. and Park, F. C., *Modern Robotics*, Cambridge University Press (2017), chapters on rigid-body motion, forward kinematics, inverse kinematics, and D–H in Appendix C. Official companion material; used as a conceptual cross-check, without copying book figures.
3. Craig, J. J., *Introduction to Robotics: Mechanics and Control* (reference book listed in syllabus); used for the standard D–H and manipulator modelling concepts.
4. SPPU older paper 6583-177, November 2025, 2019 Pattern, physical p. 1, Q1(a) and Q2(a).
5. SPPU older paper 6403-244, May 2025, 2019 Pattern, physical p. 1, Q2(a). Figure-dependent questions excluded from numerical answers unless original geometry is verified.
6. SPPU older paper 6262-175, April 2024, 2019 Pattern, physical p. 1, Q1(b).

Status: Student-created study guide, not an official SPPU answer key. Calculations and published files are subject to separate mathematical, PDF and Drive QA. Versioned source and test scripts are part of the Zenith Notes Engine package.