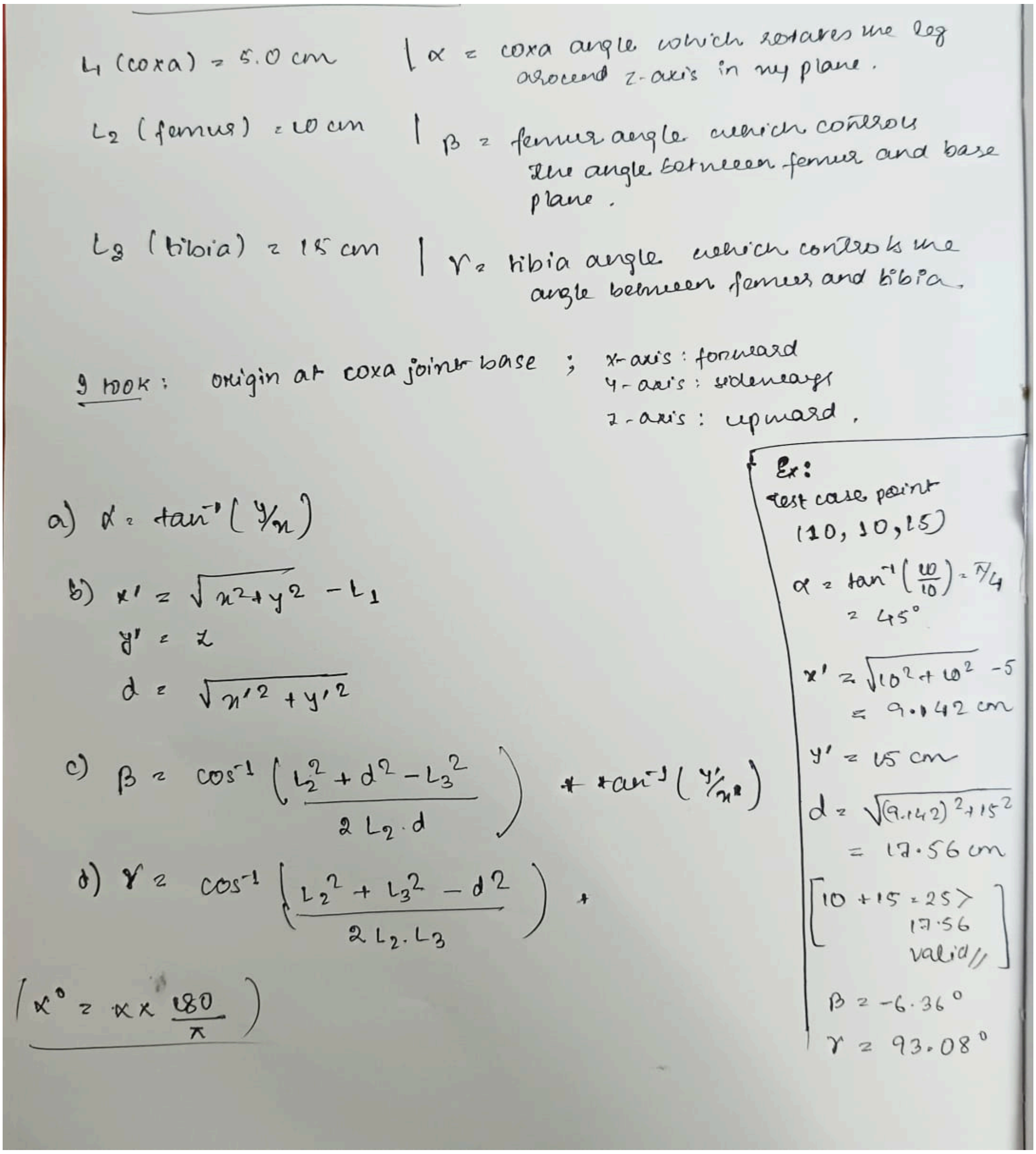
Explanation and my approach for the solution

Here we used inverse kinematics concept to determine the joint angles required to position an end effector (in this case, the foot of the hexapod leg) at a desired location in 3D space. That is starting from the desired end effector position we calculated what are the angles needed.



L_1 (coxa) = 5.0 cm | α = coxa angle which rotates the leg around z-axis in xy plane.

L_2 (femur) = 10 cm | β = femur angle which controls the angle between femur and base plane.

L_3 (tibia) = 15 cm | γ = tibia angle which controls the angle between femur and tibia.

I took: origin at coxa joint base ; x-axis : forward
y-axis : sideways
z-axis : upward.

a) $\alpha = \tan^{-1}(y/x)$

b) $x' = \sqrt{x^2+y^2} - L_1$

$y' = z$

$d = \sqrt{x'^2 + y'^2}$

c) $\beta = \cos^{-1}\left(\dfrac{L_2^2 + d^2 - L_3^2}{2 L_2 \cdot d}\right) + \tan^{-1}(y'/x')$

d) $\gamma = \cos^{-1}\left(\dfrac{L_2^2 + L_3^2 - d^2}{2 L_2 \cdot L_3}\right)$ +

$\left(\alpha^\circ = \alpha \times \dfrac{180}{\pi}\right)$

Ex:
test case point
(10, 10, 15)

$\alpha = \tan^{-1}\left(\frac{10}{10}\right) = \pi/4$
$= 45^\circ$

$x' = \sqrt{10^2 + 10^2} - 5$
≈ 9.142 cm

$y' = 15$ cm

$d = \sqrt{(9.142)^2 + 15^2}$
$= 17.56$ cm

[10 + 15 = 25 > 17.56 valid//]

$\beta = -6.36^\circ$

$\gamma = 93.08^\circ$

In the code function takes a desired 3D foot position as input and outputs the three joint angles (coxa, femur, tibia) needed to reach that position.

The function:

1. Verifies the position is physically reachable by the leg
2. Calculates the coxa angle (α) for rotation in the horizontal plane.
3. Transforms coordinates to account for the coxa position.
4. Uses trigonometry and the Law of Cosines to calculate the femur (β) and tibia (γ) angles.

So basically there are defined two functions one which calculates and the other checks the given five different scenarios. The first function follows following logical sequence:
1.Initial Reachability Check: firstly it calculates the distance from origin to target point then compares with maximum possible reach and returns none if unreachable.
2.Coxa Angle Calculation
3.coordinate transformation with x prime and h.
4.Secondary Reachability Check: checks if the remaining distance can be reached by femur and tibia.
5. Femur angle calculation where offset angle was also added looking into the elevation of target point.
6.Tibia angle calculation. It was also ensured the value lies in domain from-1 to 1

7.Output Processing :Converts all angles from radians to degrees and rounds to two decimal places then returns the three angles as a tuple

After this the testing function follows the structure:

Defines Test Cases:

Five distinct scenarios as specified in requirements

Each case tests a different aspect of the algorithm

Processes Each Test:

Calls the inverse_kinematics function

Formats and displays the results

Shows whether the position is reachable or not

Output Formatting:

Provides clear, labeled output for each test

Shows input coordinates, calculated angles, and reachability status