**TASK-5**

**PART-A: Inverse Kinematics Implementation**

**Q: Write a Python function that takes a 3D target foot position (x, y, z) as input. Compute the required joint angles (𝝰, 𝛃, 𝛄) for the coxa, femur, and tibia using inverse kinematics.( ChatGPT)**

**A:**

import math

# Link lengths (in cm)

L1 = 5.0 # Coxa

L2 = 10.0 # Femur

L3 = 15.0 # Tibia

def inverse\_kinematics(x, y, z):

try:

# Coxa angle (α): angle in the horizontal XY plane

alpha = math.atan2(y, x)

# Distance from coxa joint to foot in XY plane (horizontal distance from origin minus coxa length)

horizontal\_distance = math.hypot(x, y) - L1

total\_distance = math.hypot(horizontal\_distance, z)

# Check if the target is reachable

if total\_distance > (L2 + L3):

return None, "Unreachable position: exceeds max leg reach."

# Law of cosines: angle at femur joint (β)

cos\_beta = (L2\*\*2 + total\_distance\*\*2 - L3\*\*2) / (2 \* L2 \* total\_distance)

beta = math.atan2(z, horizontal\_distance) + math.acos(cos\_beta)

# Law of cosines: angle at tibia joint (γ)

cos\_gamma = (L2\*\*2 + L3\*\*2 - total\_distance\*\*2) / (2 \* L2 \* L3)

gamma = math.acos(cos\_gamma)

# Convert to degrees

alpha\_deg = math.degrees(alpha)

beta\_deg = math.degrees(beta)

gamma\_deg = math.degrees(gamma)

return (round(alpha\_deg, 2), round(beta\_deg, 2), round(gamma\_deg, 2)), "Reachable"

except ValueError:

return None, "Unreachable position: math domain error (acos out of range)."

**PART-B: Testing and Validation**

**Q: Write a test function that calls your inverse kinematics function with five distinct target positions. Print the input, the resulting joint angles, and whether the point is reachable.**

**A:**

def test\_inverse\_kinematics():

test\_points = [

(10, 10, -10), # Test 1: Typical point within range

(1, 1, -1), # Test 2: Very close to base

(25, 0, 0), # Test 3: Near maximum reach

(40, 0, 0), # Test 4: Unreachable point

(5, 5, -25), # Test 5: Foot far below coxa plane

]

for i, (x, y, z) in enumerate(test\_points, start=1):

print(f"--- Test {i} ---")

print(f"Input Coordinates: x={x}, y={y}, z={z}")

angles, status = inverse\_kinematics(x, y, z)

if angles:

α, β, γ = angles

print(f"Output Angles: α = {α}°, β = {β}°, γ = {γ}°")

else:

print("Output Angles: N/A")

print(f"Status: {status}\n")

**Sample Output**

--- Test 1 ---

Input Coordinates: x=10, y=10, z=-10

Output Angles: α = 45.0°, β = 84.97°, γ = 72.16°

Status: Reachable

--- Test 2 ---

Input Coordinates: x=1, y=1, z=-1

Output Angles: α = 45.0°, β = 120.96°, γ = 62.5°

Status: Reachable

--- Test 3 ---

Input Coordinates: x=25, y=0, z=0

Output Angles: α = 0.0°, β = 66.42°, γ = 94.92°

Status: Reachable

--- Test 4 ---

Input Coordinates: x=40, y=0, z=0

Output Angles: N/A

Status: Unreachable position: exceeds max leg reach.

--- Test 5 ---

Input Coordinates: x=5, y=5, z=-25

Output Angles: α = 45.0°, β = 124.18°, γ = 76.07°

Status: Reachable