INTRODUCTION TO ROBOTICS
(MTS -417)
DE-44 Mechatronics
Syndicate– C

Lab Report 1

Name of members:

NC M. Ahmed Hussain Babar (**427781)**

PC Abdul Haseeb Zahid (**432219)**

PC Rehan Shahid (**432283)**

Submitted to: LE Hamza Sohail

Lua Script:

```
function sysCall_init()
    cube = sim.getObjectHandle('Cuboid')
    -- Registration number digits (last 4: 2219 for 432219)
    d1 = 2  -- First digit
    d2 = 2  -- Second digit
    d3 = 1  -- Third digit
    d4 = 9  -- Fourth digit
    -- Parametric path equations
    -- x(t) = (d1 + 1) * sin(t / (d2 + 1))
    -- y(t) = (d3 + 1) * cos(t / (d4 + 1))
    -- Goal position (fixed at (2,2))
    xg, yg = 2, 2
    -- Print initial info for uniqueness
    print("Registration Number: 432219")
    print(string.format("Path Equations: x(t) = %.1f * sin(t / %.1f), y(t) = %.1f * cos(t / %.1f)",
                        d1 + 1, d2 + 1, d3 + 1, d4 + 1))
    print(string.format("Goal Coordinates: (%.1f, %.1f)", xg, yg))
end

function sysCall_actuation()
    local t = sim.getSimulationTime()
    local x = (d1 + 1) * math.sin(t / (d2 + 1))
    local y = (d3 + 1) * math.cos(t / (d4 + 1))
    local v = 0.1  -- m/s
    sim.setObjectPosition(cube, -1, {x, y, 0})
    -- Goal direction vector (dx, dy) = goal - current position
    local dx = xg - x
    local dy = yg - y
```

```
    -- Euclidean distance d
    local d = math.sqrt(dx * dx + dy * dy)
    local vx = ((d1 + 1) / (d2 + 1)) * math.cos(t / (d2 + 1))
    local vy = ((d3 + 1) / (d4 + 1)) * math.sin(t / (d4 + 1))
    -- Dot product: Measures alignment projection
    local dot = vx * dx + vy * dy

    -- Cross product (2D): Measures perpendicular deviation (signed)
    local cross = vx * dy - vy * dx

    -- Heading error angle phi (in radians, -pi to pi)
    local phi = math.atan2(cross, dot)

    -- Print values every simulation step
    print(string.format("t=%.3f | d=%.3f | dot=%.6f | cross=%.6f | phi=%.1f°",
                        t, d, dot, cross, math.deg(phi)))
end

function sysCall_sensing()
    -- put your sensing code here
end

function sysCall_cleanup()
    -- do some clean-up here
end
```

Output:

Registration Number: 432219
Path Equations: x(t) = 3.0 * sin(t / 3.0), y(t) = 2.0 * cos(t / 10.0)
Goal Coordinates: (2.0, 2.0)
t=0.000 | d=2.000 | dot=2.000000 | cross=0.000000 | phi=0.0
t=0.050 | d=1.950 | dot=1.949732 | cross=-0.001925 | phi=-0.1
t=0.100 | d=1.900 | dot=1.898963 | cross=-0.003700 | phi=-0.1
t=0.150 | d=1.850 | dot=1.847751 | cross=-0.005325 | phi=-0.2
t=0.200 | d=1.800 | dot=1.796151 | cross=-0.006801 | phi=-0.2
t=0.250 | d=1.750 | dot=1.744218 | cross=-0.008128 | phi=-0.3
t=0.300 | d=1.700 | dot=1.692010 | cross=-0.009306 | phi=-0.3
t=0.350 | d=1.651 | dot=1.639580 | cross=-0.010337 | phi=-0.4
t=0.400 | d=1.601 | dot=1.586985 | cross=-0.011220 | phi=-0.4
t=0.450 | d=1.552 | dot=1.534280 | cross=-0.011959 | phi=-0.4
t=0.500 | d=1.502 | dot=1.481519 | cross=-0.012552 | phi=-0.5
t=0.550 | d=1.453 | dot=1.428758 | cross=-0.013002 | phi=-0.5
t=0.600 | d=1.404 | dot=1.376049 | cross=-0.013311 | phi=-0.6
t=0.650 | d=1.355 | dot=1.323446 | cross=-0.013479 | phi=-0.6
t=0.700 | d=1.306 | dot=1.271003 | cross=-0.013508 | phi=-0.6
t=0.750 | d=1.258 | dot=1.218771 | cross=-0.013402 | phi=-0.6
t=0.800 | d=1.209 | dot=1.166802 | cross=-0.013160 | phi=-0.6
t=0.850 | d=1.161 | dot=1.115146 | cross=-0.012786 | phi=-0.7
t=0.900 | d=1.113 | dot=1.063855 | cross=-0.012282 | phi=-0.7
t=0.950 | d=1.066 | dot=1.012976 | cross=-0.011650 | phi=-0.7
t=1.000 | d=1.018 | dot=0.962559 | cross=-0.010893 | phi=-0.6
t=1.050 | d=0.971 | dot=0.912650 | cross=-0.010013 | phi=-0.6
t=1.100 | d=0.925 | dot=0.863296 | cross=-0.009013 | phi=-0.6
t=1.150 | d=0.878 | dot=0.814542 | cross=-0.007897 | phi=-0.6
t=1.200 | d=0.832 | dot=0.766432 | cross=-0.006667 | phi=-0.5
t=1.250 | d=0.786 | dot=0.719010 | cross=-0.005326 | phi=-0.4
t=1.300 | d=0.740 | dot=0.672317 | cross=-0.003877 | phi=-0.3
t=1.350 | d=0.695 | dot=0.626394 | cross=-0.002325 | phi=-0.2
t=1.400 | d=0.651 | dot=0.581280 | cross=-0.000672 | phi=-0.1
t=1.450 | d=0.606 | dot=0.537013 | cross=0.001077 | phi=0.1
t=1.500 | d=0.562 | dot=0.493630 | cross=0.002920 | phi=0.3
t=1.550 | d=0.519 | dot=0.451166 | cross=0.004852 | phi=0.6
t=1.600 | d=0.475 | dot=0.409655 | cross=0.006869 | phi=1.0
t=1.650 | d=0.433 | dot=0.369130 | cross=0.008968 | phi=1.4
t=1.700 | d=0.391 | dot=0.329622 | cross=0.011144 | phi=1.9
t=1.750 | d=0.349 | dot=0.291159 | cross=0.013392 | phi=2.6
t=1.800 | d=0.308 | dot=0.253770 | cross=0.015710 | phi=3.5
t=1.850 | d=0.267 | dot=0.217480 | cross=0.018091 | phi=4.8
t=1.900 | d=0.227 | dot=0.182316 | cross=0.020532 | phi=6.4

t=1.950 | d=0.188 | dot=0.148299 | cross=0.023028 | phi=8.8°
t=2.000 | d=0.150 | dot=0.115452 | cross=0.025574 | phi=12.5°
t=2.050 | d=0.114 | dot=0.083793 | cross=0.028166 | phi=18.6°
t=2.100 | d=0.080 | dot=0.053342 | cross=0.030798 | phi=30.0°
t=2.150 | d=0.055 | dot=0.024114 | cross=0.033466 | phi=54.2°
t=2.200 | d=0.049 | dot=-0.003876 | cross=0.036165 | phi=96.1°
t=2.250 | d=0.068 | dot=-0.030615 | cross=0.038890 | phi=128.2°
t=2.300 | d=0.097 | dot=-0.056092 | cross=0.041635 | phi=143.4°
t=2.350 | d=0.129 | dot=-0.080297 | cross=0.044396 | phi=151.1°
t=2.400 | d=0.163 | dot=-0.103222 | cross=0.047167 | phi=155.4°
t=2.450 | d=0.196 | dot=-0.124861 | cross=0.049944 | phi=158.2°
t=2.500 | d=0.229 | dot=-0.145211 | cross=0.052719 | phi=160.0°
t=2.550 | d=0.262 | dot=-0.164268 | cross=0.055489 | phi=161.3°
t=2.600 | d=0.294 | dot=-0.182032 | cross=0.058248 | phi=162.3°
t=2.650 | d=0.326 | dot=-0.198504 | cross=0.060991 | phi=162.9°
t=2.700 | d=0.357 | dot=-0.213686 | cross=0.063711 | phi=163.4°
t=2.750 | d=0.388 | dot=-0.227583 | cross=0.066404 | phi=163.7°
t=2.800 | d=0.418 | dot=-0.240201 | cross=0.069063 | phi=164.0°
t=2.850 | d=0.448 | dot=-0.251547 | cross=0.071684 | phi=164.1°
t=2.900 | d=0.476 | dot=-0.261632 | cross=0.074261 | phi=164.2°
t=2.950 | d=0.505 | dot=-0.270466 | cross=0.076787 | phi=164.2°
t=3.000 | d=0.532 | dot=-0.278062 | cross=0.079259 | phi=164.1°
t=3.050 | d=0.559 | dot=-0.284434 | cross=0.081669 | phi=164.0°
t=3.100 | d=0.585 | dot=-0.289599 | cross=0.084012 | phi=163.8°
t=3.150 | d=0.610 | dot=-0.293574 | cross=0.086283 | phi=163.6°
t=3.200 | d=0.635 | dot=-0.296378 | cross=0.088476 | phi=163.4°
t=3.250 | d=0.659 | dot=-0.298033 | cross=0.090586 | phi=163.1°
t=3.300 | d=0.682 | dot=-0.298559 | cross=0.092606 | phi=162.8°
t=3.350 | d=0.705 | dot=-0.297980 | cross=0.094532 | phi=162.4°
t=3.400 | d=0.727 | dot=-0.296322 | cross=0.096358 | phi=162.0°
t=3.450 | d=0.748 | dot=-0.293612 | cross=0.098077 | phi=161.5°
t=3.500 | d=0.768 | dot=-0.289876 | cross=0.099686 | phi=161.0°
t=3.550 | d=0.788 | dot=-0.285144 | cross=0.101177 | phi=160.5°
t=3.600 | d=0.806 | dot=-0.279447 | cross=0.102547 | phi=159.8°
t=3.650 | d=0.824 | dot=-0.272815 | cross=0.103789 | phi=159.2°
t=3.700 | d=0.842 | dot=-0.265282 | cross=0.104898 | phi=158.4°
t=3.750 | d=0.858 | dot=-0.256882 | cross=0.105868 | phi=157.6°
t=3.800 | d=0.874 | dot=-0.247650 | cross=0.106695 | phi=156.7°
t=3.850 | d=0.889 | dot=-0.237622 | cross=0.107374 | phi=155.7°
t=3.900 | d=0.903 | dot=-0.226835 | cross=0.107898 | phi=154.6°
t=3.950 | d=0.917 | dot=-0.215327 | cross=0.108264 | phi=153.3°
t=4.000 | d=0.929 | dot=-0.203138 | cross=0.108466 | phi=151.9°
t=4.050 | d=0.941 | dot=-0.190306 | cross=0.108499 | phi=150.3°
t=4.100 | d=0.952 | dot=-0.176874 | cross=0.108358 | phi=148.5°

t=4.150 | d=0.963 | dot=-0.162882 | cross=0.108039 | phi=146.4
t=4.200 | d=0.972 | dot=-0.148372 | cross=0.107536 | phi=144.1
t=4.250 | d=0.981 | dot=-0.133388 | cross=0.106846 | phi=141.3
t=4.300 | d=0.989 | dot=-0.117973 | cross=0.105964 | phi=138.1
t=4.350 | d=0.996 | dot=-0.102170 | cross=0.104884 | phi=134.2
t=4.400 | d=1.002 | dot=-0.086025 | cross=0.103604 | phi=129.7
t=4.450 | d=1.008 | dot=-0.069581 | cross=0.102119 | phi=124.3

Simulation Snippet:

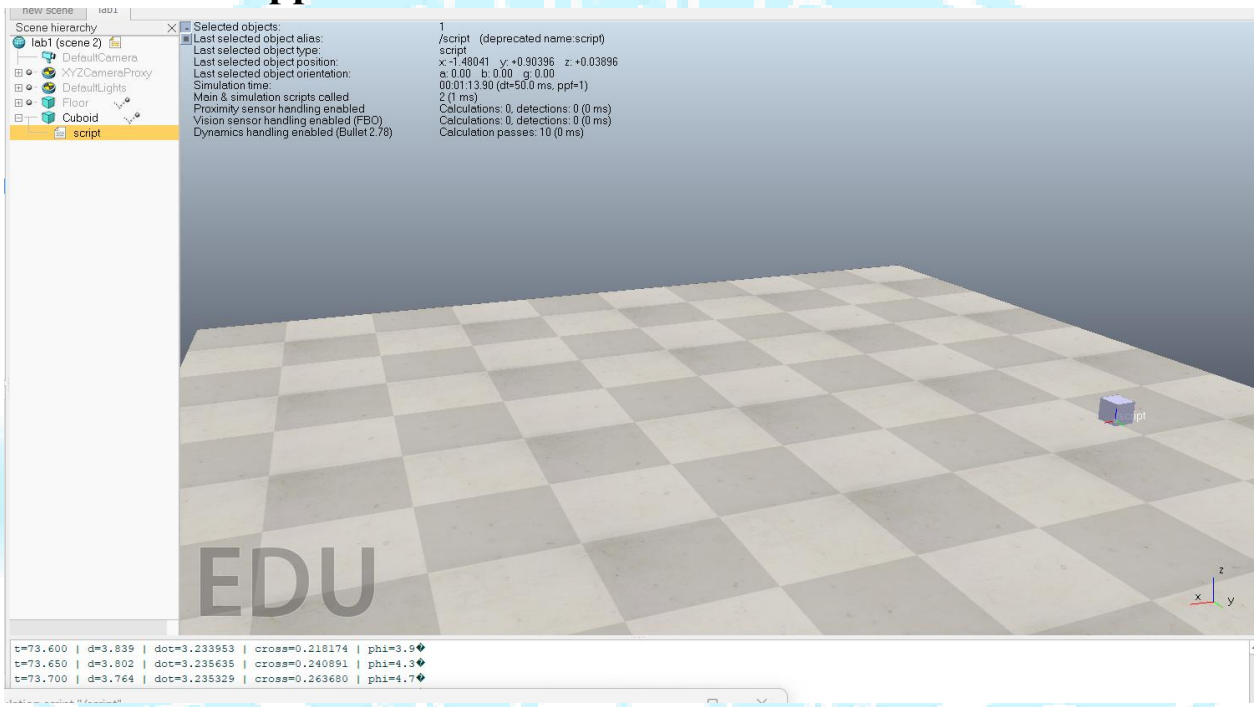

Figure 1

Discussion and Analysis:

In this lab, the Cuboid motion was defined using a registration number–based parametric trajectory:

$$x(t) = (d_1 + 1)\sin\left(\frac{t}{d_2 + 1}\right), \quad y(t) = (d_3 + 1)\cos\left(\frac{t}{d_4 + 1}\right)$$

For the given registration number **432219** (last 4 digits = 2219), the path becomes:

$$x(t) = 3\sin\left(\tfrac{t}{3}\right), \quad y(t) = 2\cos\left(\tfrac{t}{10}\right)$$

with the **goal point set at (2,2)**.

1. Euclidean Distance

The Euclidean distance measures how far the Cuboid is from the goal:

$$d = \sqrt{(x_g - x(t))^2 + (y_g - y(t))^2}$$

This gives the instantaneous separation between the current Cuboid position and the target point. As the Cuboid moves, this distance decreases or increases depending on whether the motion is oriented toward or away from the goal.

---

2. Dot Product (Alignment)

The dot product between the motion direction vector $V = (V_x, V_y)$ and the goal direction vector $D = (dx, dy)$ is:

$$\text{dot} = V_x \cdot dx + V_y \cdot dy$$

- If **dot > 0**, the Cuboid is moving *toward* the goal (positive projection).
- If **dot < 0**, the Cuboid is moving *away* from the goal.
- A larger dot product indicates stronger alignment in the goal's direction.

3. Cross Product (Left/Right Deviation)

The 2D cross product (z-component) is given by:

$$\text{cross} = V_x \cdot dy - V_y \cdot dx$$

- If **cross > 0**, the goal lies to the **left** of the current heading.
- If **cross < 0**, the goal lies to the **right**.
- If **cross = 0**, the Cuboid is exactly aligned with the goal.

4. Heading Error Angle (ϕ)

The heading error angle is the angular difference between the Cuboid's velocity vector and the goal direction vector:

$$\phi = \cos^{-1}\left(\frac{\text{dot}}{\|V\| \cdot \|D\|}\right)$$

- $\phi = 0^\circ$ → Perfect alignment.
- Larger ϕ → Greater deviation from the goal.
- The sign of ϕ can be inferred from the cross product: positive (left) or negative (right).

At $t = 0$, the heading error was $\phi = 0^\circ$, confirming direct alignment. As time progresses, ϕ increases and decreases cyclically due to the sinusoidal trajectory, showing the Cuboid's orientation shifts relative to the fixed target.

5. Overall Analysis

- The **distance metric** shows whether the Cuboid is approaching or moving away from the target.
- The **dot product** quantifies how well the Cuboid's heading aligns with the goal direction.
- The **cross product** indicates the lateral deviation (left/right drift).
- The **heading error angle** provides a direct measure of angular misalignment, combining the effects of dot and cross.

References: Grok, Chatgpt