

# MODULE 3

VIDEO LINK: <https://vimeo.com/1055709405/ac5fd275c6>

## SCRIPTS

- 1) Create a sphere that moves back and forth on a line by stopping at the end points and moving with maximum velocity on the mid-point using CoppeliaSim APIs.

```
function sysCall_init()
    sim = require('sim')
    sphereHandle = sim.getObject('..')

    -- Get initial position of the sphere in world coordinates
    local initPos = sim.getObjectPosition(sphereHandle, -1)

    -- Define endpoints relative to the initial position:
    local offset = 0.5
    endpoints = {
        {initPos[1] - offset, initPos[2], initPos[3]}, -- left endpoint
        {initPos[1] + offset, initPos[2], initPos[3]}, -- right endpoint
    }
end

function sysCall_thread()
    -- Ruckig motion profile
    local maxVel = {0.5, 0.5, 0.5, 0.1}
    local maxAccel = {0.2, 0.2, 0.2, 0.1}
    local maxJerk = {0.1, 0.1, 0.1, 0.1}

    local i = 1
    while not sim.getSimulationStopping() do
        local targetPos = endpoints[i]
        local params = {
            object = sphereHandle,
            targetPose = {targetPos[1], targetPos[2], targetPos[3], 0,0,0,1},
            maxVel = maxVel,
            maxAccel = maxAccel,
            maxJerk = maxJerk,
            flags = 0,
        }

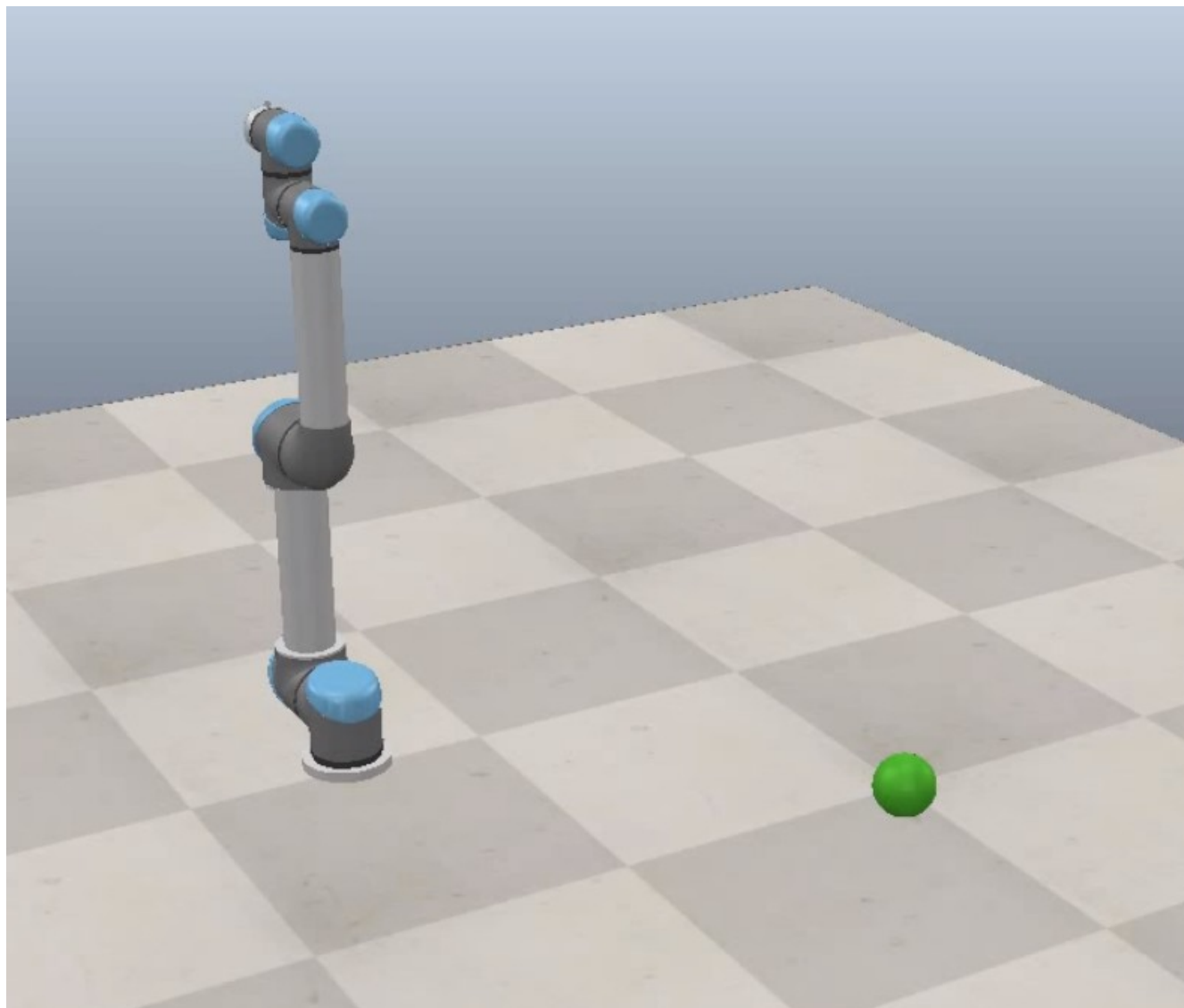
        sim.moveToPose(params)
        sim.wait(1) -- pause at endpoints
        i = (i == 1) and 2 or 1
    end
end
```

- 2) Use the CoppeliaSim model database UR5 and program it to move to three different poses by commanding its joint angles.

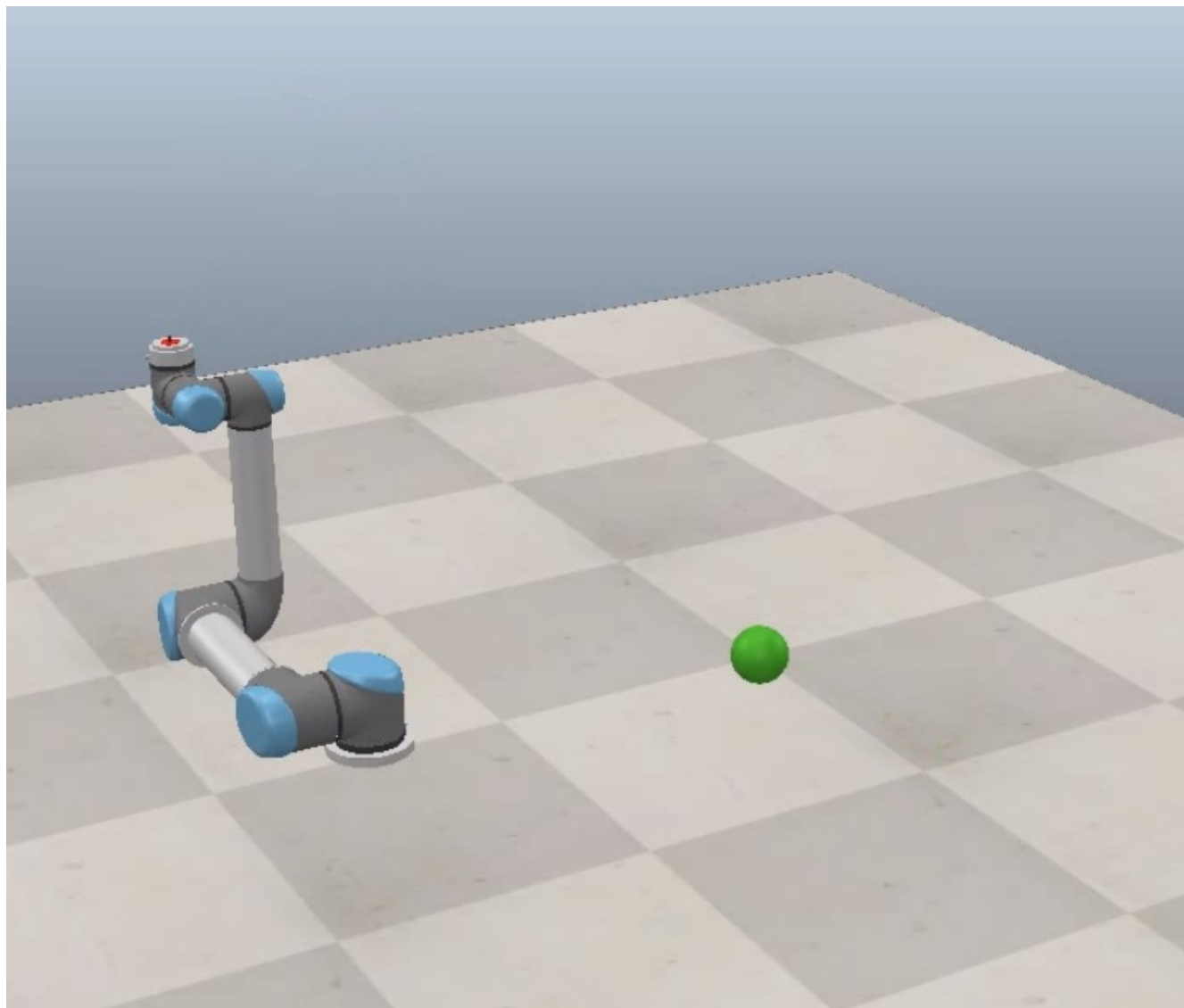
```
sim = require('sim')

function moveToConfig(handles, maxVel, maxAccel, maxJerk, targetConf)
    local params = {
        joints = handles,
        targetPos = targetConf,
        maxVel = maxVel,
        maxAccel = maxAccel,
        maxJerk = maxJerk,
    }
    local poses = {
        -- Pose 1
        {90 * math.pi / 180, 90 * math.pi / 180, -90 * math.pi / 180, 90 * math.pi / 180, 90 *
math.pi / 180, 90 * math.pi / 180},
        -- Pose 2
        {-90 * math.pi / 180, 45 * math.pi / 180, 90 * math.pi / 180, 135 * math.pi / 180, 90 *
math.pi / 180, 90 * math.pi / 180},
        -- Pose 3
        {30 * math.pi / 180, -90 * math.pi / 180, 45 * math.pi / 180, 120 * math.pi / 180, -60 *
math.pi / 180, 0},
        -- Pose 4
        {0, 0, 0, 0, 0, 0}
    }
    while not sim.getSimulationStopping() do
        for i, pose in ipairs(poses) do
            moveToConfig(jointHandles, maxVel, maxAccel, maxJerk, pose)
            sim.wait(2) -- Wait before switching to next pose
        end
    end
end
```

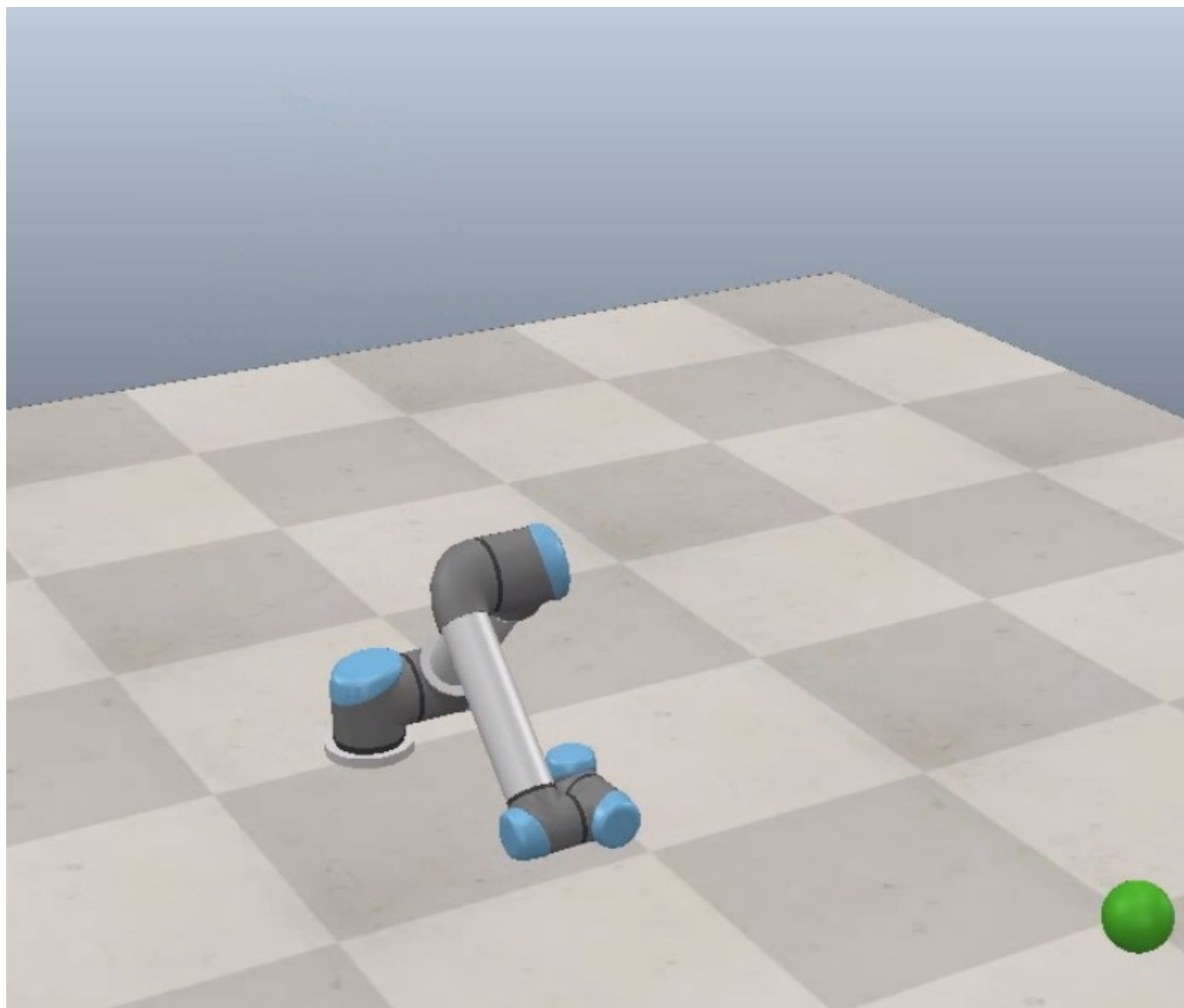
**Starting Position:**



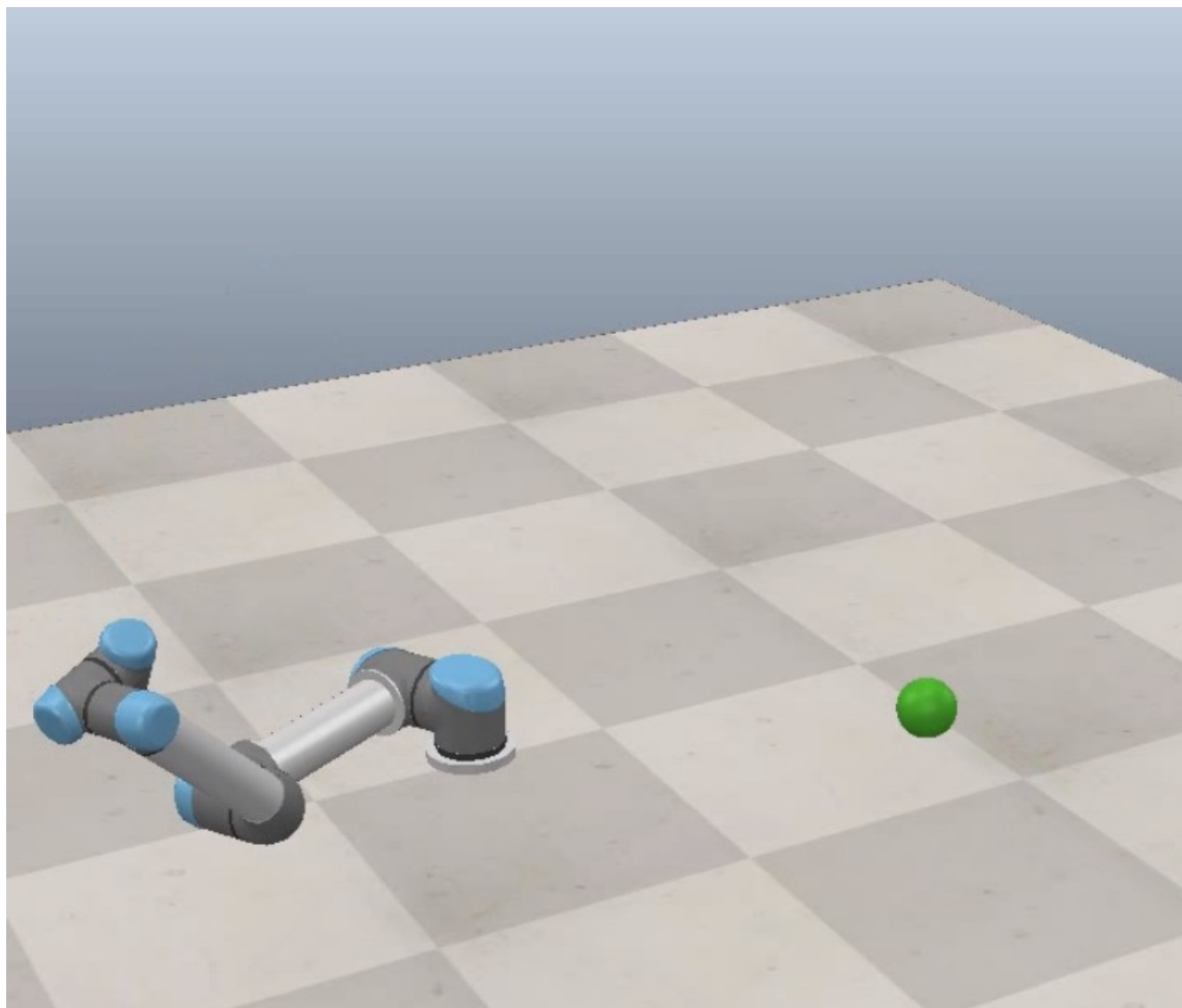
**UR5 at first position while ball moving to first end pos**



**UR5 moves to second pos while ball rapidly passed through it's center point and decelerates to its other endpoint**



**UR5 gets to third pos while sphere is moving back to other endpoint (captured almost 75% of the way there)**



**UR5 resets back to original pos and ball continues to move to other endpoint**

