

---

## Table of Contents

.....	1
Goal: .....	1
Insert code here: record position .....	1
Insert code here: plot trajectory (position) in 3D .....	2

```
% Clear the workspace
clc
clear all
close all
addpath("Api");
```

## Goal:

Collect the drone position and plot it over time. Move the target in the simulator smoothly.

```
% Initialize Communication with CoppeliaSim
[ret_status, sim, clientID] = initializeComm();

% Make sure that initialization is succesful
if (ret_status == 0)

    % Reference the 'Quadricopter' object in CoppeliaSim as 'Quad' in MATLAB
    [returnCode, Quad] = getObjectReference(sim, clientID, 'Quadricopter');
    % Return Quad's position as quad_pos
    [returnCode, quad_pos] = getObjectPosition(sim, clientID, Quad, 1);
    % Reference the 'Quadcopter_target' object in CoppeliaSim as 'target' in
    MATLAB
    [returnCode, target] = getObjectReference(sim,
clientID, 'Quadricopter_target');

    t=clock;
    startTime=t(6);
    currentTime=t(6);
    p_x = [];
    p_y = [];
    p_z = [];

    T = 12; % How long you want to collect the data for
    i = 0; j = 0; k = 1;

    while (currentTime-startTime < T)
        if(sim.simxGetConnectionId(clientID) ~= -1)
```

## Insert code here: record position

```
        % Here is an example line of code on getting the position of the
        target object
```

---

```

0);

    [returnCode, quad_pos] = getObjectPosition(sim, clientID, Quad,

% Add the newly read position to the positions array.
p_x = [p_x; quad_pos(1)];
p_y = [p_y; quad_pos(2)];
p_z = [p_z; quad_pos(3)];

% Move target in a circle
position = [ 0.50*cos(i)-0.50, 0.50*sin(j), k];
% Move x and y
i = i + 0.2;
j = j + 0.2;

% Read current time
t = clock;
currentTime = t(6);
% Tell the target to move
[returnCode] = setObjectPosition(sim, clientID, target, position);
% Make sure to add some delay...
pause(0.15) % This delay will be computer dependent

end
end
positions = [p_x, p_y, p_z];

```

## Insert code here: plot trajectory (position) in 3D

```

figure(1)
plot3(positions(:,1),positions(:,2),positions(:,3),'linewidth',3);
title('3D Quadrotor path from CoppeliaSim')
xlabel('x [m]')
ylabel('y [m]')
zlabel('z [m]')
saveas(gcf,'Part1Circle.png');
figure(2)
plot(positions(:,1),positions(:,2))
title('2D Quadrotor path from CoppeliaSim')
xlabel('x [m]')
ylabel('y [m]')
saveas(gcf,'Part1_2d.png');

% Kill the connection to CopelliaSim
uninitializeComm(sim, clientID)

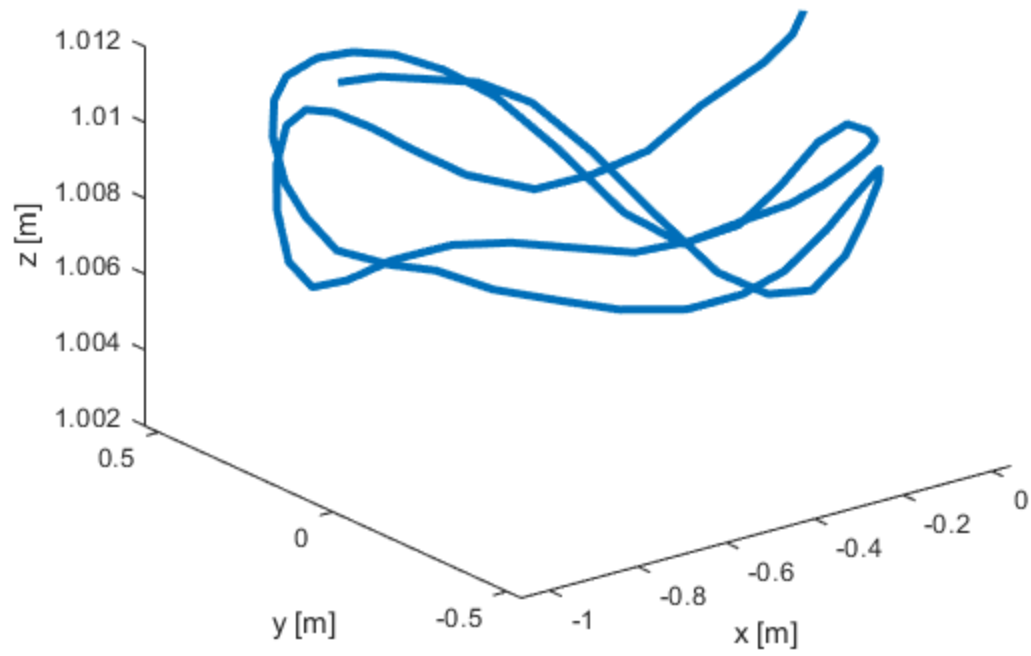
ans =

0

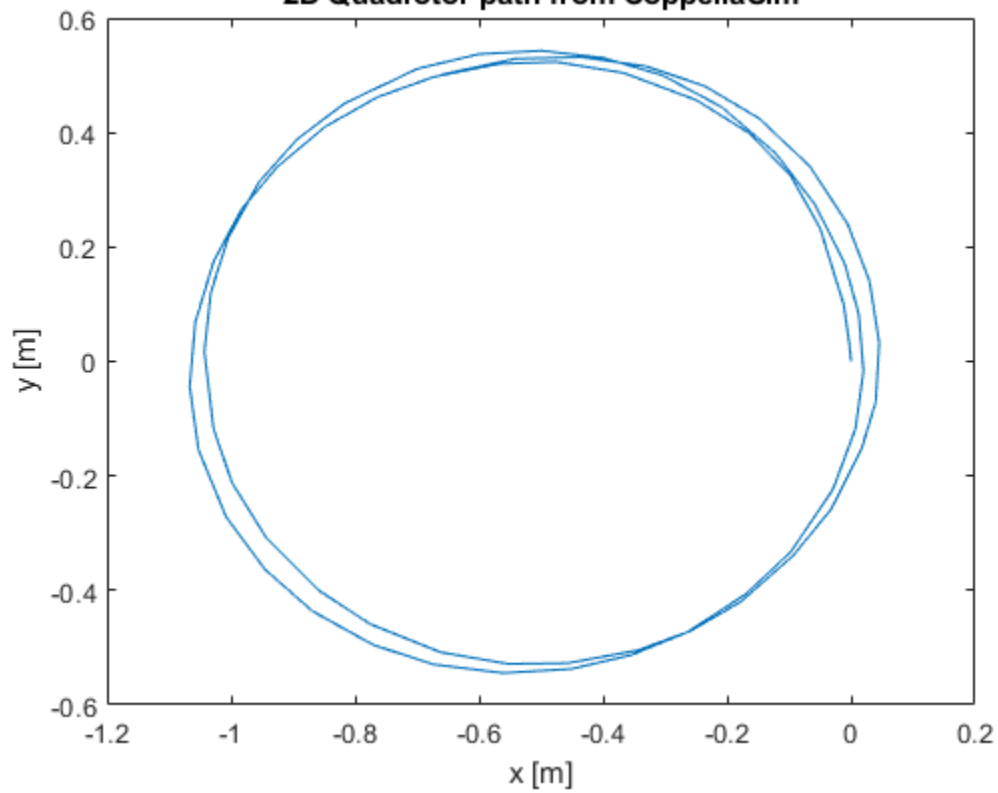
```

---

**3D Quadrotor path from CoppeliaSim**



**2D Quadrotor path from CoppeliaSim**



---

```
else
    disp('Unable to connect to CopelliaSim')
end
```

*Note: always make sure you use the corresponding remoteApi library  
(i.e. 32bit Matlab will not work with 64bit remoteApi, and vice-versa)  
Connected to CopelliaSim*

*Published with MATLAB® R2022b*