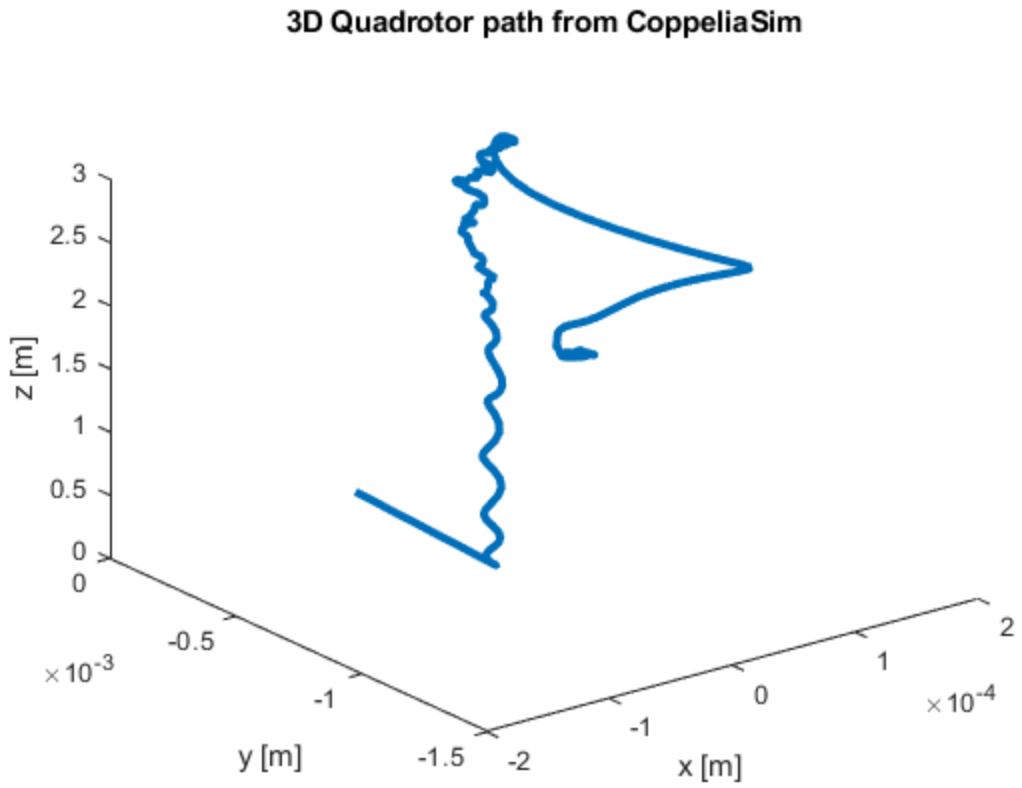
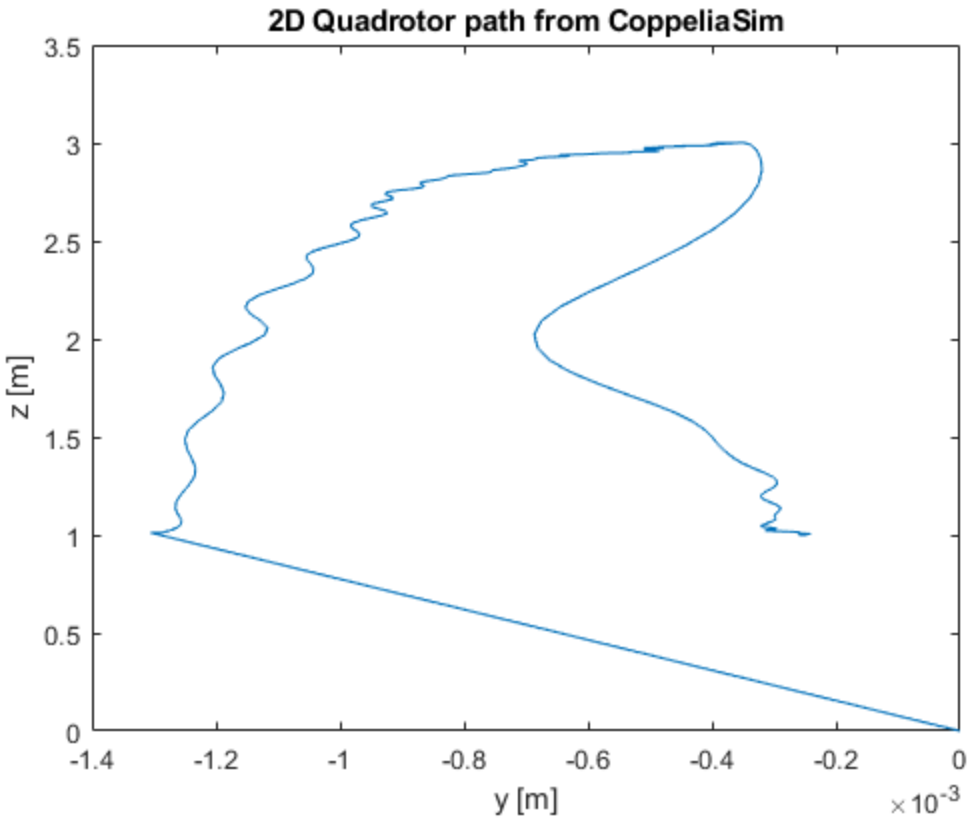
```
% Clear the workspace
clc
clear all
close all
addpath("Api")

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

% Make sure that initialization is succesful
if (ret_status == 0)
    % pull required variables from lect 13 code

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

    % sphere position
    p_x_star = target_position(1);
    p_y_star = target_position(2);
    p_z_star = target_position(3);

    % Current Positions
    p_x = [];
    p_y = [];
    p_z = [];

    % Iterations
    i = 0;
    j = 0;
    k = 1;

    % Set a tolerance bc we can't get to targets without an overshoot
    err = 0.08;

    % Add some time
    % variables for time/ when to stop sim
    T = 14; % How long you want to collect the data for

    % for publish because it doesn't draw the figure with T=14
    %T = 40;

    % variables for time
    t=clock;
```

```
startTime=t(6);
currentTime=t(6);

% While sim is on
 while(currentTime-startTime < T)
    % if the sim is running do this
    if(sim.simxGetConnectionId(clientID) ~= -1)
    % Insert code here:
    % remove the above example, and fly to (0,0,3) and then to (0,0,1)

    % Collect info on position
    [returnCode, quad_pos] = getObjectPosition(sim, clientID, Quad, 0);
    p_x = [p_x; quad_pos(1)];
    p_y = [p_y; quad_pos(2)];
    p_z = [p_z; quad_pos(3)];


    % Tell Target to move, if statements
    p_x_star = 0;
    p_y_star = 0;

    if(p_z<=3)
        targZ =3+err;
        targZinv =1;
    else
        targZ =1-err;
        targZinv =3;
    end

    p_z_star = 1/(i+1)*targZinv + i/(i+1)*targZ;
    position = [p_x_star,p_y_star,p_z_star];
    i = i + 0.001;

    % Read current time
    t = clock;
    currentTime = t(6);

    % Send to target
    [returnCode] = setObjectPosition(sim, clientID, target, position);



    end

 end
% plot
positions = [p_x, p_y, p_z];
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,'Part2_3d.png');
```

```
    figure(2)
    plot(positions(:,2),positions(:,3))
    title('2D Quadrotor path from CoppeliaSim')
    xlabel('y [m]')
    ylabel('z [m]')
    saveas(gcf,'Part2_2d.png');

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

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

ans =

     0
```





Published with MATLAB® R2022b