Network Ninjas Task 2

```
% Quadcopter Simulation Assignment

clear; clc; close all;

% Parameters
m = 1.0; % Mass (kg)
g = 9.81; % Gravity (m/s^2)
I = diag([0.02, 0.02, 0.04]); % Moment of inertia matrix (kg·m^2)
dt = 0.01; % Time step (s)
T = 10; % Total simulation time (s)
n_steps = T / dt; % Number of steps

% Initial States
state = zeros(12, 1); % [x, y, z, vx, vy, vz, roll, pitch, yaw, p, q, r]

% Control inputs: [u1 (thrust), u2 (roll torque), u3 (pitch torque), u4
(yaw torque)]
controls = [m * g; 0; 0; 0]; % Initial hover condition

% State storage for visualization
trajectory = zeros(n_steps, 12);

% Simulation loop
for k = 1:n_steps
    % Extract current state
    roll = state(7); pitch = state(8); yaw = state(9);
    vx = state(4); vy = state(5); vz = state(6);
    p = state(10); q = state(11); r = state(12);


    if k < n_steps / 4
        controls(1) = m * g + 5.0;
    elseif k < n_steps / 2
        controls(2) = 0.01;
    elseif k < 3 * n_steps / 4
        controls(3) = 0.01;
    else
        controls(4) = 0.01;
    end

    % Translational accelerations
    acc = [controls(1)/m * (sin(yaw)*sin(roll) +
cos(yaw)*sin(pitch)*cos(roll)); controls(1)/m * (-cos(yaw)*sin(roll) +
sin(yaw)*sin(pitch)*cos(roll)); controls(1)/m * (cos(pitch)*cos(roll)) -
g];

    % Rotational accelerations
    angular_acc =
[controls(2)/I(1,1);controls(3)/I(2,2);controls(4)/I(3,3)];

    % Update velocities
    state(4:6) = state(4:6) + acc * dt; % Update linear velocity
    state(10:12) = state(10:12) + angular_acc * dt; % Update angular
velocity
```

```
    % Update positions and angles
    state(1:3) = state(1:3) + state(4:6) * dt; % Update position
    state(7:9) = state(7:9) + state(10:12) * dt; % Update orientation

    % Store trajectory
    trajectory(k, :) = state';
end

% Visualization (provided, do not modify)
time = (0:n_steps-1) * dt;
pos = trajectory(:, 1:3); % [x, y, z]
angles = trajectory(:, 7:9); % [roll, pitch, yaw]

% Plot positions
figure;
subplot(3, 1, 1); plot(time, pos(:, 1)); title('X Position'); xlabel('Time
(s)'); ylabel('X (m)');
subplot(3, 1, 2); plot(time, pos(:, 2)); title('Y Position'); xlabel('Time
(s)'); ylabel('Y (m)');
subplot(3, 1, 3); plot(time, pos(:, 3)); title('Z Position'); xlabel('Time
(s)'); ylabel('Z (m)');

% Plot angles
figure;
subplot(3, 1, 1); plot(time, angles(:, 1)); title('Roll'); xlabel('Time
(s)'); ylabel('Roll (rad)');
subplot(3, 1, 2); plot(time, angles(:, 2)); title('Pitch'); xlabel('Time
(s)'); ylabel('Pitch (rad)');
subplot(3, 1, 3); plot(time, angles(:, 3)); title('Yaw'); xlabel('Time
(s)'); ylabel('Yaw (rad)');

% 3D Trajectory
figure;
plot3(pos(:, 1), pos(:, 2), pos(:, 3), 'b', 'LineWidth', 1.5);
grid on;
title('Quadcopter 3D Trajectory');
xlabel('X (m)'); ylabel('Y (m)'); zlabel('Z (m)');
```

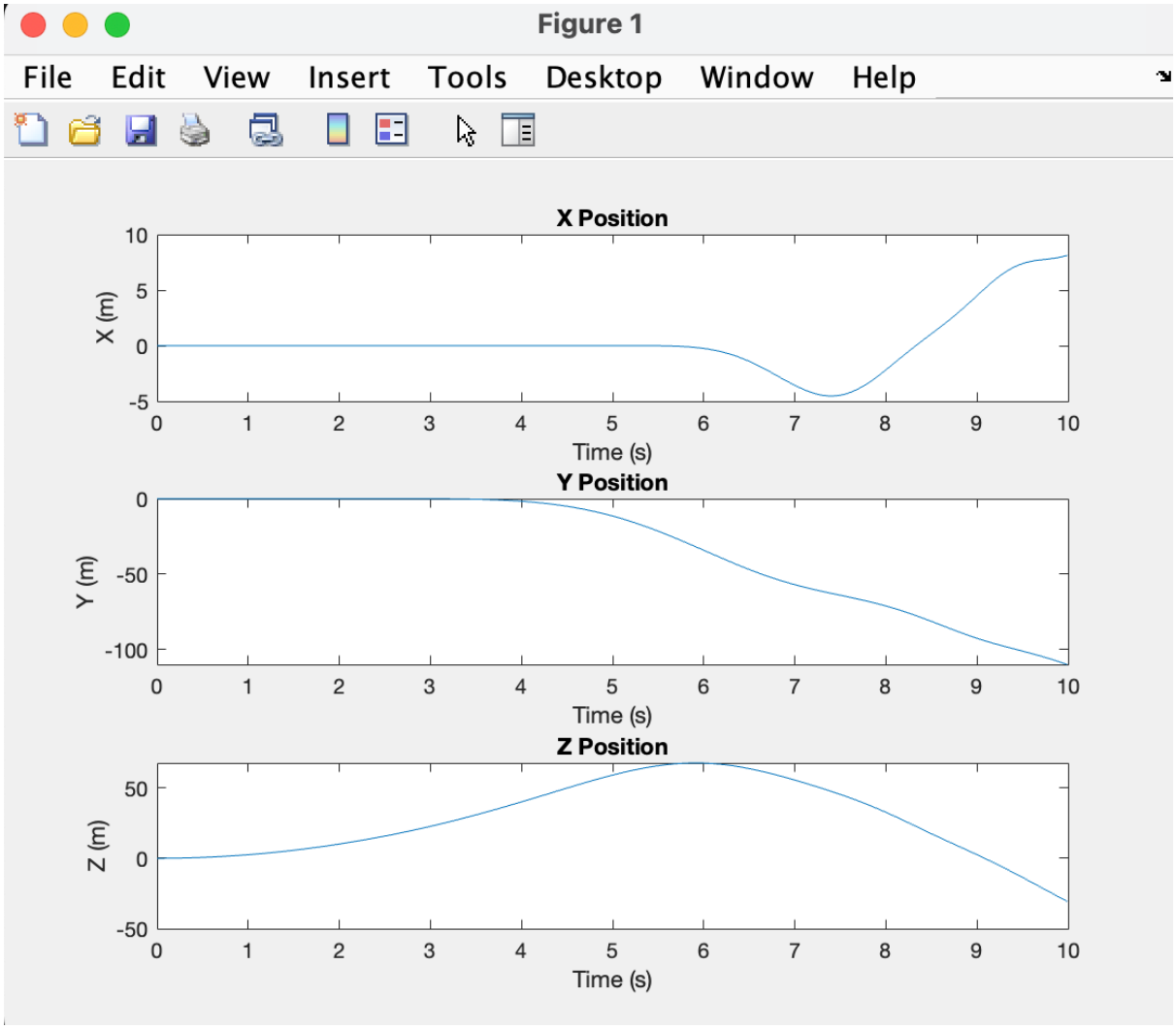
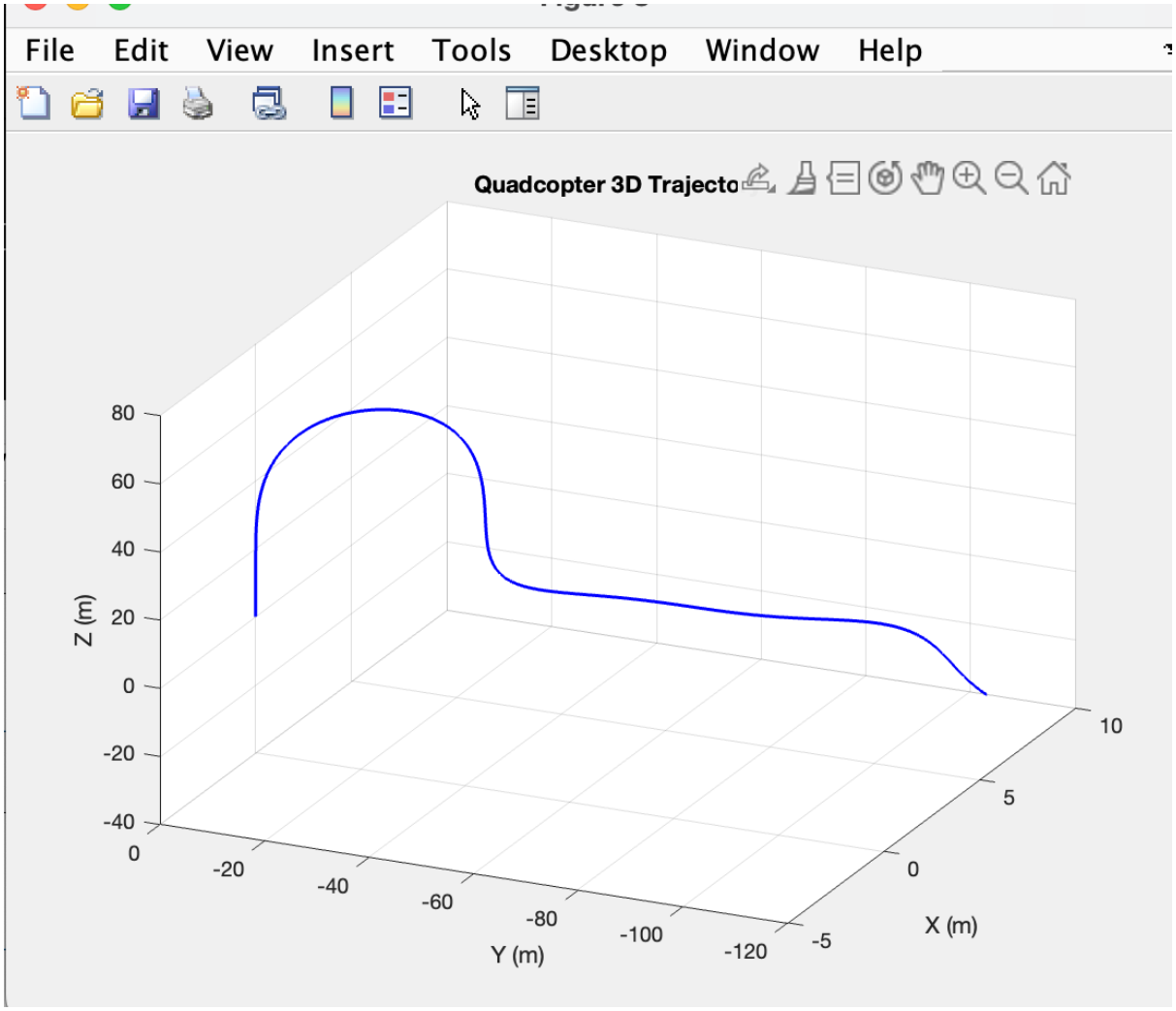


Figure 1
File Edit View Insert Tools Desktop Window Help
X Position
X (m)
10
5
0
-5
0 1 2 3 4 5 6 7 8 9 10
Time (s)
Y Position
Y (m)
0
-50
-100
0 1 2 3 4 5 6 7 8 9 10
Time (s)
Z Position
Z (m)
50
0
-50
0 1 2 3 4 5 6 7 8 9 10
Time (s)


File Edit View Insert Tools Desktop Window Help
Quadcopter 3D Trajecto
80
60
40
20
0
-20
-40
Z (m)
0
-20
-40
-60
-80
-100
-120
Y (m)
-5
0
5
10
X (m)

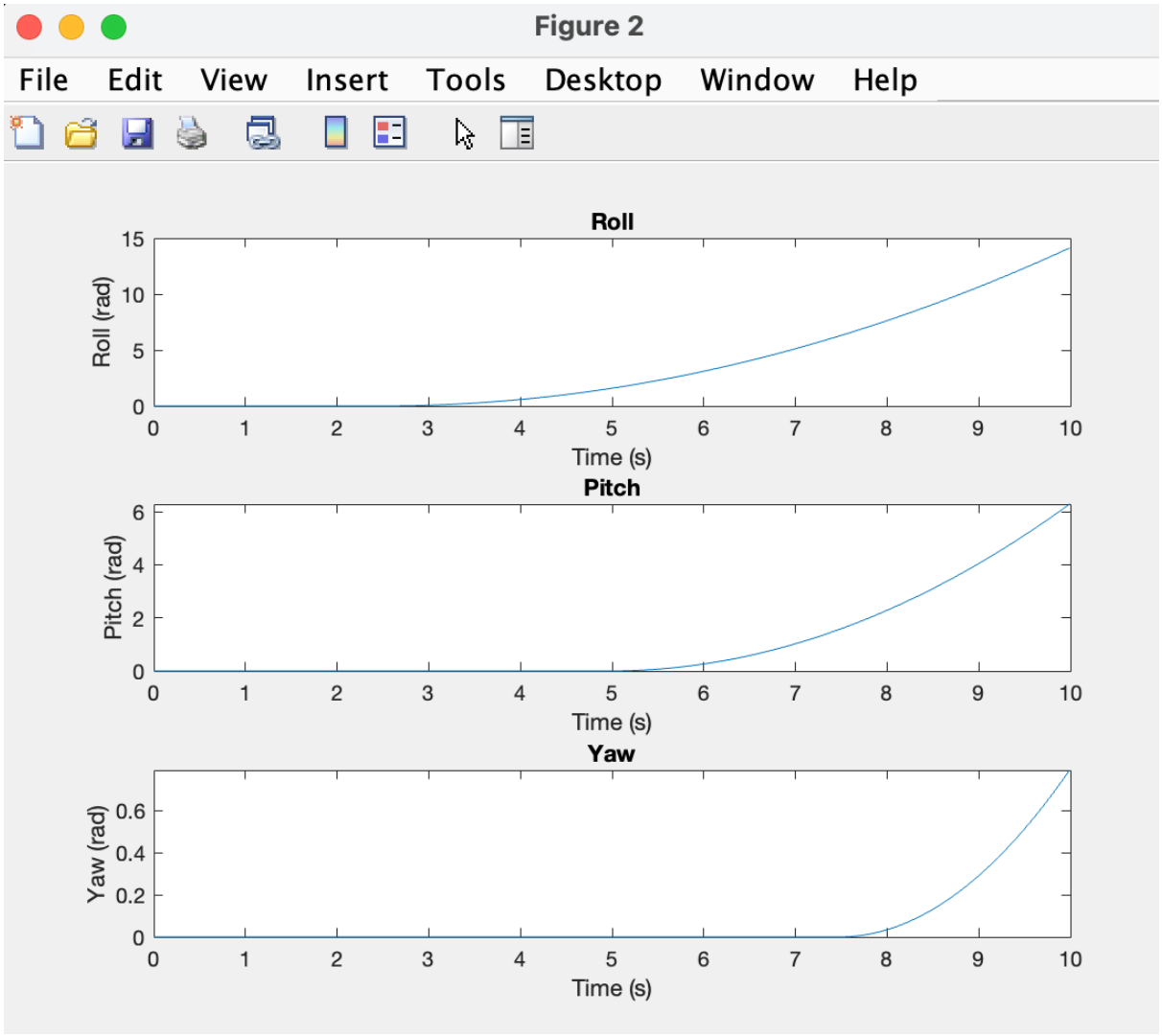


Figure 2
File Edit View Insert Tools Desktop Window Help
Roll
Roll (rad)
Time (s)
Pitch
Pitch (rad)
Time (s)
Yaw
Yaw (rad)
Time (s)