Quadcopter Drone Simulation

Ashish Bhogate, Dheeraj Kumar, Dhrumil Shah, Prasad Borkar

Abstract— **This study investigates quadcopter drone dynamics in 3DOF (rotational) and 6DOF (translational and rotational). Open-loop analysis assesses controllability, observability, and stability. Closed-loop controllers are designed for Z-direction altitude control. Results inform the distinct properties of each system and provide practical insights for enhancing drone control.**

I. Introduction

Significant advancements have been made in small Unmanned Aerial Vehicles (UAVs) over the last several decades. Their growing importance lies in their capacity to replace manned aerial vehicles for various tasks, thereby reducing associated costs. UAVs serve numerous non-military purposes, including climate monitoring, news reporting, and law enforcement. In this example, we explore the flight principles of a QuadCopter, a four-rotor helicopter. It is an underactuated, dynamic vehicle with four input forces (one for each rotor) and six Degrees Of Freedom (6DOF). The Quadcopter's motion in 6DOF is controlled by individually varying the RPM of its four rotors, influencing lift and rotational forces. Tilt, induced by the slow-spinning motor, facilitates roll and pitch, dividing thrust into two directions for linear motion. Rotors rotate in clockwise and anticlockwise pairs, as depicted in Figure [1], controlling yaw due to the drag force on propellers.

In addressing the challenge of varying motor efficiency and the Center of Gravity (CG) being nearly aligned with the rotor plane, our team opted for a PID controller to assess the Quadcopter's performance. To analyze stability, controllability, and observability, the team simulated the closed-loop of the drone using MATLAB, Simulink, and Simscape. This approach ensures a comprehensive examination of the UAV system, emphasizing the significance of controller selection and its impact on the Quadcopter's behavior in response to the identified challenges.

II. Modeling

A. Quadcopter Dynamics

This study delves into the dynamics of drone systems, exploring both translational and rotational aspects.

Forces Equations in Rotational Motion :

The roll motion equation for a drone can be expressed as :

$$\tau_x = \sum_{i=1}^{4} r_i \cdot f_i$$

where:

τ_x : Rolling moment about the x-axis

r_i : Distance of individual force F_i from the axes

f_i : Individual force applied by the ith rotor

$$\tau_x = r_1 \cdot f_1 + r_2 \cdot f_2 + r_3 \cdot f_3 + r_4 \cdot f_4$$

$$r_1 = r_3 = 0$$

$$r_2 = d, r_4 = -d$$

$$\tau_x = d \cdot f_2 - d \cdot f_4$$

The pitch motion equation for a drone can be expressed as:

$$\tau_y = r_1 \cdot f_1 + r_2 \cdot f_2 + r_3 \cdot f_3 + r_4 \cdot f_4$$

$$r_2 = r_4 = 0, r_1 = -r_3 = d$$

$$\tau_y = d \cdot f_1 - d \cdot f_3$$

The Yaw motion equation for a drone can be expressed as τ_z and is the torque about z-axis as shown in figure[3].

$$\tau - z = \sum_{i=1}^{4} Q_i$$

where:

Q_i : Reactive torque due to air drag on rotor blade

$$\tau_z = c(-F_1 + F_2 - F_3 + F_4)$$

Forces Equations in Translational Motion :

The vertical motion of the quad copter is governed by the total thrust force 'T' and weight 'w' along the z-axis figure [10]

$$T = \sum_{i=1}^{4} F_i,$$
$$z'' = \frac{T - mg}{m}$$

The Horizontal motion can be written as:

$$x'' = \frac{T\cos(\theta) - mg}{m}$$

$$x'' = -g\theta$$

$$y'' = -g\phi$$

B. State-Space Representation

We conduct a comparative analysis for 3DOF and 6DOF systems.The former focuses exclusively on rotational dynamics (roll, pitch, yaw), while the latter considers translational dynamics in the x, y, and z directions, along with rotational dynamics.

Below table illustrates critical variables for modeling system dynamics, including drone and propeller mass, thrust, etc. These variables serve as the basis for understanding the system's behavior.

Var.	Value
Drone mass(M)	500 gm
Radius of drone body(R)	5 cm
Rotor mass(m)	1 gm
Force to moment scaling factor(c)	1 (Unitless)
Length between m and $M(d)$	22.5 cm
Gravity Constant(g)	9.8 m/s^2

3 DOF

- Roll angle (Φ)
- Pitch angle (θ)
- Yaw angle (Ψ)
- Roll rate (Φ')
- Pitch rate (θ')
- Yaw rate (Ψ')

6 DOF

- P_x - x-axis(x)
- P_y - y-axis(y)
- P_z - z-axis(z)
- V_x - x-axis(x')
- V_y - x-axis(y')
- V_z - x-axis(z')
- Roll angle (Φ)
- Pitch angle (θ)
- Yaw angle (Ψ)
- Roll rate (Φ')
- Pitch rate (θ')
- Yaw rate (Ψ')

State-space representations, such as :

$$\dot{x} = Ax + Bu$$

$$y = Cx + D$$

establish connections between the system's state, input, and output. The matrices A, B, C, and D are computed using MATLAB.

Below matrices are for 3 DOF system:

A = $\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

B = $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.0037 & 0 & -0.0037 \\ 0.0037 & 0 & -0.0037 & 0 \\ -0.0001 & 0.0001 & -0.0001 & 0.0001 \end{bmatrix}$

C = $\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$

D = $\begin{bmatrix} 0 \end{bmatrix}$

Below matrices are for 6 DOF system:

$$A = \begin{bmatrix} 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&-9.8&0&0&0&0 \\ 0&0&0&0&0&0&9.8&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0.002&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0.0002&0&0 \\ 0&0&0.0002&0 \\ 0&0&0&1.0002 \end{bmatrix}$$

$$C = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \end{bmatrix}$$

Once these values are obtained, we analyze the system's properties, including controllability, observability, and stability for both 3DOF and 6DOF cases.

III. Analysis of properties

A. Controllability

In a fully controllable system, the length of the state vector should match the rank of the controllability matrix. The 3DOF system is controllable, with a rank of 6, aligning with the number of states. However, the 6DOF system, with a state vector length of 12 and a rank of 10, is uncontrollable. Full controllability implies influence over all positions, angles, and their rates of change, while a rank of 10 indicates two aspects beyond control with the current set of controls.

B. Observability

Both systems are observable, as the rank of the observability matrix equals the number of states in each case. In practical terms, this characteristic enhances the feasibility and effectiveness of implementing control strategies, as the system's internal states are effectively discernible from observable outputs

C. Stability

Eigenvalues for both systems are zero, indicating marginal stability. In the context of roll dynamics, a marginally stable system tends to maintain a constant roll angle without returning to the equilibrium position, potentially leading to persistent tilt without growth or decay over time. The drone's symmetrical design ensures even mass and thrust distribution, contributing to stable roll positions. However, this inherent stability necessitates active control measures. Implementing a PID controller in the closed-loop system is crucial for optimal performance and responsiveness, particularly in the presence of external disturbances or perturbations. Continuous monitoring and adjustment enhance the drone's adaptability during flight.

D. Open Loop Response

We plotted the system response using the lsim() function in MATLAB. Due to its physical system properties, the system is naturally unstable and rapidly collapses when subjected to a slight perturbation. The open loop response is shown in Figure [4]. The open-loop system exhibits a linear relationship between attitude and time, suggesting a consistent and constant rate of ascent and descent. This behavior results from the drone's inherent characteristics and applied input in the absence of active control corrections. To stabilize the system, a PID controller is implemented, forming a closed-loop system.

E. Selection of Z-axis

Vertical motion is pivotal for the quadcopter, impacting hovering, lift-off, altitude maintenance, and stability. The state-space representation manages control intricacies, with a focus on vertical dynamics. Isolating the Z-axis streamlines the system, facilitating an in-depth exploration of fundamental dynamics and refining precise control mechanisms. This strategic approach enhances the efficiency of the simulation project, emphasizing the critical role of vertical motion control and the isolation of the Z-axis in achieving success.

IV. SIMULATION RESULTS

A. Matlab: Closed Loop Response

A linear state feedback PID controller was employed to control gains for each state. The gain matrix K was obtained by placing arbitrary unique eigenvalues with negative real parts using the place() function in MATLAB; this is possible due to the controllability of the system. As the K value was raised, the settling time decreased but the overshoot and oscillations significantly rose. The closed-loop responses are shown in *Figure 8* (see Appendices: Figures). With eigenvalues with the noted properties placed in our controlled closed loop response, it is ensured that stability can be achieved although it required some tuning.

B. *System Performance*

Key parameters illuminate the quadcopter's dynamic behavior. A commendable rise time of 1.42 seconds showcases agility in achieving elevations. A transient time of 4.25 seconds underscores swift equilibrium restoration after disturbance. The settling time of 4.43 seconds demonstrates efficiency in maintaining steady-state post-input changes. A modest 5.48 percent overshoot and 1.68 seconds peak time contribute to stability and responsiveness. These characteristics collectively highlight the quadcopter's dynamic behavior, crucial for precision and reliability in applications.

C. *Simulation in Matlab SIMULINK*

In crafting our quadcopter simulation, we tailored insights from the Parrot Minidrone Hover model in Simulink to meet project demands. This involved overhauling the model, fine-tuning parameters through calculated adjustments. The simulation utilized tools such as Simulink, Aerospace Blockset™, Aerospace Toolbox, Control System Toolbox™, Signal Processing Toolbox™, and Simulink 3D Animation for robust modeling, analysis, and visualization.

Distinct subsystems within the model replicated real-world dynamics. The flight command subsystem dictated movements, the flight control system executed commands, and sensor and environment models simulated inputs and conditions. The multicopter model encapsulated physical traits, while the flight visualization model provided graphical representation. Achieving simulation objectives, our team ensured stable hovering along the Z-axis, validating theoretical calculations and offering crucial insights for future physical prototype development and optimization.

V. CONCLUSIONS AND DISCUSSIONS

In concluding our quadcopter simulation project, major design decisions focused on precise vertical movement tracking along the Z-axis using Matlab Simulink. This approach facilitated the creation of a detailed model, highlighting the importance of complex control mechanisms for flight stability. Throughout the project, we recognized the pivotal role of sophisticated control mechanisms in achieving quadcopter flight stability. Delving into Matlab Simulink deepened our understanding of interactions between torques, rotor forces, and resulting motion. Effective communication and collaborative problem-solving were essential for error resolution.

VI. APPENDICES

A. *Contribution of Team Members*

1) Ashish Bhogate: Quadcopter dynamics calculations, System modeling and Simulation, Literature review, Report, Presentation
2) Dheeraj Kumar: Controller Design and Tuning, Report, Presentation
3) Dhrumil Shah: Free body diagram, Controller Design , Report, Presentation
4) Prasad borkar: Quadcopter dynamics calculations, Literature review, Report, Presentation

B. *Figures*

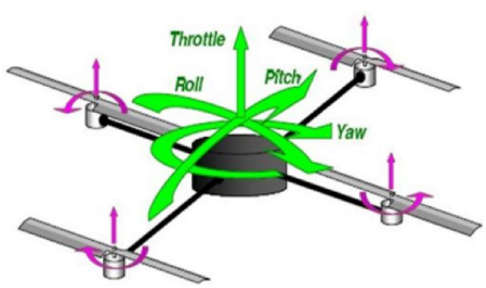

Fig. 1. Quadcopter Schematic

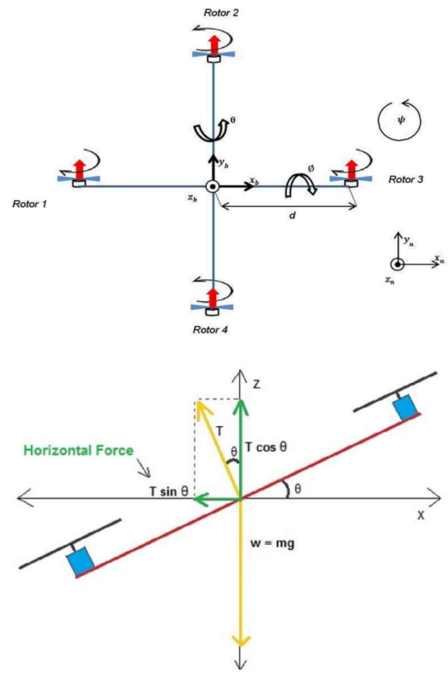

Fig. 2. Quadcopter Dynamics represenation and Free Body Diagram

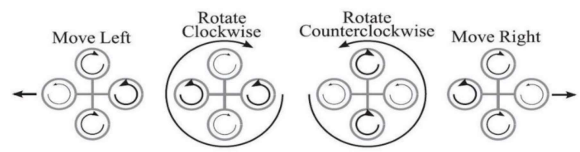

Fig. 3. Yaw and linear motion due to rotor forces

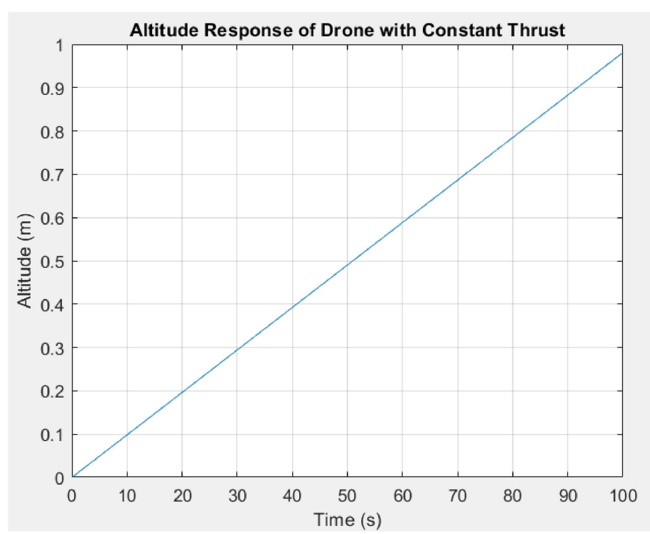

Fig. 4. Altitude Response of Drone with Constant Thrust

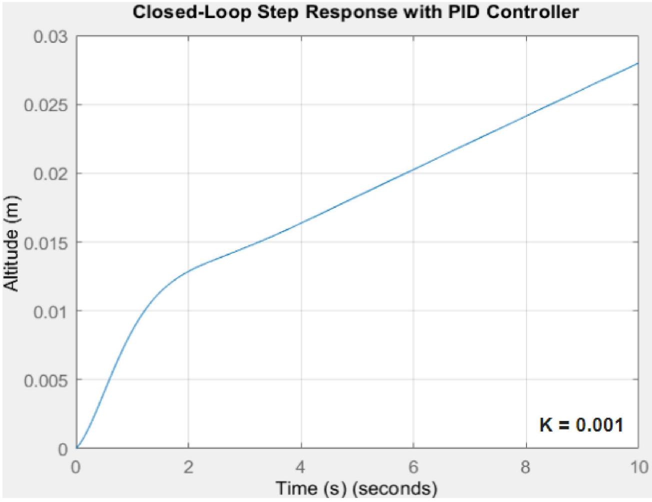

Fig. 5. Closed loop response at K = 0.001

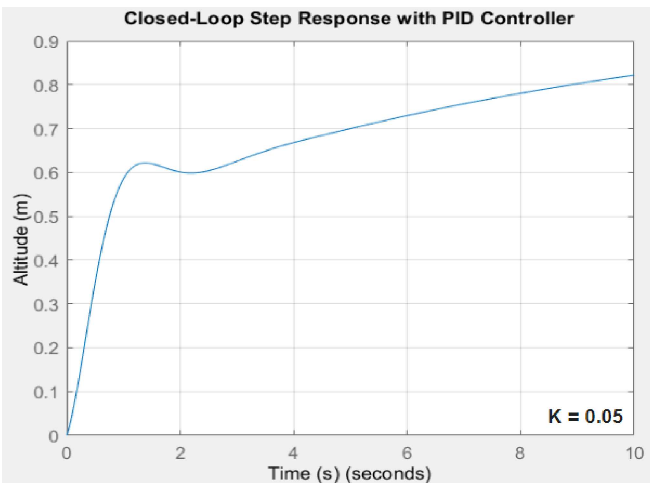

Fig. 6. Closed loop response at K = 0.05

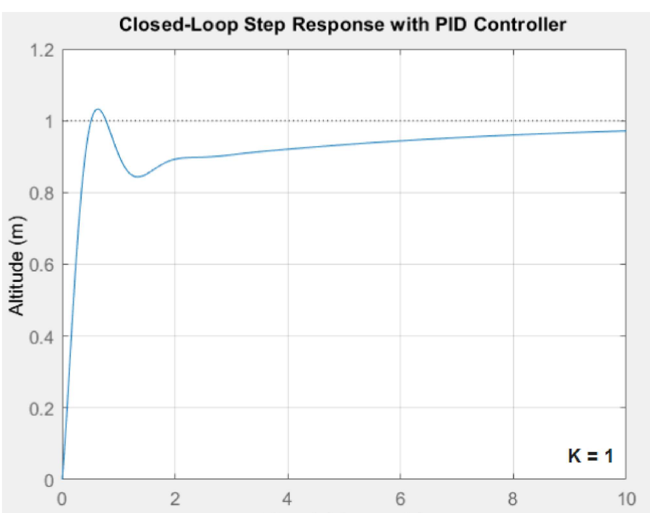

Fig. 7. Closed loop response at K = 1

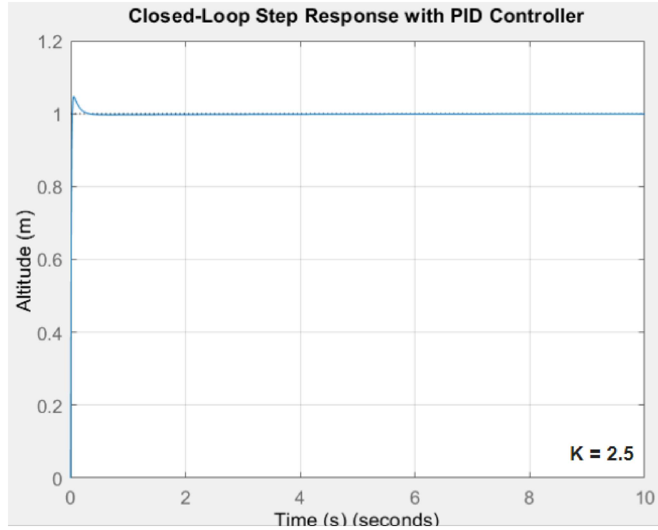

Fig. 8. Closed loop response at K = 2.5

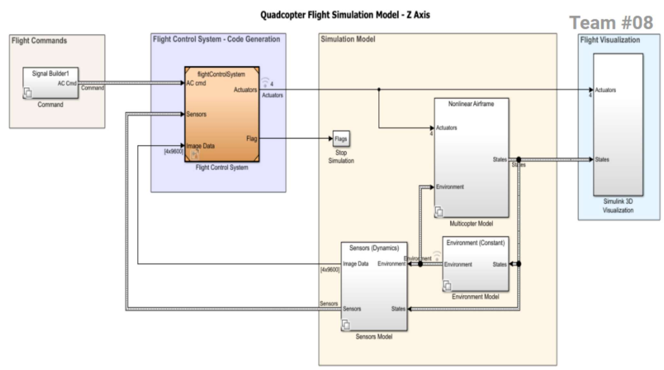

Fig. 9. Quadcopter Flight Simulation Model based on Parrot-minidrone hover

Fig. 10. Minidrone Flight Visualization

C. Codes

```
% Code for 3 Degree of Freedom
% System parameters
M = 500;      % gm
m = 1;        % gm
R = 5;        % cm
d = 22.5;     % cm
g = 9.8;        % m/s^2
c1 = 1;
Ixx = ((2*M*R^2)/5) + 2*m*d^2;  % gm*cm^2
Iyy = ((2*M*R^2)/5) + 2*m*d^2;  % gm*cm^2
Izz = ((2*M*R^2)/5) + 4*m*d^2;  % gm*cm^2;

A = [0 0 0 1 0 0; 0 0 0 0 1 0; 0 0 0 0 0 1; 0 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0];
B = [0 0 0 0; 0 0 0 0; 0 0 0 0; 0 d/Ixx 0 -d/Ixx; d/Iyy 0 -d/Iyy 0; -c1/Izz c1/Izz -c1/Izz c1/Izz];
C = [1 0 0 0 0 0; 0 1 0 0 0 0; 0 0 1 0 0 0];
D = 0;

% Checking Stability
Lambda = eig(A)
% Checking controllability
P = ctrb(A, B)
rank(P)
if rank(P) == size(A, 1)
    disp('System is controllable!');
else
    disp('System is not controllable.');
end

% Checking observability
Q = obsv(A, C)
rank(Q)
if rank(Q) == size(A, 1)
    disp('System is observable!');
else
    disp('System is not observable.');
end
```

```
% Code for 6 Degree of Freedom
% System parameters
M = 500;     % gm
m = 1;       % gm
R = 5;       % cm
d = 22.5;    % cm
g = 9.8;     % m/s^2
c1 = 1;
Ixx = ((2*M*R^2)/5) + 2*m*d^2;  % gm*cm^2
Iyy = ((2*M*R^2)/5) + 2*m*d^2;  % gm*cm^2
Izz = ((2*M*R^2)/5) + 4*m*d^2;  % gm*cm^2;

A = [0 0 0 1 0 0 0 0 0 0 0 0; 0 0 0 0 1 0 0 0 0 0 0 0; 0 0 0 0 0 1 0 0 0 0 0 0; 0 0 0 0 0 0 0 -g 0 0 0 0; 0 0 0 0 0 0 g 0 0 0 0 0; 0 0 0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0 0 1 0 0; 0 0 0 0 0 0 0 0 0 0 1 0; 0 0 0 0 0 0 0 0 0 0 0 1; 0 0 0 0 0 0 0 0 0 0 0 0; 0 0
```

```
    0 0 0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0
    0 0 0 0];
B = [0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0
    0; 1/M 0 0 0; 0 0 0 0; 0 0 0 0; 0 0
    0 0; 0 0 0 0; 0 1/Ixx 0 0; 0 0 1/Iyy
    0; 0 0 0 1/Izz];
C = [1 0 0 0 0 0 0 0 0 0 0 0; 0 1 0 0 0
    0 0 0 0 0 0 0; 0 0 1 0 0 0 0 0 0 0 0
    0; 0 0 0 0 0 0 1 0 0 0 0 0; 0 0 0 0
    0 0 0 1 0 0 0 0; 0 0 0 0 0 0 0 0 1 0
    0 0];
D = 0;

% Checking Stability
lambda = eig(A)
% Checking controllability
P = ctrb(A, B)
rank(P)
if rank(P) == size(A, 1)
    disp('System is controllable!');
else
    disp('System is not controllable.');
end

% Checking observability
Q = obsv(A, C)
rank(Q)
if rank(Q) == size(A, 1)
    disp('System is observable!');
else
    disp('System is not observable.');
end
```

References

[1] Zaid Tahir*, Mohsin Jamil, Saad Ali Liaqat, Lubva Mubarak, Waleed Tahir, Syed Omer Gilani. *State Space System Modeling of a Quad Copter UAV. Indian Journal of Science and Technology, 2015.*

[2] Shaun Sawyer. *Gain-Scheduled Control of a Quadcopter UAV. UWSpace. http://hdl.handle.net/10012/948, 2015.*

[3] Nicholas Andrew Johnson. *Control of a folding quadrotor with a slung load using input shaping. School of Mechanical Engineering, Georgia Institute of Technology, 2017.*

[4] Aniket Ravindra Shirsat. *Modeling And Control of a Quadrotor Aircraft UAV. Arizona State University, 2015.*