Numerical Simulation of Quadrotor Attitude Dynamics Using Runge-Kutta Methods

Riyad Babayev*, Gregor P. McKenzie†, Hamza Rimawi‡, and Bradley Wassef§
University of Illinois Urbana-Champaign, Urbana, IL, 61801

This project models and simulates the nonlinear attitude dynamics of a quadrotor to explore stability, control authority, parameter sensitivity, and disturbance rejection. Using a fourth-order Runge-Kutta method, we integrate Euler's equations of motion and perform targeted simulations to address each question. Results highlight the importance of torque magnitude and inertia in shaping system response and confirm that external disturbances lead to persistent changes without active control.

I. Nomenclature

$\omega_x, \omega_y, \omega_z$	=	rotation rates of the aircraft in the body frame
J_x, J_y, J_z	=	moments of inertia of the aircraft about the x,y, and z axes
τ_x, τ_y, τ_z	=	applied control torques about the x,y, and z axes
t	=	time (s)
y	=	state variable (e.g., angular velocity vector)
h	=	time step size
$f(y,t)$	=	right-hand side of ODE
k_1, k_2, k_3, k_4	=	RK4 intermediate slope estimates
y_k	=	state at time t_k
y_{k+1}	=	state after RK4 update
$O(h^n)$	=	Big-O error notation

II. Dynamical System

We chose to model the dynamic system of the attitude of a quadrotor aircraft. The rotational equations of motion take the form

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \frac{1}{J_x}(\tau_x + (J_y - J_z)\omega_y\omega_z) \\ \frac{1}{J_y}(\tau_y + (J_z - J_x)\omega_z\omega_x) \\ \frac{1}{J_z}(\tau_z + (J_x - J_y)\omega_x\omega_y) \end{bmatrix}. \tag{1}$$

These ordinary differential equations are nonlinear and thus cannot be solved analytically except in very specific cases. A quadrotor is able to control its attitude simply by adjusting the torque (and thus rotation rate) of specific rotors, though the equations here have been simplified such that the torques are combined into generic τ_x, τ_y, and τ_z that represent the effective torque over the whole body.

Specific numeric parameters were chosen to enable the study of this system. Since a quadcopter is likely planar and symmetric, the moments of inertia were selected as

$$J_x = J_y = 0.002 \text{ kg m}^2 \text{ and } J_z = 0.004 \text{ kg m}^2. \tag{2}$$

Additionally, the control torque was limited to ± 0.1 Nm and the rotation rates, ω_x, ω_y, and ω_z, were generally limited to the range ± 30 rad/s with some exceptions where it was relevant to show effects with high angular velocity.

*Undergraduate Student, Department of Aerospace Engineering
†Undergraduate Student, Departments of Aerospace Engineering and Astronomy
‡Undergraduate Student, Department of Aerospace Engineering
§Undergraduate Student, Department of Aerospace Engineering

This system is important because quadrotors are widely used across different fields and have applications beyond aerospace engineering. Because of their versatility, quadrotors of different varieties have found uses everywhere from military and surveillance applications to farming and filmmaking. This wide range of implementations makes the system of quadrotor attitude dynamics a critical one to study and understand.

In this project, we looked to study four specific aspects of the quadrotor attitude dynamic system. Firstly, it is important to know when a quadrotor's attitude is stable. Second, since the attitude state is controlled by the rotor torque, we must know how control torque affects the dynamics of the body. Third, the specific design of a quadrotor affects its moment of inertia, so it is important to know the degree of sensitivity the system has to its moment of inertia. And lastly, in a real world environment, a quadrotor would be subject to perturbing forces, such as wind, so we must know how the attitude dynamics respond to disturbances.

III. Numerical Method

To simulate the attitude dynamics of a quadrotor, the fourth-order Runge-Kutta (RK4) method was chosen. RK4 offers a needed trade-off between accuracy, stability, and computational efficiency. This is crucial for studying systems governed by nonlinear ordinary differential equations like the ones that govern a quadrotor.

The quadrotor model does not show stiffness under standard operating conditions. This makes RK4 a valid approach to modeling the system. Unlike lower-order methods, RK4 offers a substantially higher accuracy per step without needing to evaluate derivatives beyond the first. This allows for the reliable simulation of quadrotor attitude behavior with varied control inputs and disturbances while maintaining computational efficiency.

We consider a general initial value problem:

$$\frac{dy}{dt} = f(y, t), \quad y(t_0) = y_0, \tag{3}$$

where f is smooth but potentially nonlinear. The exact solution can be expanded as a Taylor series:

$$y(t+h) = y(t) + hf + \frac{h^2}{2} f_t + \frac{h^3}{6} f_{tt} + \frac{h^4}{24} f_{ttt} + O(h^5). \tag{4}$$

Evaluating these higher derivatives is not practical nor computationally efficient for complex nonlinear systems. RK4 avoids this by approximating the solution through a weighted average of four slope estimates:

$$k_1 = f(y_k, t_k), \tag{5}$$

$$k_2 = f\left(y_k + \frac{h}{2} k_1, t_k + \frac{h}{2}\right), \tag{6}$$

$$k_3 = f\left(y_k + \frac{h}{2} k_2, t_k + \frac{h}{2}\right), \tag{7}$$

$$k_4 = f\left(y_k + h k_3, t_k + h\right), \tag{8}$$

$$y_{k+1} = y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4). \tag{9}$$

These intermediate slopes correspond to evaluations at the beginning, midpoint, and end of the time step. The weighted average cancels lower-order error terms. Because of this, RK4 has a local truncation error of $O(h^5)$ and a global error of $O(h^4)$. It is suitable for moderately stiff or non-stiff problems, provided h is chosen appropriately. In this case, the appropriate h is 0.0125.

Algorithm Summary: To advance the solution from t_k to t_{k+1}:

1) Evaluate $k_1 = f(y_k, t_k)$.
2) Evaluate $k_2 = f(y_k + \frac{h}{2} k_1, t_k + \frac{h}{2})$.
3) Evaluate $k_3 = f(y_k + \frac{h}{2} k_2, t_k + \frac{h}{2})$.
4) Evaluate $k_4 = f(y_k + h k_3, t_k + h)$.
5) Update $y_{k+1} = y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$.

IV. Implementation

The simulation of the quadrotor's rotational dynamics was carried out in Python. Euler's equations of motion, which describe the nonlinear rotational behavior of a rigid body, were implemented to model the system's attitude dynamics. These equations were expressed in terms of angular velocity components and control torques about the body-fixed axes and were solved numerically using the classical fourth-order Runge-Kutta (RK4) method.

A function was defined to compute the time derivatives of the angular velocity vector, $\dot{\boldsymbol{\omega}}$, based on the current state $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$, the applied control torques $\boldsymbol{\tau} = [\tau_x, \tau_y, \tau_z]$, and the principal moments of inertia $\mathbf{J} = [J_x, J_y, J_z]$. The nonlinear coupling terms inherent in Euler's equations, such as $(J_y - J_z)\omega_y\omega_z$, were incorporated directly to capture the full dynamics.

To integrate the equations of motion, a fourth-order Runge-Kutta algorithm was implemented as a dedicated function. This integrator advanced the state vector $\boldsymbol{\omega}$ over discrete time steps by computing a weighted average of four intermediate slope estimates. The RK4 method was selected for its balance between computational efficiency and accuracy, given the non-stiff nature of the system.

A simulation framework was constructed to evolve the system state over time. This framework initialized the angular velocity vector with user-specified conditions and iteratively applied the RK4 method across a defined time horizon. The solver was designed to allow flexible inputs for torques and inertial parameters, enabling batch simulations under varying physical and control configurations.

To validate the numerical implementation, a convergence study was conducted by varying the time step size h and computing the L_2 norm of the error in angular velocity at a fixed final time. Figure 1 demonstrates that the RK4 method exhibits fourth-order accuracy, as expected. The slope of the error curve closely matches the reference line corresponding to $O(h^4)$, confirming both the correctness and efficiency of the integrator.

The modular structure of the codebase allowed for rapid prototyping and experimentation. All simulations were performed with a constant time step, and numerical accuracy was verified through both convergence analysis and qualitative behavior.

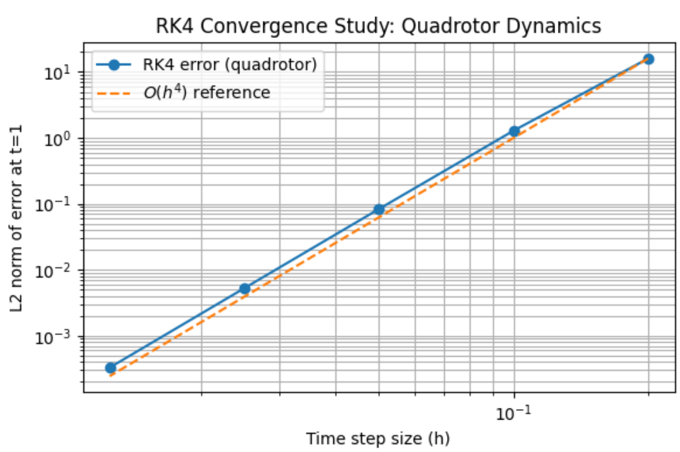

Fig. 1 RK4 convergence study showing L_2 error at $t = 1$ versus step size h.

V. Results and Discussion

The simulation results provide insight into the rotational behavior of the quadrotor under time-varying torque inputs. The following figures present the evolution of key dynamic quantities: angular velocity, control torque, angular acceleration, and rotational kinetic energy. Each plot highlights the coupled and nonlinear dynamics that emerge from Euler's equations.

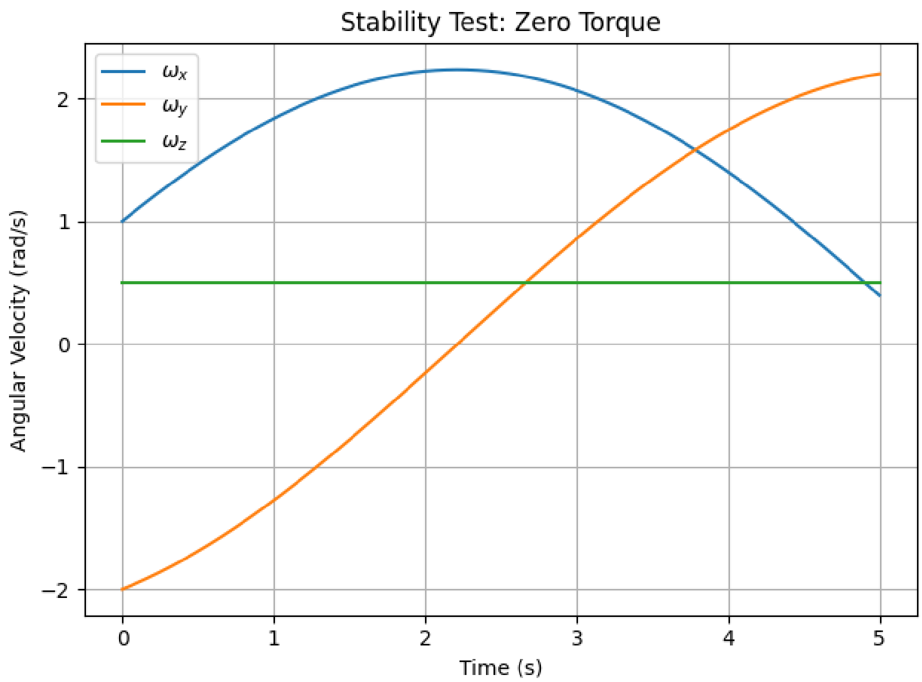

Fig. 2 Time evolution of angular velocity components ω_x, ω_y, and ω_z.

Figure 2 displays the components of the angular velocity over time. Initially, only ω_z is nonzero, while ω_x and ω_y begin at zero. As the simulation progresses, the nonlinear coupling terms in Euler's equations, such as $(J_y - J_z)\omega_y\omega_z$, cause rotational energy to transfer between the axes. This results in all components becoming oscillatory, demonstrating the interconnected nature of rotational dynamics in a rigid body system. The oscillations are smooth and continuous, indicating stable integration with the RK4 method.

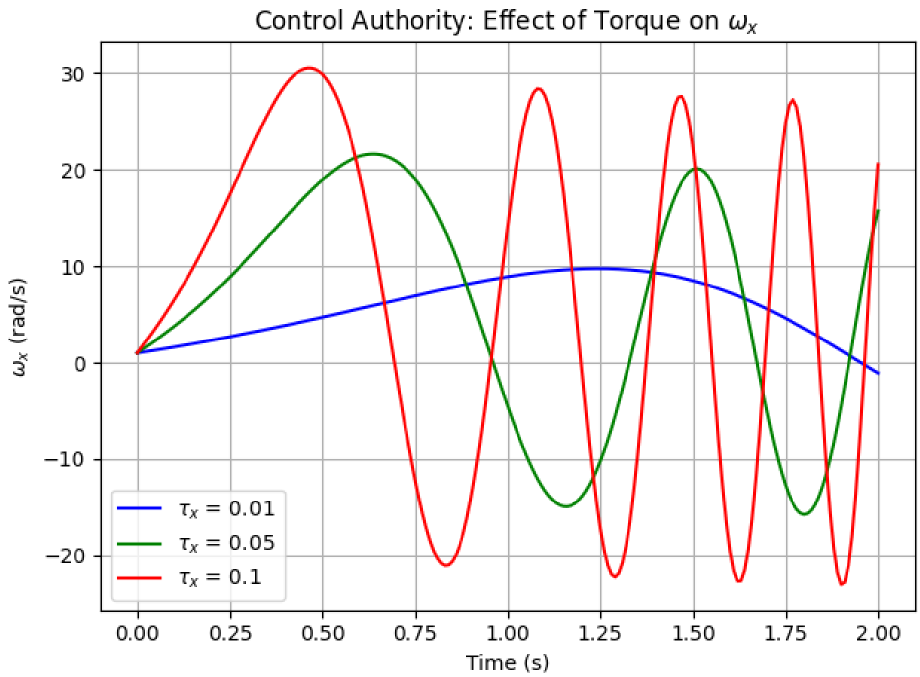

Fig. 3 Applied control torques τ_x, τ_y, and τ_z as a function of time.

Figure 3 shows the input torques applied about each body-fixed axis. These torques are sinusoidal and phase-shifted, introducing periodic excitation to the system. The presence of harmonically varying inputs helps drive the rotational motion and test the full nonlinear coupling in the equations of motion.

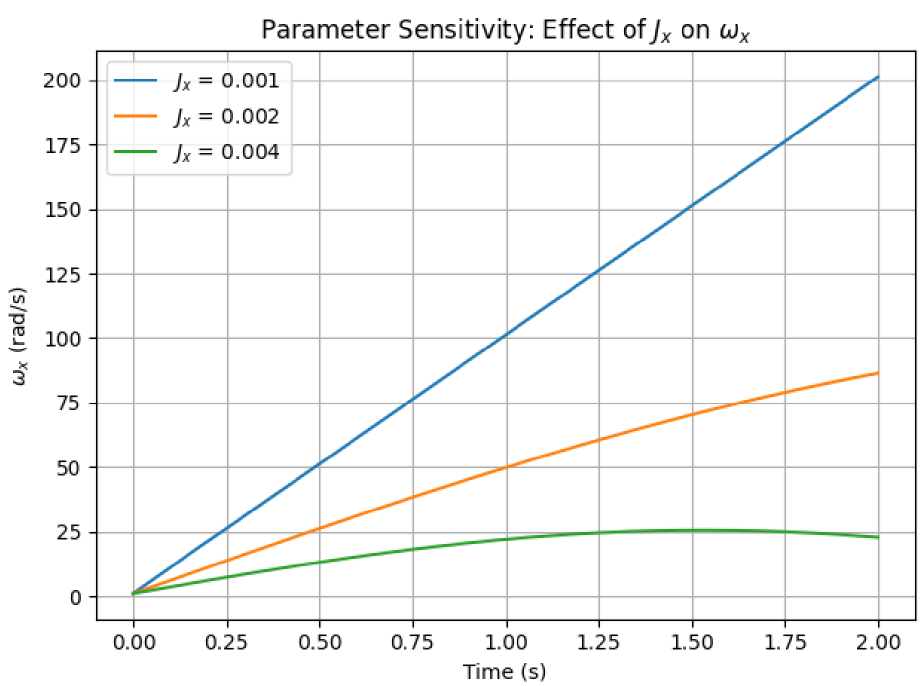

Fig. 4 Angular acceleration components $\dot{\omega}_x$, $\dot{\omega}_y$, and $\dot{\omega}_z$ versus time.

Figure 4 presents the components of the angular acceleration, which are the time derivatives of the angular velocity. These quantities are directly influenced by both the applied torques and the coupling effects among the angular velocity components. The resulting accelerations are non-trivial, reflecting the system's sensitivity to even small disturbances due to its non-linear nature.

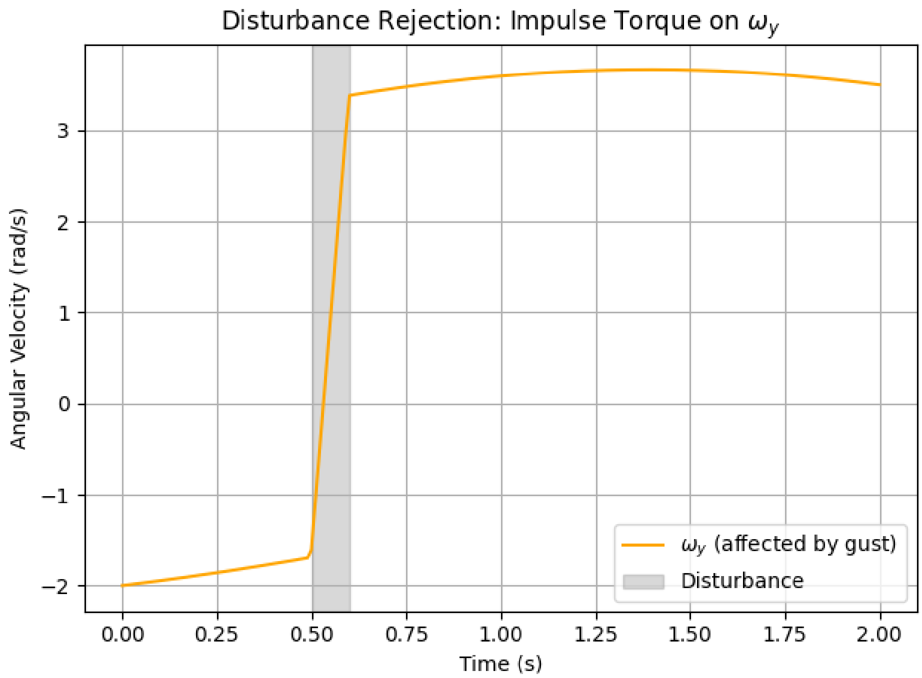

Fig. 5 Rotational kinetic energy of the quadrotor over time.

Finally, Figure 5 shows the total rotational kinetic energy of the system. The energy oscillates as torques are applied and withdrawn, indicating the periodic exchange of rotational energy between different axes. The bounded nature of the energy curve confirms that the system remains stable and energy-conserving in the absence of dissipative forces. This further validates the accuracy of the numerical integrator and correctness of the dynamic model.

In general, the simulation captures the essential features of rigid body dynamics under time-varying torques. The results are consistent with theoretical expectations and highlight the importance of nonlinear coupling in rotational motion.

Contributions of Individual Group Members

Riyad Babayev authored Sections IV and V of the technical report and contributed to result analysis and interpretation. His work equates to 25% of the project.

Gregor McKenzie authored Sections I and II, including the definition of the dynamical system and the justification for the chosen numerical method. His work equates to 25% of the project.

Hamza Rimawi wrote the code used for the simulations presented in Sections IV and V. He focused on generating plots, validating outputs, and formatting visualizations. His work equates to 25% of the project.

Bradley Wassef wrote the code for Sections II and III, including the RK4 implementation and system modeling functions. He also authored Section III of the report. His work equates to 25% of the project.

Acknowledgements

Riyad Babayev, Gregor McKenzie, Hamza Rimawi, and Bradley Wassef give their thanks to ChatGPT, which was instrumental in efficiently writing the code used for this project. ChatGPT was used to assist with code generation and debugging. All content was reviewed, finalized, and led by the team.

Respository Link

The link to the GitHub repository containing all components for this study, including the code, can be found here: `https://github.com/bwassef2/AE370`