A Numerical Simulation and Analysis of UAV Pitch Dynamics

Sufyan Syed, Ben Litvak, Angelina Yan, Samik Singh, Suraj Duvvapu

April 11, 2025

Abstract

This report presents a detailed numerical study of UAV pitch dynamics employing a fourth-order Runge-Kutta (RK4) integration scheme. A comprehensive dynamical model is developed based on a second-order differential equation that captures the essential characteristics of the UAV's pitch motion. Key physical parameters—including the moment of inertia (I_{yy}), aerodynamic stiffness (k), and damping (c)—are systematically varied to investigate their impact on the system's transient response. Furthermore, the study compares different control strategies, specifically a step input and a proportional-derivative (PD) controller, to evaluate the resultant differences in stability and performance. The analysis is supported by time-series plots and phase portraits, from which quantitative metrics such as overshoot and settling time are derived and discussed. These results provide critical insights into the dynamic behavior of UAV pitch systems and establish a basis for optimizing control performance in advanced aerospace applications.

Nomenclature

I_{yy}	Moment of inertia about the pitch axis (kg·m^2).
k	Aerodynamic stiffness (N·m/rad).
c	Damping coefficient (N·m·s/rad).
$\theta(t)$	Pitch angle at time t (rad).
$q(t)$	Pitch rate at time t (rad/s).
$u(t)$	External control moment applied to the system (N·m).
K_p	Proportional gain of the controller.
K_d	Derivative gain of the controller.
θ_{ref}	Desired reference pitch angle (rad).
Δt	Time step for numerical integration (s).
T	Total simulation time (s).
ζ	Damping ratio, defined as $\zeta = \frac{c}{2\sqrt{I_{yy}\,k}}$.
OS	Overshoot (typically expressed as a percentage or a decimal fraction).
$\mathbf{x}(t)$	State vector at time t, where $\mathbf{x}(t) = \begin{bmatrix}\theta(t)\\ q(t)\end{bmatrix}$.
$\mathbf{k}_1,\ \mathbf{k}_2,\ \mathbf{k}_3,\ \mathbf{k}_4$	Intermediate increments computed in the RK4 integration scheme.

1 Introduction

Unmanned Aerial Vehicles (UAVs) are widely used today in military, commercial, and environmental applications because they are versatile, agile, and cost-effective. Some examples include, aerial photography, delivery service, and search and rescue operations. One key factor that affects how well a UAV performs is its pitch dynamics, which control the aircraft's up and down motion. Good control of pitch is essential for stability, maneuverability, and overall safety.

In this project, we study the pitch dynamics of UAVs using a simple mathematical model. We will try to find out how physical parameters such as damping, stiffness, and moment of inertia affect the dynamic response. We will see how effectively a feedback loop (proportional derivative controller) can lessen the oscillations that occur in the pitch angle. Finally, we will try to find out what the effective regions are (under-damped, overdamped, or critically damped) under what our system should operate.

Our model is based on a second-order differential equation that describes how the pitch angle changes over time, taking into account factors like inertia, damping, and aerodynamic stiffness, as well as external control inputs.

We examine the effect of key physical parameters, including moment of inertia, damping, and stiffness, on the UAV's pitch response. These parameters are analyzed in terms of their impact on performance measures such as settling time, overshoot, and oscillation amplitude. We compare the performance of a simple step input with that of a proportional-derivative (PD) controller under different initial conditions. In addition, we evaluate how variations in controller gains influence the overall system performance and determine the conditions under which the system behaves as under-damped, critically damped, or overdamped.

Our work uses the fourth-order Runge-Kutta (RK4) method to perform accurate numerical simulations of the UAV pitch dynamics. The results from our simulations help us understand the system better and guide the design of improved and robust control systems for future UAVs.

2 Mathematical Model and Numerical Method

To investigate UAV pitch dynamics, we begin with a simplified physical model assuming planar motion about the lateral (pitch) axis. The rotational dynamics can be derived from Euler's equations of motion. For pitch dynamics about the body's y-axis, the governing second-order ordinary differential equation (ODE) is given by

$$I_{yy}\,\ddot{\theta}(t) + c\,\dot{\theta}(t) + k\,\theta(t) = u(t), \tag{1}$$

where $\theta(t)$ denotes the pitch angle (in radians) at time t, I_{yy} is the moment of inertia about the y-axis, c is the damping coefficient, k is the aerodynamic stiffness, and $u(t)$ is the external control moment applied to the system.

2.1 Conversion to a First-Order System

For numerical integration, it is more convenient to convert the second-order ODE into a system of first-order equations. We define a new variable $q(t)$ as the pitch rate:

$$q(t) = \dot{\theta}(t).$$

Thus, the ODE in (1) can be rewritten as the following system of first-order equations:

$$\dot{\theta}(t) = q(t), \tag{2}$$

$$\dot{q}(t) = -\frac{c}{I_{yy}}\,q(t) - \frac{k}{I_{yy}}\,\theta(t) + \frac{u(t)}{I_{yy}}. \tag{3}$$

The state vector is defined as

$$\mathbf{x}(t) = \begin{bmatrix} \theta(t) \\ q(t) \end{bmatrix},$$

so that the system becomes

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)) = \begin{bmatrix} q(t) \\ -\frac{c}{I_{yy}}\, q(t) - \frac{k}{I_{yy}}\, \theta(t) + \frac{u(t)}{I_{yy}} \end{bmatrix}.$$

2.2 Fourth-Order Runge-Kutta (RK4) Integration Scheme

To solve the state equations, we use the fourth-order Runge-Kutta (RK4) method. Given the state $\mathbf{x}_n = \mathbf{x}(t_n)$ at time t_n and a time step Δt, the RK4 method computes the state at the next time step $t_{n+1} = t_n + \Delta t$ as follows:

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}(t_n, \mathbf{x}_n), \\
\mathbf{k}_2 &= \mathbf{f}\Big(t_n + \frac{\Delta t}{2},\, \mathbf{x}_n + \frac{\Delta t}{2}\, \mathbf{k}_1\Big), \\
\mathbf{k}_3 &= \mathbf{f}\Big(t_n + \frac{\Delta t}{2},\, \mathbf{x}_n + \frac{\Delta t}{2}\, \mathbf{k}_2\Big), \\
\mathbf{k}_4 &= \mathbf{f}(t_n + \Delta t,\, \mathbf{x}_n + \Delta t\, \mathbf{k}_3), \\
\mathbf{x}_{n+1} &= \mathbf{x}_n + \frac{\Delta t}{6}\Big(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\Big).
\end{aligned}$$

This method has a local truncation error of $\mathcal{O}(\Delta t^5)$ and a global error of $\mathcal{O}(\Delta t^4)$. The RK4 method is efficient and accurate, making it well-suited for capturing the transient behavior of the UAV pitch dynamics.

2.3 Control Input and Simulation Procedure

The control input $u(t)$ varies with time and is updated at each integration step. Depending on the experimental setup, $u(t)$ may be a simple step input or determined by a proportional-derivative (PD) controller. In the case of a PD controller, the control law is

$$u(t) = K_p\, (\theta_{\text{ref}} - \theta(t)) - K_d\, q(t), \tag{4}$$

where K_p and K_d are the proportional and derivative gains, respectively, and θ_{ref} is the target pitch angle. After every RK4 step, the new state $\mathbf{x}_{n+1}$ is computed, and the values of $\theta(t)$ and $q(t)$ are recorded. This data is then used to generate time-series plots, phase portraits, and to extract performance metrics such as overshoot and settling time.

The simulation begins by setting the initial conditions, $\theta(0)$ and $q(0)$, and defining the simulation parameters including I_{yy}, c, k, and, if applicable, the control gains K_p and K_d. A time step Δt is chosen to balance accuracy and computational efficiency. The RK4 integration is then applied iteratively across the simulation period, with the control input updated at each step and the state vector stored for further analysis. This systematic approach ensures a reliable and efficient simulation of the UAV pitch dynamics.

2.4 Justification and Error Analysis of the Numerical Method

We study the UAV pitch dynamics described by

$$\ddot{\theta}(t) + \frac{c}{I_{yy}}\dot{\theta}(t) + \frac{k}{I_{yy}}\theta(t) = \frac{u(t)}{I_{yy}}, \tag{5}$$

using the RK4 method because it is both accurate and stable for our non-stiff system. The RK4 method has a local error of $\mathcal{O}(\Delta t^5)$ and a global error of $\mathcal{O}(\Delta t^4)$. Although it requires four function evaluations per time step, the low dimensionality of our system keeps the computational cost low. The method is derived by matching the Taylor series expansion of the true solution, thereby canceling lower-order error terms. For a linear test problem $y'(t) = \lambda y(t)$, the stability of RK4 is characterized by the stability function

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}, \tag{6}$$

with $z = \lambda \Delta t$. The method remains stable if $|R(z)| \leq 1$. With proper selection of Δt, the RK4 method is stable for the parameters used in our UAV pitch dynamics study.

2.5 Algorithmic Summary and Implementation Validation

To update the solution from t_k to $t_{k+1} = t_k + \Delta t$, we compute the intermediate slopes $\mathbf{k}_1$ through $\mathbf{k}_4$ and combine them as described above. We verify our implementation by running an error convergence study. In this study, a homogeneous version of the UAV pitch dynamics (with $u(t) = 0$) is simulated using initial conditions $\theta(0) = 1$ and $\dot{\theta}(0) = 0$ with parameters $I_{yy} = 1.0$, $k = 2.0$, and $c = 0.5$. The analytical solution is obtained by solving the characteristic equation

$$r^2 + \frac{c}{I_{yy}} r + \frac{k}{I_{yy}} = 0. \tag{7}$$

Under under-damped conditions, we define the natural frequency as $\omega_n = \sqrt{\frac{k}{I_{yy}}}$ and the damped frequency as $\omega_d = \sqrt{\omega_n^2 - \left(\frac{c}{2I_{yy}}\right)^2}$. The analytical solution takes the form

$$\theta(t) = e^{-\frac{c}{2I_{yy}} t} \left(\cos(\omega_d t) + \frac{\frac{c}{2I_{yy}}}{\omega_d} \sin(\omega_d t) \right). \tag{8}$$

By simulating the system using various Δt values over a fixed time interval and comparing the numerical solution with the analytical solution, we compute the absolute error. A log-log plot of the error versus Δt shows a slope of approximately 4, confirming the fourth-order convergence of our implementation. We also examine how the time step affects final errors and settling time to select an optimal Δt.

In summary, the Fourth-Order Runge-Kutta method is well-suited for our study due to its high accuracy, robust stability for non-stiff systems, and efficient computational cost. This numerical approach provides a solid framework for investigating the dynamic behavior and parameter sensitivity of the UAV pitch system.

2.6 Demonstration of Correct Implementation

To confirm the accuracy of our RK4 implementation, we perform an error convergence study and evaluate the impact of the time step. First, we simulate the homogeneous UAV pitch dynamics (with $u(t) = 0$) using the initial conditions $\theta(0) = 1$ and $\dot{\theta}(0) = 0$ and compare the numerical results with the analytical solution derived from the characteristic equation. The error plotted on a log-log scale shows fourth-order convergence. Second, we assess the effect of different Δt values on both the final error and the settling time, visually confirming that our chosen time step accurately captures the transient and steady-state behavior. The Python code that implements these tests is provided in the supplementary file (`AE370_Project_1_Code.ipynb`).

This comprehensive method forms the backbone of our numerical study of UAV pitch dynamics and ensures that we capture the system behavior and parameter sensitivity with high fidelity.

3 Simulation Setup

The simulation of UAV pitch dynamics was implemented in our Python Notebook using the RK4 method, as described earlier. Two types of control inputs are tested: a step input and a proportional–derivative (PD) controller.

For the step control, the input is defined as

$$u(t) = \begin{cases} 0, & t < 2\,\mathrm{s}, \\ U_0, & t \geq 2\,\mathrm{s}, \end{cases}$$

with $U_0 = 1.0\,\mathrm{N{\cdot}m}$. For the PD controller, the control law is

$$u(t) = -K_p\,\theta(t) - K_d\,q(t), \tag{9}$$

where $K_p = 4.0$ and $K_d = 1.0$. These values and equations are set up directly in the notebook.

The physical parameters are also defined in the code. The moment of inertia is set to $I_{yy} = 1.0\,\mathrm{kg}\cdot\mathrm{m}^2$ and the aerodynamic stiffness to $k = 2.0\,\mathrm{N{\cdot}m/rad}$. The damping coefficient c is varied over values such as 0.5, 1.0, 1.5, and 2.0 N·m·s/rad to study its effect on the pitch response.

The simulation runs over a 6-second interval and uses a time step of $\Delta t = 0.001\,\mathrm{s}$, as chosen by our convergence analysis. In the code, the initial conditions are set so that the UAV is at rest: $\theta(0) = 0\,\mathrm{rad}$ and $q(0) = 0\,\mathrm{rad/s}$. The RK4 method is then applied in a loop over the simulation time. At each step, the code updates the control input $u(t)$ based on the chosen control law, computes the new state vector $\mathbf{x}(t) = \begin{bmatrix} \theta(t) \\ q(t) \end{bmatrix}$, and saves these values.

The recorded data is later used to generate time-series plots, phase portraits, and to calculate performance metrics such as overshoot and settling time. All of the simulation setup and analysis code is included in the IPython Notebook file (`AE370_Project_1_Code.ipynb`).

4 Results and Discussion

This section presents the simulation outcomes that address the key research questions outlined earlier. The primary focus is to evaluate how the physical parameters and control strategies impact the dynamic response of the UAV pitch model. Specifically, we investigate the following:

- The transient and steady-state behavior of the pitch angle in response to a step control input. Metrics such as overshoot, settling time, and oscillatory behavior are quantified to assess the effect of varying damping and other physical parameters.
- A comparative analysis of two control strategies—namely, a simple step control input and a proportional–derivative (PD) controller—to determine their effectiveness in stabilizing the pitch dynamics.
- The influence of damping on system stability, which is further explored through phase-plane portraits that visually illustrate the transition from under-damped to overdamped behavior.

The results are presented in a series of figures that include time-series plots of the pitch response and phase portraits. These visualizations, along with the extracted quantitative metrics, provide a comprehensive understanding of the UAV pitch dynamics and offer valuable insights for control design and parameter tuning.

In the following subsections, we describe each set of results in detail and discuss how they inform our understanding of the system's performance.

4.1 Error Convergence Study

To validate our numerical integration of the UAV pitch dynamics using the fourth-order Runge-Kutta (RK4) method, we conducted an error convergence study. In this study, the system was simulated with a variety of time step sizes, Δt, and the absolute error at a fixed final time, T, was computed by comparing the numerical solution with the analytical solution of the homogeneous damped oscillator. The convergence plot is presented on a log-log scale and shows a clear linear trend. The slope of the fitted line corresponds to the order of convergence of the numerical method. For the RK4 method, the global error is expected to scale as $\mathcal{O}(\Delta t^4)$; our experimental results confirm this expectation with an observed slope of approximately 4. This finding not only verifies the accuracy of our RK4 implementation, but also ensures that the chosen time step is sufficiently small to capture the dynamic behavior of the UAV pitch model accurately.

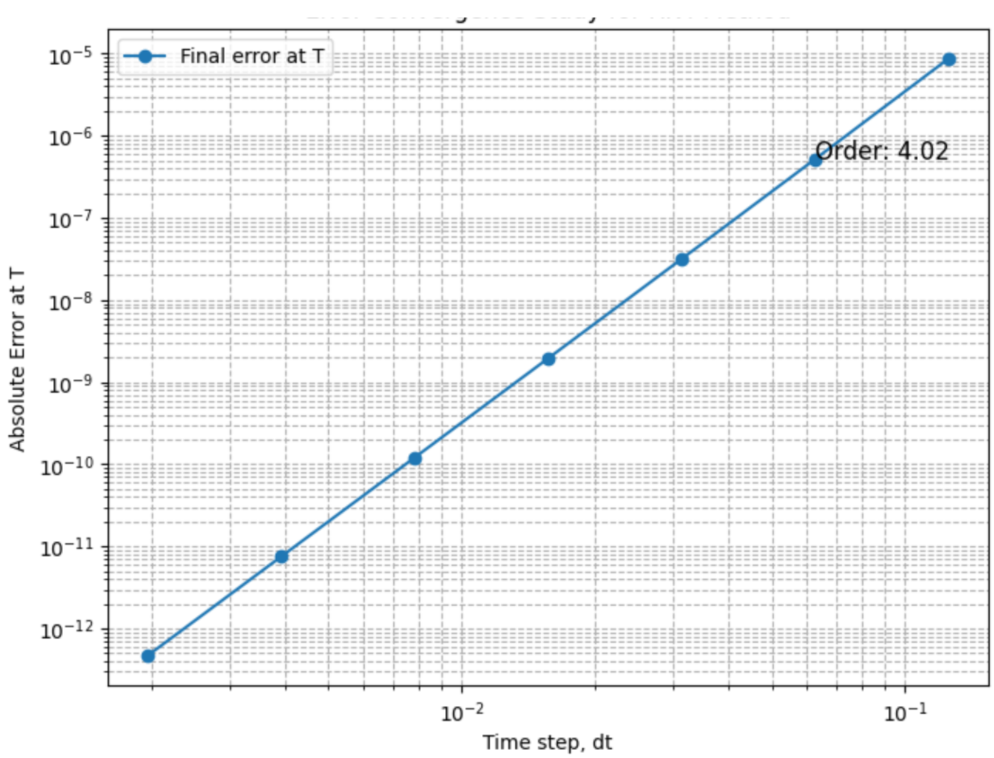

Figure 1: Convergence plot of the absolute error versus time step Δt on a log-log scale. The linear trend with a slope of approximately 4 verifies the global error of $\mathcal{O}(\Delta t^4)$ for the RK4 method.

4.2 Function Plot and Comparison of Time Steps

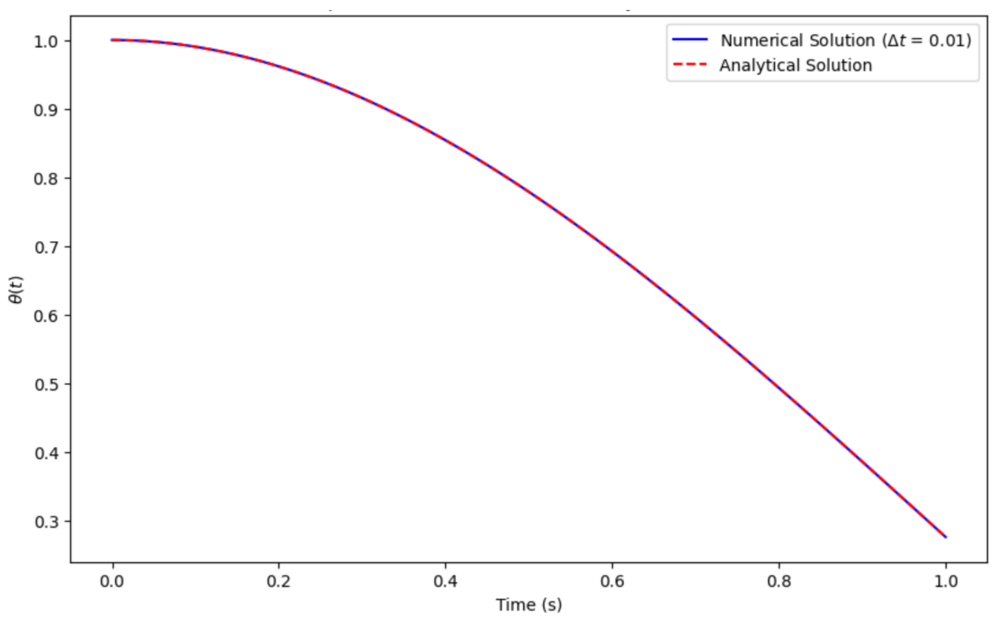

Figure 2: Function Plot of the Numerical Solution comparing to the Analytical Solution

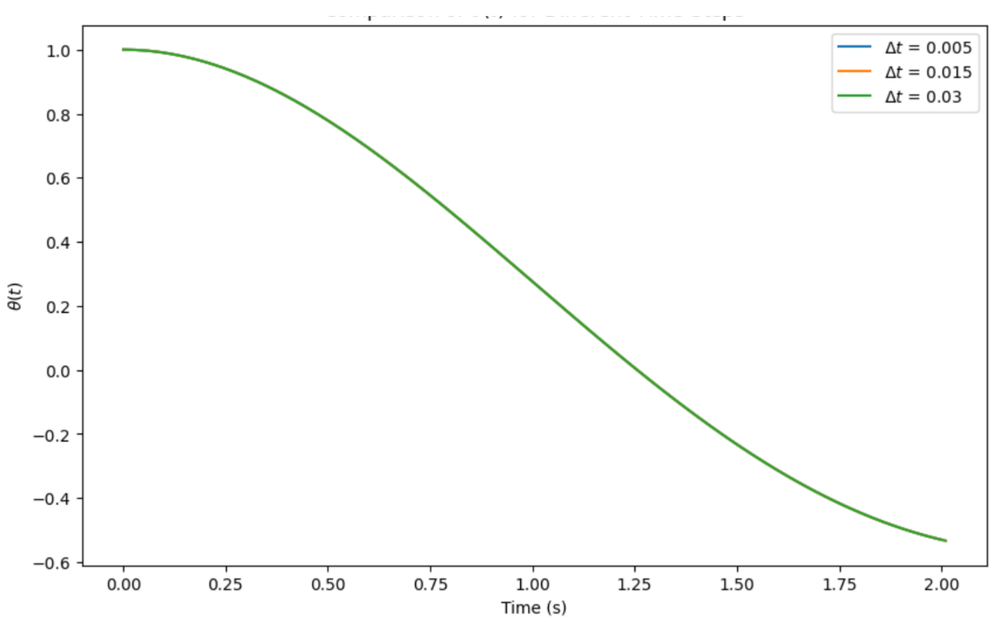

Figure 3: Comparison of $\theta(t)$ with Time Steps $\Delta(t) = 0.005$, $\Delta(t) = 0.015$, $\Delta(t) = 0.03$

In order to analyze whether the RK4 method was valid numerical method, a numerically iterated solution was plotted over a random harmonic-damped oscillator system at a time step of 0.01. The numerical system matched the analytical system perfectly, validating the RK4 for our specific dynamical system.

Our next step was to find out the correct time step. To do this we plotted time steps of 0.005, 0.015, and 0.03. We found that they all plotted very similar graphs, showing us that anything on the order of 0.3 or less were valid time steps.

4.3 Step Input Response

To better understand how our pitching system works, we plotted the it under three differing conditions: under-damped, critically damped, and overdamped oscillation. What we expect in for a general oscillatory system is to see that our system follows overshoot theory where

$$OS \approx \exp\left(-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right). \tag{10}$$

is the generalized equation. The overshoot is minimized at the critical damping ratio, much greater at while under-damped, and oscillates while overdamped The equation that describes our critical damping ratio is as follows:

$$\zeta = \frac{c}{2\sqrt{I_{yy}k,}} \tag{11}$$

where c, I_{yy}, and k are inputs of our simulation. When $zeta$ = 1 our system is critically damped, when it is less than one our system is under-damped and when its greater it is overdamped. Below is the results of our simulation.

Figure 4 shows the pitch response $\theta(t)$ under a step control input applied at $t = 2$ seconds. Three damping cases are shown:

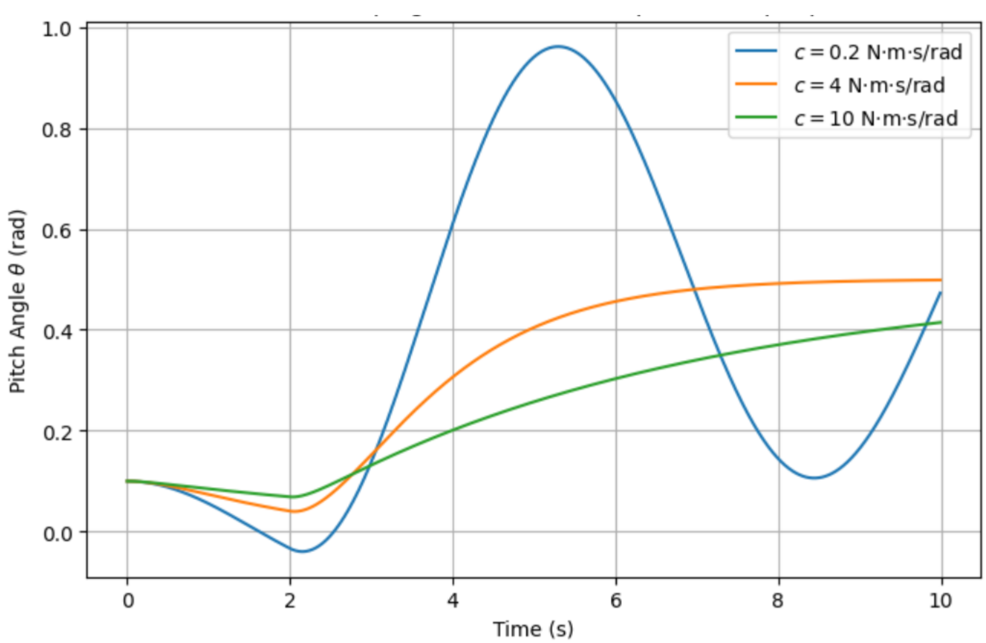

Figure 4: Pitch response $\theta(t)$ for damping values $c = 0.2$, 4, and 10 N·m·s/rad under a step input at $t = 2$ s.

The pitching system follows with our expected results. As we can see, at an under-damped oscillation ($c = 0.2$), the system will oscillate at a lower and lower amplitude, at a critically damped oscillation ($c = 4$), the pitch angle will reach zero in period, and at an overdamped oscillation($c = 10$), the system will never oscillate.

The pitching system follows our expected results. As we can see, at an under-damped condition ($c = 0.2$), the system exhibits oscillatory behavior with gradually decreasing amplitudes, which is indicative of energy dissipation over time. This behavior confirms that the system, when lightly damped, will overshoot its equilibrium value and then slowly settle as the oscillations diminish. In the critically damped scenario ($c = 4$), the pitch angle reaches zero in a single, smooth motion without any oscillatory behavior. This is the optimum condition for a rapid return to equilibrium without overshoot, which is desirable in many control applications. On the other hand, at an overdamped condition ($c = 10$), the system does not oscillate at all; however, the convergence to equilibrium is significantly slower. Although the absence of oscillations can be beneficial, the sluggish response might not be ideal for dynamic control where a fast response is necessary.

These results are consistent with theoretical expectations for second-order systems. The under-damped case demonstrates the trade-off between a quick but oscillatory transient and the energy dissipation required for stability. Critical damping represents the balance point where the system rapidly reaches equilibrium without oscillation. Conversely, the overdamped condition, while stable, results in a slower overall response. This analysis not only validates our model but also highlights the importance of tuning the damping parameter to achieve the desired transient behavior and performance in UAV pitch control.

4.4 Phase Portrait Analysis

The phase portraits in Figure 5 plot $\theta(t)$ against pitch rate $q(t)$ and show how damping changes system behavior.

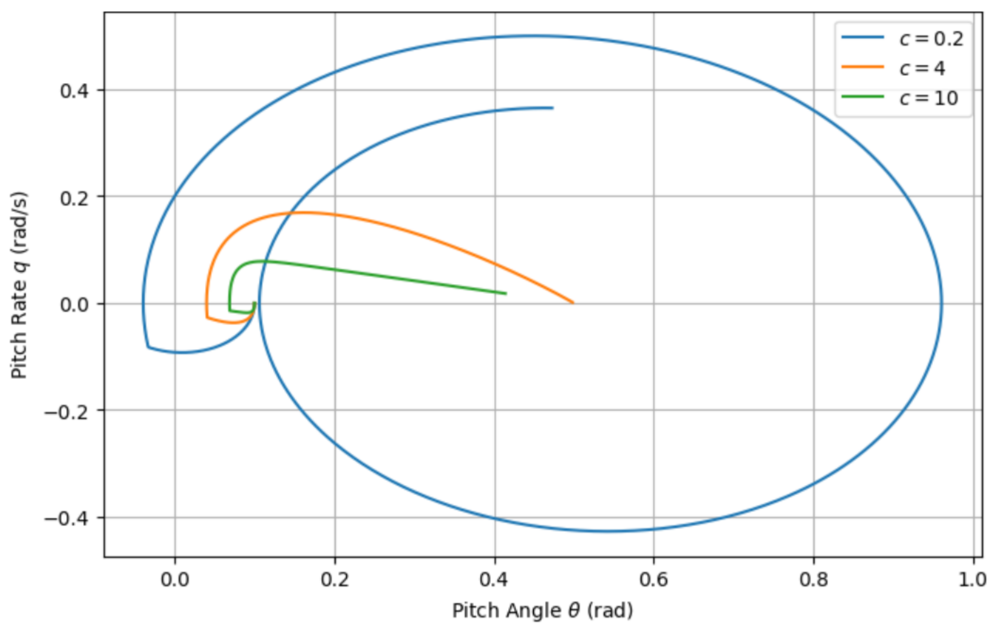

Figure 5: Phase portraits of $\theta(t)$ vs $q(t)$ for different damping values.

What these phase portraits show us is a better understanding of our pitch angle graphs. The under-damped oscillation will continue to rotate around the phase plot until infinity. This is shown by the phase plot circling the pitch angle and rate. The critically damped oscillation reaches the origin in one oscillation, and the overdamped system will theoretically never reach the origin.

These plots are important because it shows us the behavior our PD controller wants to avoid. Underdamped oscillation leads to the greatest pitching disparity. This is also reflected in the overshoot equation mentioned above. Ideally, our PD controller either acts as a critically damped or overdamped system, as it minimizes oscillations.

4.5 Step vs PD Controller

Figure 6 compares the response using a step input and a PD controller.

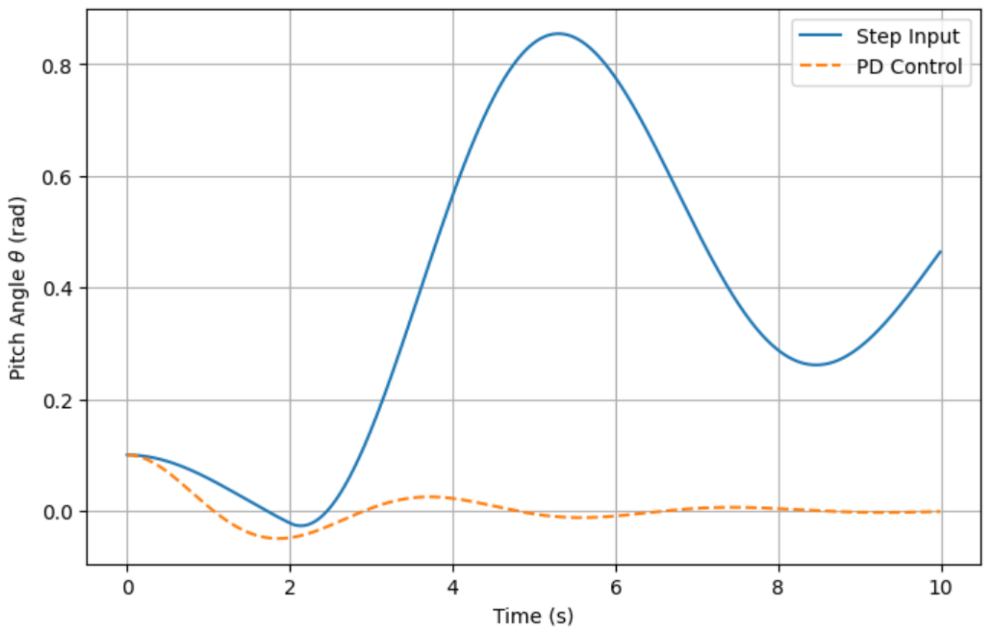

Figure 6: Comparison of $\theta(t)$ using step control vs PD control ($K_p = 4.0$, $K_d = 1.0$).

The PD controller stabilizes the pitch faster and with fewer oscillations. It adjusts the control moment in real-time, reducing overshoot and improving settling time compared to a fixed step input. This is exactly what we expected it to do, as it matches how an overdamped system would act. While creating a critically damped system would be ideal, in the real world, overdamped systems are used to as they are more realistic.

The comparison between the step control input and the PD controller is illustrated in Figure 6. In the case of the step control, the pitch angle $\theta(t)$ exhibits a pronounced transient response, characterized by significant overshoot and oscillations as the system reacts to the sudden application of the control moment at $t = 2$ s. This response shows that while the step input forces the system, it does not actively attenuate the induced oscillations, resulting in a prolonged settling time. In contrast, the PD controller, which continuously adjusts the control input based on both the current pitch angle and its rate of change, produces a notably smoother response. The PD-controlled response converges to equilibrium more rapidly, with a reduced overshoot and minimal oscillatory behavior. This improvement highlights the advantage of implementing feedback control strategies that not only respond to the error but also counteract the momentum of the system. Overall, the PD controller proves to be more effective in stabilizing the UAV pitch dynamics, ensuring a faster and more controlled transition to steady state.

4.6 Summary

Higher damping leads to smoother, faster stabilization. The PD controller outperforms the step input by adapting in real-time, improving stability and performance. Both time-domain plots and phase portraits support the analytical predictions and demonstrate the effectiveness of the chosen control strategies.

5 Conclusion

This study presented a numerical investigation of UAV pitch dynamics using the fourth-order Runge-Kutta (RK4) method. We developed a simple mathematical model based on a second-order differential equation to capture the key behavior of the UAV's pitch motion under different physical parameters and control strategies.

Our results show that the transient response is very sensitive to the damping coefficient c. Lower damping results in an under-damped response with large overshoot and oscillations, while higher damping leads to a smoother, faster approach to steady state. These outcomes agree with our theoretical predictions using the damping ratio $\zeta = \frac{c}{2\sqrt{I_{yy}\,k}}$ and overshoot formula $OS \approx \exp\left(-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right)$.

We also compared a step input control with a proportional–derivative (PD) controller. The PD controller gave a faster and smoother convergence to equilibrium than the step input, thanks to its real-time adjustment of the control moment. Time-series and phase-plane analyses confirmed these improvements.

Overall, our work provides practical guidance for tuning controller parameters and improving the stability and responsiveness of UAV pitch systems. The full Python code and simulation notebooks are provided as supplementary material, which can be used for further studies. Future research will extend this model to include nonlinear aerodynamic effects and adaptive control strategies.

Group Contributions

This project was a group effort between all members, with each person contributing meaningfully to at least one of the rubric items (1)–(6). The division of work was as follows:

Sufyan Syed (20%): He was our team lead in terms of generating an idea for this project, as well as primarily being responsible for formulating the mathematical equations of motion and validating the assumptions underlying the physical system for our methods section. He also aided in iterating the code for the results section with out plots. His work contributed mainly to these rubric sections (1), (3), (4), and (6).

Benjamin Litvak (20%): He mainly focused on implementing the numerical method in Python through iteration as well as iterate the code for the results section with the plots, and also conducting convergence analyses. He also worked on the general report by making sure the plots look good in our document and wrote the contributions section. His work contributed significantly to rubric items (3), (4), and (5).

Samik Singh (20%): He took the lead in formatting the report for this project. He helped write the latex math for our document. He also helped to write the mathematical model, simulation setup, discussion, and other data sections of the report. Samik also helped with the iteration of the code when finding our results for this project. His work contributed significantly to rubric sections (2), (3), and (4).

Angelina Yan (20%): She also took the lead in iterating the code for this project. She also helped us with finding the topic of UAV pitch dynamics to start this project. Angelina contributed to the numerical methods section by iterating on the math involved with our topic and also wrote our nomenclature. SHe also helped with making the document look nice. Her work contributed significantly to rubric sections (1), (2), and (3).

Suraj Duvvapu (20%): He mainly worked on iterating the code to make sure our math and plots looked correct, and also worked on aspects on our report like the Introduction, conclusion, and the results and discussion sections by making sure the information being presented was clear and correct. His work contributed significantly to rubric sections (1), (3) and (4).

All members reviewed the final report and provided feedback on each section to ensure consistency and technical accuracy. The contributions percentages reflect the effort spent on research, implementation, writing, and coordination across the entire length of the project.

Appendix

The complete Python code for the simulations and figure generation is included in the supplementary file(`AE370_Project_1_Code.ipynb`). The project code and detailed documentation are available on our GitHub repository: https://github.com/ssyed68/AE370-Project1. .

References

The python code that was used for this project was done with the help of ChatGPT. We specifically used ChatGPT to help us with using the RK4 method, as well as helping us with our methods and our plots for the results portion.