ES
Engineering and Science

Rodrigues e Lima, ES Engineering and Science 2026, 15:2
Edição Especial

Analysis of feedforward control approaches for autonomous quadcopters

Análises de abordagens de controle feedforward para quadricópteros autônomos

[1]Vinícius Freitas Rodrigues, [2]Gabriela Vieira Lima

[1]Mestrando em Engenharia Elétrica no Programa de Pós-Graduação de Engenharia Elétrica (PPGEELT) – Universidade Federal de Uberlândia (vinicius.freitas@ufu.br)
[2]Professor Adjunto da Faculdade de Engenharia Elétrica (FEELT) – Universidade Federal de Uberlândia (gabriela.lima@ufu.br)

ABSTRACT: This paper bridges the conceptual gap between trajectory-based feedforward control, common in academic motion control literature, and the dynamic inversion approach used in practical open-source flight control firmwares like Ardupilot. While classical methods aim for zero tracking error by anticipating reference trajectories, practical implementations prioritize actuator smoothness and effort prediction. A comparative analysis is conducted on the z-axis of a simulated quadcopter across 162 scenarios, varying velocity and acceleration feedforward gains. Performance is evaluated using Integral Absolute Error (IAE), Integral of Squared Input (ISU), and Total Variation (TV). Results indicate that while the classical approach reduces position tracking error by up to ten times, it induces severe velocity overshoots (up to 36.1%) and extreme actuator chattering at high acceleration gains. The study concludes that aggressive feedforward tuning degrades physical performance, validating the stability-focused architecture adopted by available flight controllers.

Keywords: Controls. Feedforward. Quadcopter. Ardupilot. Performance Assessment.

***RESUMO*:** Este artigo preenche a lacuna conceitual entre o controle feedforward baseado em perfil de trajetória, comum na literatura acadêmica sobre controle de movimento, e a abordagem de inversão dinâmica usada em firmwares de controle de voo de código aberto, como o Ardupilot. Enquanto os métodos clássicos visam erro de rastreamento zero antecipando trajetórias de referência, as implementações práticas priorizam a suavidade do atuador e a predição do esforço de controle. Uma análise comparativa é conduzida no eixo z de um quadricóptero simulado em 162 cenários, variando os ganhos de feedforward de velocidade e aceleração. O desempenho é avaliado usando o Erro Absoluto Integrado (IAE), a Integral do Quadrado da Entrada (ISU) e a Variação Total (TV). Os resultados indicam que, embora a abordagem clássica reduza o erro de rastreamento de posição em até dez vezes, ela induz ultrapassagens severas de velocidade (até 36,1%) e oscilações extremas do atuador em altos ganhos de aceleração. O estudo conclui que o ajuste agressivo do feedforward degrada o desempenho físico, justificando a arquitetura focada em estabilidade adotada pelas controladoras de voo disponíveis.

Palavras-chave: Controle. Feedforward. Quadricóptero. Ardupilot. Análise de Desempenho.

Recebido em: 25/06/2026
Publicado em: 07/07/2026

1. INTRODUCTION

In recent years, research on Unmanned Aerial Vehicles (UAVs) has grown significantly and is expected to remain extensive due to variations in design, technology, applications, and regulatory requirements. These differences influence both the dynamic modeling of the aircraft and the design of their control strategies.

Academic literature shows that modern control techniques for rotary wing aircraft are vast and that trends focus on optimized or flexibly parameterized algorithms. Seeking to meet these demands justifies the academic community's focus on model-based or learning-based algorithms. Even so, major drone manufacturers such as DJI, Parrot, 3D Robotics, and AscTec continue to use PID-based control (Li et al. 2015).

PID has been studied for UAVs since the earliest works on the subject to the most recent. Bouabdallah et al. (2004) analyzes attitude control and disturbance rejection in hovering, Li and Li (2011) expands this work by including position control, Lima et al. (2015) presents this strategy for trajectory tracking, and Pinto et al. (2019) uses PID to control all 12 state variables of the system. However, these studies, like most others, only analyze the pure PID or implement the classic technique to establish a benchmark with modern algorithms.

Open-source projects like Ardupilot also rely on the use of classical control (Ardupilot Development Team 2024a). This does not mean, however, that the control architecture of these systems is purely PID-based. Ardupilot, for example, uses feedforward, digital filtering, and applies non-linear techniques to complement the limitations of classical control. This work seeks to analyze in detail one of these adaptations used in conjunction with standard feedback control: the feedforward.

The use of feedforward control is constantly discussed in the literature for controlling industrial servomechanisms, robotic manipulators, and processes. However, studies on this topic are academically neglected for application in UAVs, given the number of modern techniques available for such systems. Nonetheless, open-source flight control projects such as Ardupilot (Ardupilot Development Team 2024b) and Crazyflie (Lima 2019) make use of this feature in their control algorithms.

This work aims to contribute to the understanding of feedforward use in drones, and to this end it applies to qualitative and quantitative metrics. This helps to better understand particularities of the existing feedforward strategies for UAVs and provides a stronger basis for future benchmarking between commercial controllers and academic modern methods.

2. FUNDAMENTALS OF MOTION CONTROL AND FEEDFORWARD TECHNIQUES

Motion profiling is a control strategy that establishes a continuous and smooth trajectory, obtained through mathematical modeling, to be followed by a machine or robot. This approach coordinates position, velocity, and acceleration over time, avoiding abrupt variations.

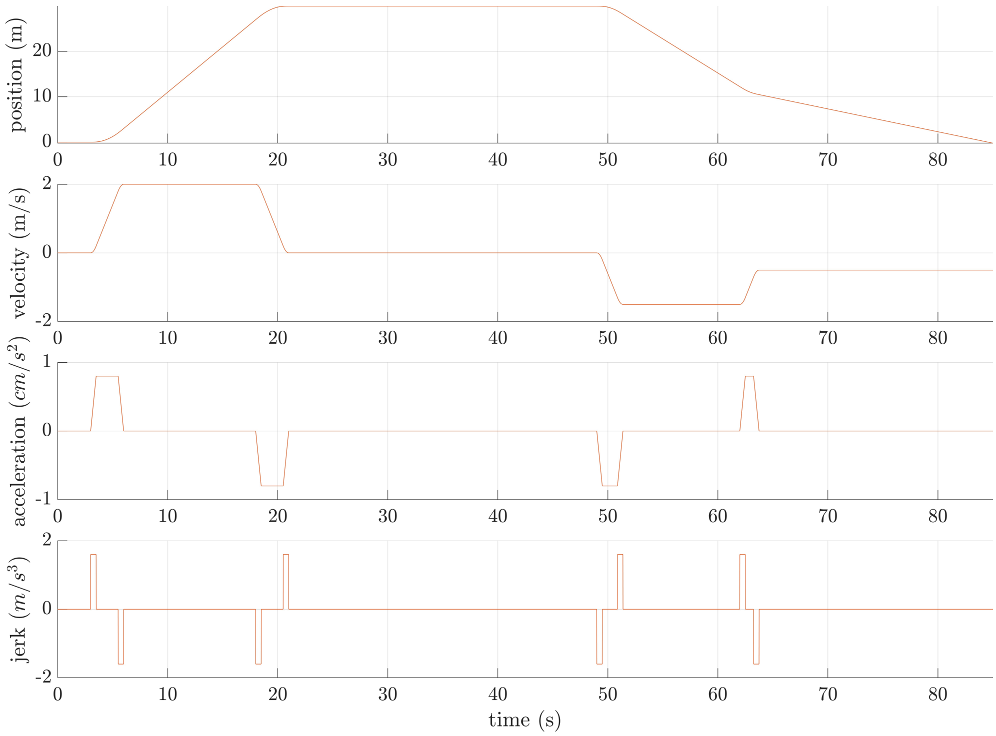
A traditional profile is the so called S-Curve. This technique generates trajectories in which the jerk (acceleration derivative) is finite, causing the trajectory to be as smooth as possible. The jerk limitation guarantees that no instantaneous force/torques and current commands appear on the motor. It reduces vibrations, improves the system's accuracy, and increases the motor's longevity. The Ardupilot also uses jerk limited trajectory generation algorithm. Figure 1 shows a S-curve trajectory for the takeoff, hovering, and landing of a quadcopter.

Once the motion profile is obtained, the position curve is entered as the setpoint of the control system. The most basic control structure used in these cases is a cascade in which speed control is enclosed within the position control loop.

There is also an even more internal loop responsible for controlling the current or rotation of the motors; however, this system is separate from the motion controller, and this task is assigned to electronic drivers.

ES - Engineering and Science ISSN: 2358-5390 DOI: 10.18607/ES20261521820 Volume 15, Número 2 – Edição Especial

Figure 1 – A jerk limited motion profile based on velocity s-curve. The profile depicted represents a simple takeoff and land trajectory in z-axis.



Source: The Author (2026).

Not only is this type of control used in industrial machines as shown in (Gurocak 2016), other works such as (Mahony et al. 2012) show that the control of multirotor drones follows the same cascade strategy. In this case, there is a cascade of position control, attitude control and speed control of the brushless motors.

In servo or UAV systems, tracking errors occur during movement, even if the integrator sets this error to zero after reaching the desired waypoint. A common approach in industrial motion control systems is to combine classical feedback techniques with feedforward control. By injecting a velocity-proportional command directly into the output, the system responds faster to changes in movement compared to the feedback-driven counterpart of the system, which only acts after the existence of a position error.

One of the ways to apply feedforward control, and the most seen in motion control literature, is getting the derivative commands from motion profile, multiplying by predetermined gains and adding it to the position control signal. This work uses Gurocak's book (Gurocak 2016) as reference for this type of structure, shown in Figure 2.

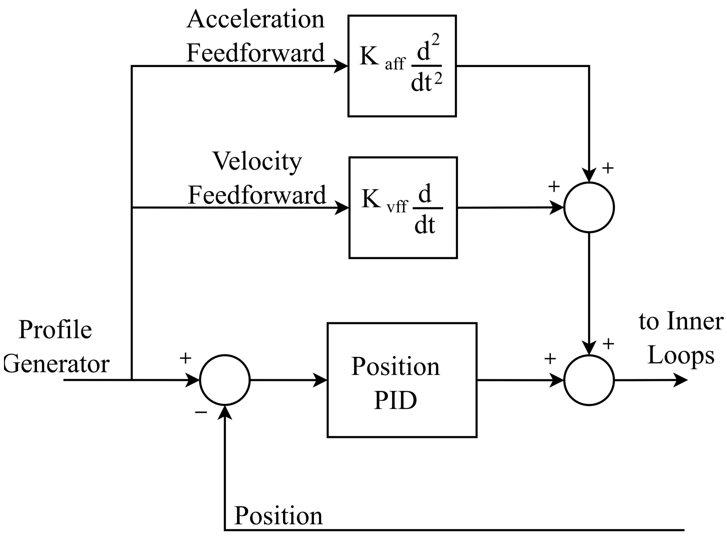
When the speed profile input is feedforward to the speed controller, the system responds faster to sudden changes in the desired trajectory. However, if the gain is set to a high value, the feedforward will create overshoots in the speed response.

One way to solve the overshoot caused by the velocity feedforward is by adding an additional feedforward action proportional to the second-order derivative of position. This so called acceleration feedforward eliminates the overshoots in speed while keeping the position tracking improvement.

Since servo errors can happen even in the jerk phases of the motion, further feedforward gains can be added. Jerk and jerk derivative feedforward are less common, but also used, for example in ultraprecision motion control applications.

Another philosophy for approaching the application of feedforward control is to treat it not as an anticipator of the position trajectory, but as a predictor of the control effort required. If disturbances are known to occur in a system, a feedforward term can be added to help the system reach the desired setpoints with less delay and less dependence on the integral term. This can be done through physical modeling of the disturbance, applying inverse transfer functions of the plant, or simply adding constant gains. This control technique is known as 2 degrees-of-freedom (2-DOF) PID, since it creates parallel paths to handle both disturbance rejection and setpoint following.

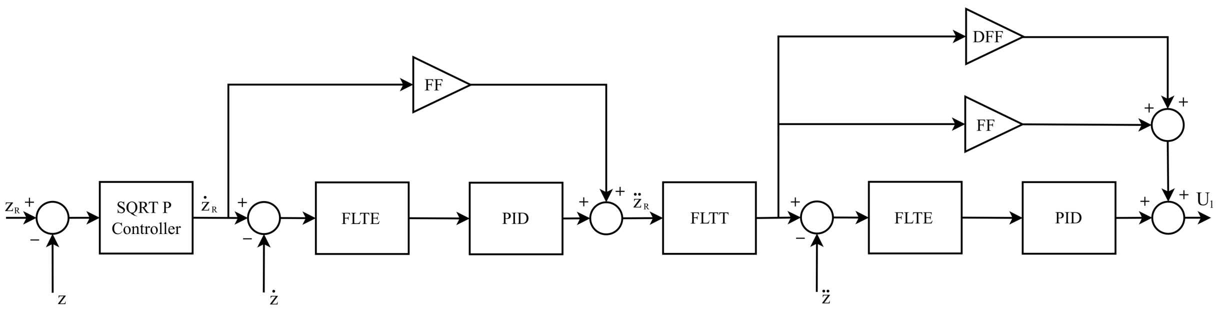
Figure 2 – A PID position controller with additional velocity and acceleration feedforward gains, a common structure in motion control literature.



Source: The Author (2026).

Figure 3 shows that Ardupilot uses 2-DOF PIDs with constant feedforward gains for its velocity and acceleration controllers. In this configuration, each of the controllers, except the position, have a feedforward action. The usage of feedforward remains the same in the x-axis and y-axis velocity controllers and in the roll, pitch and yaw rate controllers. The z-axis acceleration also uses an *acceleration derivative feedforward* gain, which has an equivalent function to a jerk feedforward seen in motion control literature.

Figure 3 – Ardupilot's altitude control loops. It contains a cascade control of position, velocity, and acceleration and uses PID, feedforward and digital filters.



Source: The Author (2026).

3. AN OVERVIEW ON QUADCOPTER MODELING AND ARDUPILOT'S CONTROL STRATEGIES

Most of the time, multirotor drones are modeled as rigid bodies, that is, bodies that do not deform regardless of the forces and torques exerted on them. In this sense, a commonly used approach is applying the Newton-Euler formalism, which translates the rigid body into a set of forces and torques.

Another detail to be considered in the dynamic modeling of a robotic system, such as a UAV, is the reference frames. The aircraft can be represented with a coordinate frame attached to it. Since the UAV is always attached to this frame, its position and orientation relative to the frame are always known. This coordinate frame B, fixed to the rigid body, is named the body frame (or non-inertial frame). Subsequently, it is necessary to relate the moving frame B to a fixed frame, relative to a fixed point on the Earth. This last coordinate frame E is named the Earth frame (or inertial frame). In the control system, it is common to aim to control the body in relation to a fixed reference point, while measurements from some sensors, such as IMUs, are taken in relation to the body frame.

To transform from one frame to another, the most common approach is to use homogeneous transformation matrices and Euler angles. The set of rotations known as Tait-Bryan angles (sequential x-y-z rotations) shown in Li and Li (2011) and an inverse transformation for angular quantities shown in Lima (2015) were used in this work.

Let the linear position and attitude in the inertial frame be defined as $\vec{\xi} = [x\, y\, z]^T$ and $\vec{\eta} = [\phi\, \theta\, \psi]^T$, respectively. Also assume that the resultant rotation of the 4 motors are considered to be Ω. Also, it is considered that the notation for the drones parameters and its numerical values are defined in Table 1. Using the Newton-Euler formalism and homogeneous transformation matrices, as demonstrated in Bouabdallah (2007) and Lima (2015), it is possible to obtain the set of differential equations that describe the motion of the body with respect to the inertial frame of reference by the Equations 1.

$$\begin{cases} \ddot{x} = \frac{1}{m}(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)U_1 \\ \ddot{y} = \frac{1}{m}(\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)U_1 \\ \ddot{z} = -g + \frac{1}{m}(\cos\phi\cos\theta)U_1 \\ \ddot{\phi} = \dfrac{(I_{yy} - I_{zz})}{I_{xx}} \cdot \dot{\theta} \cdot \dot{\psi} - \dfrac{J_r}{I_{xx}} \cdot \dot{\theta} \cdot \Omega + \dfrac{U_2}{I_{xx}} \\ \ddot{\theta} = \dfrac{(I_{zz} - I_{xx})}{I_{yy}} \cdot \dot{\phi} \cdot \dot{\psi} - \dfrac{J_r}{I_{yy}} \cdot \dot{\phi} \cdot \Omega + \dfrac{U_3}{I_{yy}} \\ \ddot{\psi} = \dfrac{(I_{xx} - I_{yy})}{I_{zz}} \cdot \dot{\phi} \cdot \dot{\theta} + \dfrac{U_4}{I_{zz}} \end{cases} \quad (1)$$

The next step in the mathematical model is to determine the propeller's rotational speeds. This is achieved by identifying the dynamics of brushless motors (Bouabdallah 2007), applying motor mixing algorithms (Ardupilot Development Team 2024b), or simply relating the propeller rotational speed to the thrust produced under steady-state conditions (Lima 2015).

The modeling in this work considered that the time constants of Brushless Direct-Current (BLDC) motors are much smaller than those of rigid body dynamics, and therefore the simulation only directly relates the thrust and torques required to the system's steady-state rotations.

Table 1 – Physical parameters of the quadcopter used in the simulations.

Name	**Parameter**	**Value**	**Unit**
Mass	m	1.2290	kg
Inertia on x axis	I_{xx}	1.3349e-02	$kg \cdot m^2$
Inertia on y axis	I_{yy}	1.4145e-02	$kg \cdot m^2$
Inertia on z axis	I_{zz}	2.1007e-02	$kg \cdot m^2$
Rotor inertia	J_r	1.4495e-07	$kg \cdot m^2$
Propeller thrust coefficient	a	1.4873e-08	$\frac{kgf}{rpm^2}$
Propeller drag coefficient	b	2.2037e-10	$\frac{kgf \cdot m}{rpm^2}$
Arm length	l	22.5	cm

Assume that the relationship between thrust/torque and rotor angular velocity is quadratic (Pinto et al. 2019), and that the drone has configuration X with the same motor numbering pattern as the Ardupilot. Then the relationship between the inputs generated by the control loops and the required speeds of the drone's motors is defined by Equation 2:

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1}{4}\left(\frac{U_1}{b} - \frac{\sqrt{2}}{bl}U_2 - \frac{\sqrt{2}}{bl}U_3 - \frac{U_4}{d}\right)} \\ \sqrt{\frac{1}{4}\left(\frac{U_1}{b} + \frac{\sqrt{2}}{bl}U_2 + \frac{\sqrt{2}}{bl}U_3 - \frac{U_4}{d}\right)} \\ \sqrt{\frac{1}{4}\left(\frac{U_1}{b} + \frac{\sqrt{2}}{bl}U_2 - \frac{\sqrt{2}}{bl}U_3 + \frac{U_4}{d}\right)} \\ \sqrt{\frac{1}{4}\left(\frac{U_1}{b} - \frac{\sqrt{2}}{bl}U_2 + \frac{\sqrt{2}}{bl}U_3 + \frac{U_4}{d}\right)} \end{bmatrix} \tag{2}$$

where the vector $[\Omega_1\ \Omega_2\ \Omega_3\ \Omega_4]^T$ contains the rotation speed of each motor.

To control multirotor drones, it is almost universally agreed that translational and rotational dynamics must be controlled separately. This is because angular movements in these drones are considerably faster than linear movements. Furthermore, there is strong coupling between movement in the xy plane and the roll and pitch angles. To resolve this issue, flight controllers use a cascade of translational and rotational control.

The Ardupilot not only uses this cascading strategy but also separates speed and position controls into further cascades. This is done for both rotational and translational subsystems. Furthermore, for the z-axis, there is a dedicated loop for acceleration control.

The Ardupilot firmware control structure also features numerous digital filters, including tracking filters (FLTT), error filters (FLTE), and derivative filters (FLTD). All of these filters are Exponential Moving Average (EMA) filters. Other features of the Ardupilot, these non-linear, include square root-limited proportional gain (SQRT Controller), clamping of the sum of the P and D control signals, and an adaptive slew rate limiter. Analysis of these features is beyond the scope of this work.

4. METHODOLOGY

This work evaluates the performance using a simulated model that implements both the non-linear system dynamics and part of the Ardupilot's control structure in Simulink. The same Ardupilot's architecture is also remodeled to use the feedforward gains similarly to the motion control literature. The pure derivative, shown for didactic purposes in Figure 2, cannot be used in reality because it would lead to inappropriately large gains at high frequencies. The simulation then uses the transfer function shown in Equation 3, which is a derivative with a first-order low-pass filter. A MATLAB batch script evaluates a takeoff and land trajectory for different feedforward gains in both Simulink models.

$$D(s) = \frac{50s}{s+50} \quad (3)$$

The takeoff and land motion profiles are extracted directly from an Ardupilot mission log that runs in a Software-in-the-loop (SITL) environment. The acceleration profile is obtained through differentiating the velocity signal and applying a Savitzky-Golay filter. It was found that the profile generated in this way has minimal variation with respect to the "Desired Acceleration Down" variable in the Ardupilot logs.

Each of the batch simulations generates a data-rich *.mat* file for later analysis. An initial qualitative analysis is performed by comparing the motion curves and percent overshoot (PO) on the z-axis. Since altitude control is the only one with 3 loops and the one with the lowest coupling to other variables, the analysis is performed only on that loop. In this test 6 scenarios are evaluated for each architecture.

To quantitatively evaluate the performance of the control systems, three main performance indexes are taken: the Integral of Absolute Error (IAE), the Integral Squared Input (ISU), and Total Variation (TV).

The IAE measures the total absolute deviation from the route. Unlike other performance indices, it does not concern itself with when the deviation occurs or the intensity of the peaks. It is defined by Equation 4.

$$IAE = \int_0^\infty |e(t)|dt \quad (4)$$

It is important not only to measure the output error but also to evaluate the control effort. In this sense, the chosen index is the ISU shown in Equation 5, which physically represents the energy spent to operate the system. It is also time-independent, but unlike the IAE, it penalizes the intensity of peaks more because it includes the squared error. In this case, the chosen index is squared because the power consumption in an electrical system (such as BLDC motors) as a function of the controlled variable is quadratic. The combination of IAE and ISU was used to compare optimal control systems for UAVs in Ahmdia et al. (2025).

$$ISU = \int_0^\infty u(t)^2 dt \quad (5)$$

Finally, TV quantifies the rate of change of the inputs. Physically, it represents how fast the actuators have to operate to deliver the inputs required by the control system. Systems with a high TV tend not to respect the physical limits of the actuator dynamics, and also represent greater wear on these components.

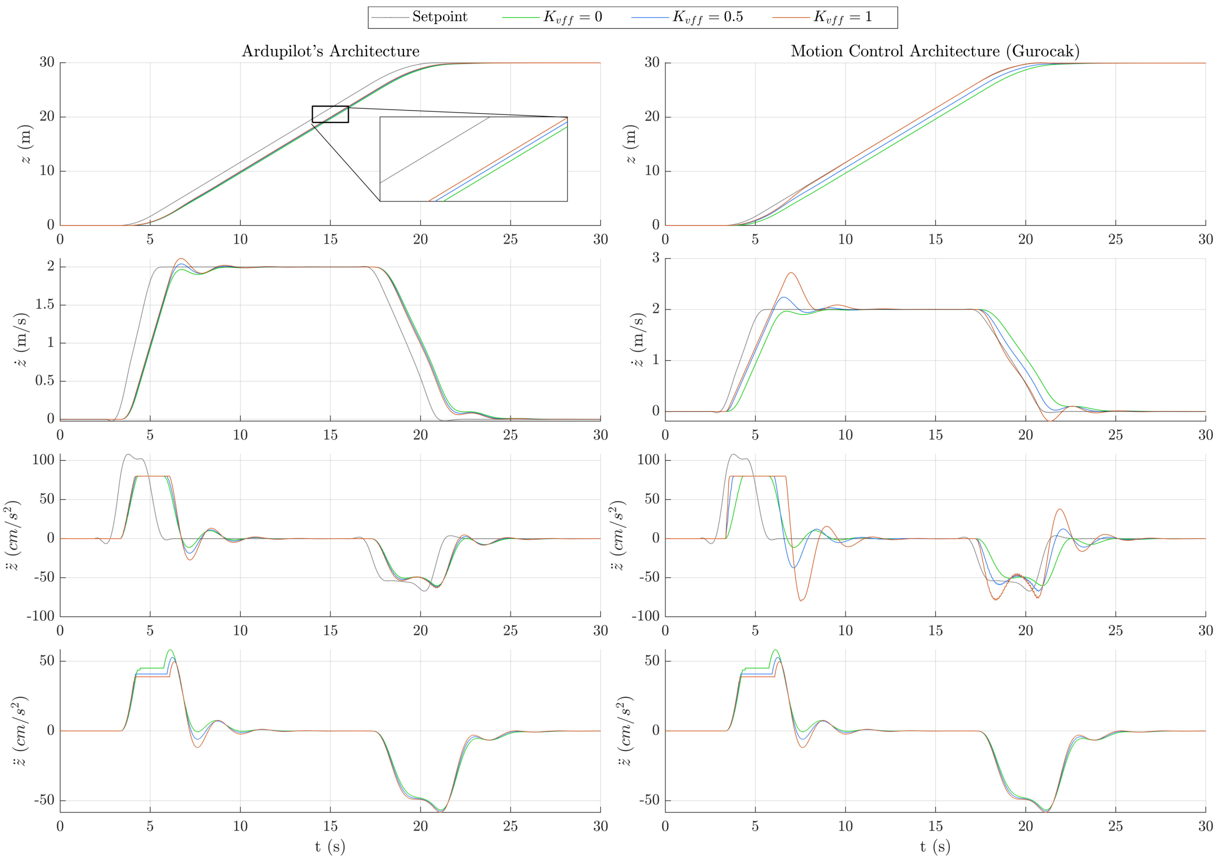
$$TV = \int_0^\infty \left|\frac{du(t)}{dt}\right| dt \tag{6}$$

To perform the quantitative analysis, a total of 162 scenarios are simulated. The batch script considers 81 combinations of K_{vff} and K_{aff} for both the Ardupilot's and Gurocak's architectures. In these simulations, all the PID gains and filters cutoff frequencies are configured using the Ardupilot's default values.

5. RESULTS AND DISCUSSION

To analyze the simulation results of the first 12 scenarios, graphs of the z-axis loop outputs are presented side-by-side. Initially, in Figure 4, only the addition of velocity feedforward is analyzed.

Figure 4 – Vertical axis outputs for different velocity feedforward gains for both Ardupilot's and Gurocak's Architecture in which, (a) and (b) are position, (c) and (d) velocity, (e) and (f) acceleration, and (g) and (h) integral signal amplitude.



Source: The Author (2026).

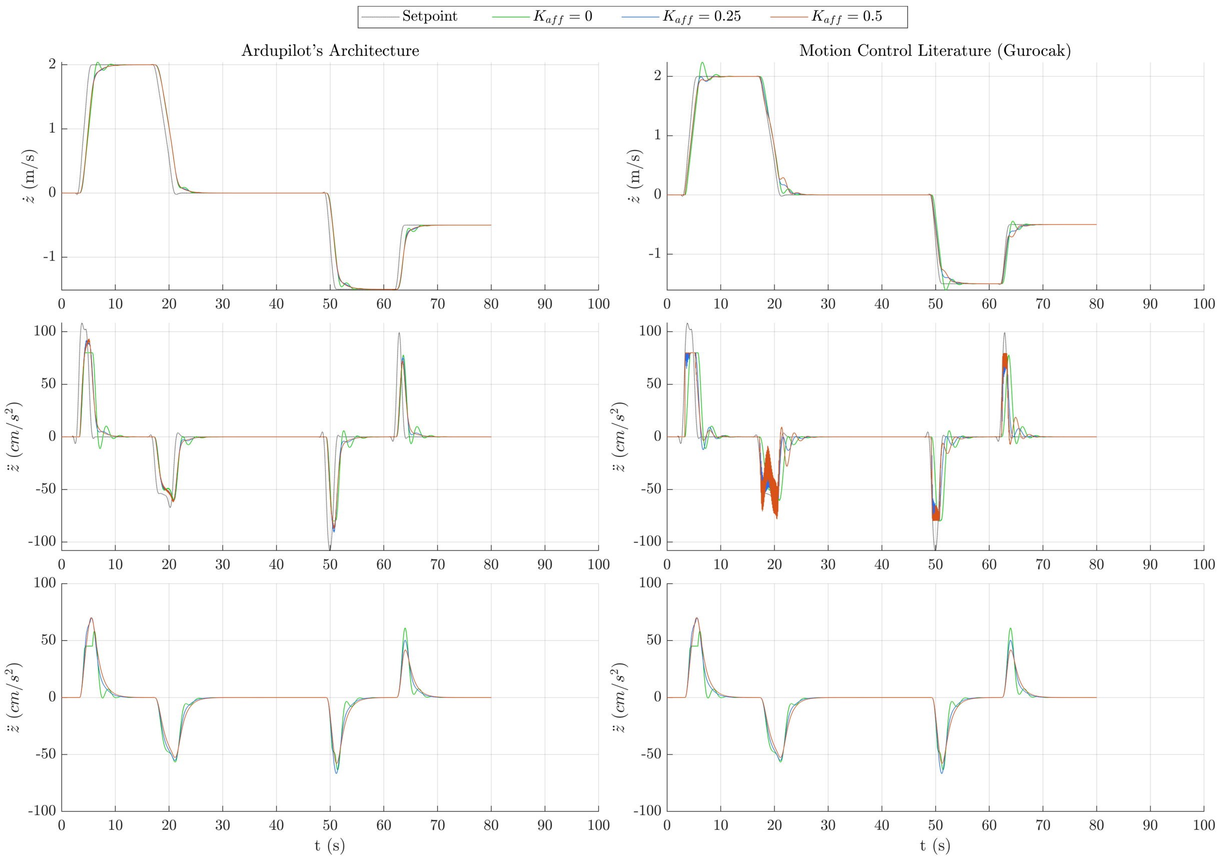
It can be seen that for either approach, adding the velocity feedforward action results in a reduction of tracking error for altitude. However, there are notable differences: the

Ardupilot's feedforward action only provides a minimal reduction in error. Meanwhile, the other approach reduces the tracking error to virtually zero.

It can be said that the cost of such a result from the Gurocak velocity feedforward is the overshoot caused in velocity. This value can reach 36.12% for $K_{vff} = 1$. In addition to this substantial velocity deviation, greater oscillation and a longer wind-up time are observed in the Gurocak architecture integrator. Meanwhile, in the Ardupilot, the velocity overshoot does not exceed 5.61%, and the oscillatory nature of the acceleration is less pronounced.

In the second simulation, it is proposed to fix the gain K_{vff} at 0.5, and vary the value of the acceleration feedforward gain. The next 6 scenarios are shown in Figure 5. For variations in K_{aff}, changes in position are imperceptible, and therefore the altitude graph has been omitted.

Figure 5 – Vertical axis outputs for different acceleration feed forward gains for both Ardupilot's and Gurocak's Architecture in which, (a) and (b) are position, (c) and (d) velocity, (e) and (f) acceleration, and (g) and (h) integral signal amplitude. K_{vff} is fixed to 0.5 in all cases



Source: The Author (2026).

For the acceleration feedforward, both strategies also show similar results. It is worth highlighting that Gurocak's architecture even corrects for the high PO = 11.93% generated by its speed feedforward.

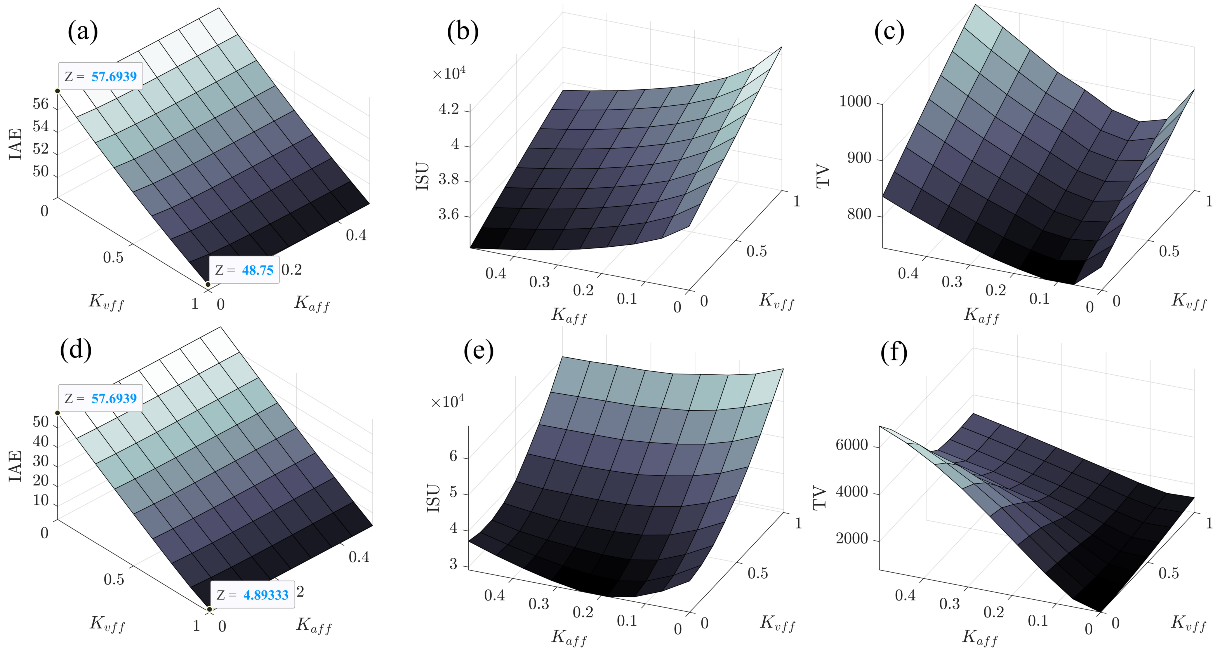
However, there is also a cost to increasing the gain in the acceleration feedforward. For high gains in this parameter, the acceleration exhibits aggressive oscillatory behavior. The thrust U_1 is the output of the acceleration control and, as observed in Equation 2, the rotation of all motors is dependent on this signal. Therefore, this oscillatory behavior substantially impacts the operation of the motors and electronic drivers.

Despite this limitation for practical applications, it is important to realize that the use of high feedforward gains actually reduces the dependence on the integral term of the PID, as proposed. In Figure 5, the integrators do not wind-up for any tested value of K_{aff} greater than zero.

These conclusions are confirmed even more thoroughly when one looks at the quantitative analysis. Figure 6 shows the relation between the feedforward gains K_{vff} and K_{aff} with the IAE, ISU and TV indices.

From Figure 6 (a) e (d), it is clear that while the K_{vff} reduces the altitude IAE, K_{aff} has almost zero effect on this index. It also shows that Ardupilot's architecture reduces the error approximately 10 times less than the traditional strategy.

Figure 6 – Visualization of the relationship between performance indices and feedforward gains where: (a), (b) and (c) are the indices for the Ardupilot system, and (d), (e) and (f) for the system presented by Gurocak.



Source: The Author (2026).

As can be seen from Figure 6 (b) and (e), the increment in K_{vff} causes the ISU to rise, and this increase is more pronounced in the motion control approach. Increasing the K_{aff} index can result in a slight reduction in the ISU. In the case of the Gurocak architecture, there is a point where this reduction is minimal before the index starts to rise again.

Finally, in Figure 6 (c) and (f), the final effect of increasing the feedforward gains for speed and acceleration is shown. Increasing the gains, especially K_{aff}, raises the TV hundreds of times in both cases. The curves show that increasing the gains in search of minimal error and energy consumption is pointless. The exorbitant TV values show that to achieve such conditions, the actuators would have to react at impractical speeds. Even when the combination of high gains is compatible with the actuator dynamics, it is important to consider the wear generated by such tuning.

The main cause of this oscillation in the final output for high K_{vff} and K_{aff} values is that the feedforward control begins to compete with the feedback control. During the ramps of the motion profile, the feedforward injects high signals directly into the downstream cascade controller. The PID of the next controller then reacts by generating aggressive signals in the opposite direction.

This undesirable competition between prediction and feedback loops cannot be matched by the actuator dynamics. As this work does not model such dynamics and practical tests were not performed, the possibility of degradation in system stability should also be considered.

6. CONCLUSION

The analysis shows that feedforward is effective for improving drone trajectory tracking performance. The work highlights significant differences in the strategies adopted for applying feedforward alongside feedback control. While the standard motion control strategy is more aggressive in reducing error, the alternative methodology adopted by Ardupilot serves more as a fine-tuning tool for PID controllers. Furthermore, the cost of adopting feedforward gains in speed and acceleration must be considered. It is observed that while this feature reduces IAE, it also increases costs. It can increase both energy consumption, as observed by ISU, and actuator demands, as observed by TV. Future studies, considering actuator dynamics and stability analysis, may show at what point feedforward could cease to mean only performance degradation and also lead to system instability.

7. REFERÊNCIAS

AHMDIA, A.; IBRAHIM, M.; SARHAN, A.; ELSAID, A. E., Comparative Analysis of Optimal Controllers Performance for Unmanned Aerial Vehicles. In: 2025 15th INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING (ICE-ENG), Cairo, Egypt., IEEE 2025. p. 1-6.

Ardupilot Development Team. **ArduPilot Copter – Copter Documentation,** 2024. Available: https://ardupilot.org/copter/index.html. Retrieved in 31 mar. 2026.

Ardupilot Development Team. **Ardupilot Development Site,** 2024. Available: https://ardupilot.org/dev/index.html. Retrieved in 31 mar. 2026.

BOUABDALLAH, S.; NOTH, A.; SIEGWART, J., PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor. In: 2004 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), Sendai, Japan: IEEE 2004. p. 2451-2456.

BOUABDALLAH, S., **Design and control of quadrotors with application to autonomous flying** 2007. Tese (Doutorado). Ecole Polytechnique Fédérale de Lausanne (EPFL) - Lausanne, 2007.

GUROCAK, H., **Industrial Motion Control: motor selection, drives, controller tuning, applications**. 1 ed. West Sussex: Wiley, 2016.

LI, L.; SUN, L.; JIN, J., Survey on Advances in Control Algorithms of Quadcopter Unmanned Aerial Vehicle. In: 2015 16th INTERNATIONAL CONFERENCE ON COMMUNICATION TECHNOLOGY (ICCT), Hangzhou, China: IEEE 2025. p. 107-111.

LI, J.; LI, L., Dynamic Analysis and PID Control for a Quadrotor. In: 2011 IEEE INTERNATIONAL CONFERENCE ON MECHATRONICS AND AUTOMATION, Beijing, China: IEEE 2011. p. 573-578.

LIMA, G., **Modelagem dinâmica e controle para navegação de um veículo aéreo não tripulado do tipo quadricóptero**. 2015. Dissertação (Mestrado). Pós-Graduação em Engenharia Elétrica, Universidade Federal de Uberlândia - Uberlândia, 2015.

LIMA, G., **Planejamento de trajetórias para quadricópteros em tarefas de perseguição.** 2007. Tese (Doutorado). Pós-Graduação em Engenharia Elétrica, Universidade Federal de Uberlândia - Uberlândia, 2019.

LIMA, G.; DE SOUZA, R.; DA SILVA, F.; DE MORAIS, A.; DE MORAIS, J., Análise Dinâmica e Controle de um Quadricóptero. In: XIII CONFERÊNCIA DE ESTUDOS EM ENGENHARIA ELÉTRICA (CEEL), Uberlândia, Brazil: 2015.

PINTO, M.; LIMA, G.; DE SOUSA, G.; DOS SANTOS, M.; SOUZA, R.; MORAIS, A., Parametrization and Experimental Control of a Mini Quadcopter. In: PROCEEDINGS OF THE 25th INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING, Uberlândia, Brazil: ABCM 2019.

MAHONY, R; KUMAR, V; CORKE, P., Multirotor Aerial Vehicles: Modeling, Estimation and Control of Quadrotor. **IEEE Robotics & Automation Magazine**, v. 19, p. 20-32, 2012. https://doi.org/10.1109/MRA.2012.2206474

NOTAS

CONTRIBUIÇÃO DE AUTORIA

Concepção e elaboração do manuscrito: V. F. Rodrigues, G. V. Lima
Conceitualização: V. F. Rodrigues
Desenvolvimento de Software: V. F. Rodrigues
Discussão dos resultados: V. F. Rodrigues
Revisão e aprovação: G. V. Lima

PREPRINTS

O manuscrito não é um preprint.

AGRADECIMENTOS

Agradecemos à comunidade de desenvolvimento de software aberto, em especial, o time responsável pelo Ardupilot, por disponibilizar código fonte, documentação e responder dúvidas nos fóruns do projeto.

CONFLITO DE INTERESSES

Os autores declaram não haver conflito de interesses.

DISPONIBILIDADE DE DADOS DE PESQUISA E OUTROS MATERIAIS

Os autores declaram ter interesse em disponibilizar os dados mas, não responderam mais ao corpo editorial sobre.

ANUÊNCIA DE AVALIAÇÃO ABERTA

Não aplicável.