drones

MDPI

Article

Sparse Online Gaussian Process Adaptive Control of Unmanned Aerial Vehicle with Slung Payload

Muhammed Rasit Kartal *, Dmitry I. Ignatyev and Argyrios Zolotas

Autonomous Systems and Control Centre for Autonomous and Cyberphysical Systems, School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Bedford MK43 0AL, UK; d.ignatyev@cranfield.ac.uk (D.I.I.); a.zolotas@cranfield.ac.uk (A.Z.)
* Correspondence: m.r.kartal@cranfield.ac.uk

Abstract: In the past decade, Unmanned Aerial Vehicles (UAVs) have garnered significant attention across diverse applications, including surveillance, cargo shipping, and agricultural spraying. Despite their widespread deployment, concerns about maintaining stability and safety, particularly when carrying payloads, persist. The development of such UAV platforms necessitates the implementation of robust control mechanisms to ensure stable and precise maneuvering capabilities. Numerous UAV operations require the integration of payloads, which introduces substantial stability challenges. Notably, operations involving unstable payloads such as liquid or slung payloads pose a considerable challenge in this regard, falling into the category of mismatched uncertain systems. This study focuses on establishing stability for slung payload-carrying systems. Our approach involves a combination of various algorithms: the incremental backstepping control algorithm (IBKS), integrator backstepping (IBS), Proportional–Integral–Derivative (PID), and the Sparse Online Gaussian Process (SOGP), a machine learning technique that identifies and mitigates disturbances. With a comparison of linear and nonlinear methodologies through different scenarios, an investigation for an effective solution has been performed. Implementation of the machine learning component, employing SOGP, effectively detects and counteracts disturbances. Insights are discussed within the remit of rejecting liquid sloshing disturbance.

Keywords: control; UAVs; incremental backstepping; slung payload; agricultural drone; Gaussian process; adaptive control; pendulum

check for updates

Citation: Kartal, M.R.; Ignatyev, D.I.; Zolotas, A. Sparse Online Gaussian Process Adaptive Control of Unmanned Aerial Vehicle with Slung Payload. *Drones* **2024**, *8*, 687. https://doi.org/10.3390/drones8110687

Academic Editors: Görres Grenzdörffer and Jian Chen

Received: 3 October 2024
Revised: 1 November 2024
Accepted: 11 November 2024
Published: 19 November 2024

1. Introduction

In the last decades, the rapid advancement of technology has skyrocketed Unmanned Aerial Vehicles (UAVs) into a pivotal role in various sectors [1], including the transportation of slung payloads emerging as a notable application domain. This application entails the carriage of payloads, such as pesticide tanks or other equipment, suspended externally from UAVs to designated locations [2]. However, the efficacy of such operations is significantly challenged by the instability inherent to slung payloads during flight.

Transporting slung payloads with UAVs presents a paradigm shift in logistics and delivery, enabling targeted and localized transportation of goods [3]. Yet, the inherent instability arising from the dynamics of the slung payloads during flight undermines the precision and reliability of such missions. The unstable movement of payloads suspended externally from these UAVs can lead to instability issues and compromised cargo delivery, ultimately impacting the overall effectiveness of the mission [4]. Ensuring consistent and controlled behavior of the slung payloads necessitates innovative control strategies that can adapt to the dynamic nature of the system [5]. To this end, the utilization of advanced control algorithms becomes essential to mitigate the adverse effects of payload-induced instability.

Through the target operation, proposed control methodology takes shape, such as the study in [6] considering the payload's vibration. However, it is a common approach to develop adaptive algorithms in this manner; [7,8]. Some researchers provide stability with the support of an artificial neural network (ANN) by training through the simulations [9]. Some researchers treated the payload movement as a disturbance and applied a disturbance observer for the solution [10]. Although every methodology has self-improvement through the described perspectives, most of the studies depend on model information, such as total weight. However a commercially confident system, which will go through different tasks, should not require pre-defined training or coefficients.

In this study, the critical issue of slung payload instability in UAV-based cargo transportation operations has been investigated. Our primary focus is on developing robust and model free control methodology to effectively manage the dynamics of slung payloads during flight. The overarching goal is to enhance the precision, efficiency, and reliability of payload delivery, thereby augmenting the overall effectiveness of cargo transportation with UAVs.

Payload and weight changes can significantly affect dynamic behavior of the drone/UAV. This is particularly true for drone platforms such as agricultural drones for spraying or drones for cargo delivery. Weight changes must be considered in the modeling to enable and enforce appropriate robust controller design for stability and performance.

With the increased nonlinearity and instability, control of drones with slung payload becomes an even tougher challenge. Nonlinear control approaches arise as promising solutions due to their capability to handle complex and non-standard systems [11]. Among these, the backstepping control methodology holds notable significance, offering a systematic way to design control laws by iteratively addressing the system's dynamics [12]. Furthermore, the incorporation of incremental control techniques demonstrates remarkable adaptability. Notably, the principles of Incremental Dynamics provides a framework to invert the dynamics of the system, while incremental backstepping control builds upon this by sequentially stabilizing intermediate systems [13].

Additionally, we consider the classic Proportional–Integral–Derivative (PID) control as a benchmark for comparison in our study. PID control is a widely used feedback control mechanism that adjusts the control input based on the error between the desired setpoint and the measured process variable. While PID control is simple to implement and often effective in many applications, it may lack the adaptability and robustness required to handle the complex dynamics of slung payloads in UAV-based cargo transportation operations. By including PID control in our comparative analysis, we aim to evaluate its performance against our proposed control methodologies, shedding light on the advantages and limitations of both traditional and advanced control techniques in this context.

In this paper, Sparse Online Gaussian Processes (SOGPs) are integrated into control strategies. SOGPs extend the capabilities of Gaussian Processes (GPs), catering to the demands of real-time control scenarios. They optimize computational efficiency and predictive precision through sequential projections and sparse representations. This framework aligns with the dynamic nature of control processes, making GP a potent choice for responsive, adaptive control systems. By leveraging SOGPs, we aim to enhance the stability and accuracy of our control methodologies in managing the dynamics of slung payloads during UAV-based cargo transportation operations.

Through a comprehensive analysis, the performance of the proposed control methodologies against traditional approaches is compared. To facilitate this evaluation, we implement a six-degree-of-freedom (6DoF) quadrotor model that accurately simulates the dynamics of cargo transportation missions. The study encompasses the formulation of distinct scenarios involving payload-induced disturbances, simulating real-world conditions.

The ensuing sections of this paper are structured as follows: Section 2 elaborates on the modeling aspects of the study; the UAV model and payload dynamics have been discussed. Section 3 details the proposed control strategies and Section 4 explains Sparse Online Gaussian Process methodology. Section 5 encompasses simulations, offering insights into the performance of the control methodologies under payload-induced disturbances. Finally, Section 6 concludes the study by summarizing the findings and presenting avenues for further research and implementation.

2. Model

2.1. UAV Model

In this paper, we consider a quadcopter UAV. Therefore, the dynamic model of the quadrotor for the inertial frame can be written as follows.

$$\ddot{x} = -\frac{f_t}{m}[\sin\phi \sin\psi + \cos\phi \cos\psi \sin\theta] \tag{1}$$

$$\ddot{y} = -\frac{f_t}{m}[\cos\phi \sin\psi \sin\theta - \cos\psi \sin\phi] \tag{2}$$

$$\ddot{z} = g - \frac{f_t}{m}[\cos\phi \cos\theta] \tag{3}$$

$$\ddot{\phi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{\tau_\phi}{I_x} \tag{4}$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{\tau_\theta}{I_y} \tag{5}$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\theta}\dot{\phi} + \frac{\tau_\psi}{I_z} \tag{6}$$

With the six dynamics equations, the linear accelerations are represented by $[\ddot{x}, \ddot{y}, \ddot{z}]^T$ and angular accelerations are represented by $[\ddot{\phi}, \ddot{\theta}, \ddot{\psi}]^T$, which affect the position and angular state of the aircraft, respectively. However, dynamics defined by force, momentum, and inertia terms is shown as follows: f_t is the total force generated by propellers, $[\tau_\phi, \tau_\theta, \tau_\psi]^T$ are the orientation torques produced by the relation of motors, and $[I_x, I_y, I_z]^T$ is inertia, which is the UAV's body's coefficient for tuning through each $X - Y - Z$ axis.

The following are the assumptions made for the design; the structure is rigid and symmetrical, the propellers are rigid, and the thrust and drag are proportional to the square of the propeller speed.

It is important to note that the quadrotor modeling purpose is mainly for the control design. However, the model which has been used for simulations has been calculated with payload. The next section will detail the proposed payload modeling.

2.2. Slung Payload Model

An accurate model with payload is essential for simulation studies. Through the measurements from simulations, controller methodology validations will be made. For unstable payload modeling, many different approaches have been studied in the literature.

In the literature, various payload modeling methodologies have been developed. Some of them are mentioned as follows: single pendulum [14–16], single pendulum dynamics by Navier–Stokes [17,18], two pendulum [19], the multi-mass spring model [20], the finite element method algorithm [21], Lagrange formalism [22,23], smoothed particle hydrodynamics(SPH) [24], the multi-body model [25], the time-independent finite difference method [26], and the infinite-dimensional model computed with a finite-dimension approximation [27,28].

After analysis of the literature, it has been found that the widely used and most accurate method is pendulum dynamics-based modeling. However, in some studies, liquid payload movement converged with a similar approach. For this study, the payload model borrowed from Guerrero's work [29].

The base approach to the state is taken as coupling two different systems. First, the pendulum system's dynamics were modeled, and then the effects of the UAV and pendulum on each other were calculated. As shown in Figure 1, the pendulum's coordinates can be calculated regarding the UAV system given below:

$$x_p = x + l\cos(\beta)\sin(\alpha) \tag{7}$$
$$y_p = y + l\sin\beta\sin\alpha \tag{8}$$
$$z_p = z - l\cos\alpha \tag{9}$$

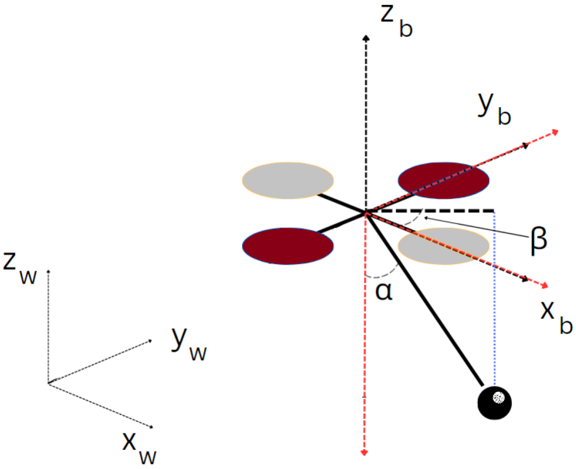

Figure 1. Pendulum and UAV frames.

With the following notation: ℓ is the pendulum cable length, α is the angle in the X_b0Z_b direction, β is the angle in the Y_b0X_b direction and $[x_b, Y_b, Z_b]$ body axes. The rotational movement of the pendulum mass has been neglected. The total potential and kinetic energy are given below.

$$\begin{aligned} K(q,\dot{q}) &= \frac{1}{2}\dot{\xi}M\dot{\xi}^T + \frac{1}{2}\dot{\eta}J\dot{\eta}^T + \frac{1}{2}\dot{\xi}_p m\dot{\xi}_p^T \\ U(q) &= Mgz + mg(z - l\cos\alpha) \end{aligned} \tag{10}$$

The notations in the Equation (10) are explained as follows: M is UAV mass, m is pendulum mass, J is UAV inertia $J = [I_{xx}, I_{yy}, I_{zz}]$, ξ is UAV position $\xi = [x, y, z]^T \in \mathbb{R}^3$, η is angular states $\eta = [\phi, \theta, \psi]^T \in \mathbb{R}^3$, and ξ_p is pendulum position $\xi_p = [x_p, y_p, z_p]$. The pendulum state derivative can be calculated through the UAV's state derivative and pendulum extension derivative.

$$\dot{\xi}_p = \dot{\xi} + \ell\begin{bmatrix} -\dot{\beta}\sin\beta\sin\alpha + \cos\beta\dot{\alpha}\cos\alpha \\ \dot{\beta}\cos\beta \\ \dot{\alpha}\sin\alpha\cos\beta + \cos\alpha\dot{\beta}\sin\beta \end{bmatrix} \tag{11}$$

The Lagrangian of the system is as follows:

$$L(q,\dot{q}) = K(q,\dot{q}) - U(q) \tag{12}$$

If the Equation (12) is described in a detailed manner;

$$\begin{aligned} L =& \frac{1}{2}(M+m)(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2}I_\psi\dot{\phi}^2 + \frac{1}{2}(I_\theta c_\phi^2 + I_\phi s_\phi^2)\dot{\theta}^2 \\ &+ \frac{1}{2}(I_\psi s_\theta^2 + I_\theta c_\theta^2 s_\phi^2 + I_\phi c_\theta^2 c_\phi^2)\dot{\psi} + (I_\theta c_\theta c_\phi s_\phi - I_\phi c_\theta c_\phi s_\phi)\dot{\psi}^2\dot{\theta}^2 + I_\psi s_\theta\dot{\psi}^2\dot{\phi}^2 \\ &+ m\dot{x}\ell(c_\beta c_\alpha\dot{\alpha} - s_\beta s_\alpha\dot{\beta}) + m\dot{y}\ell(c_\beta\dot{\beta}) + m\dot{z}\ell(c_\beta s_\alpha\dot{\alpha} + s_\beta c_\alpha\dot{\beta}) + \frac{1}{2}m\ell^2(c_\beta^2\dot{\alpha}^2 + \dot{\beta}^2) \\ &+ \frac{1}{2}I(\dot{\alpha}^2 + \dot{\beta}^2) - Mgz - mg(z - \ell c_\alpha c_\beta) \end{aligned} \tag{13}$$

Now, we can write the Euler–Lagrange equation for the system:

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}}\right] - \frac{\partial L}{\partial q} = bu \tag{14}$$

where the input signal $u = [f_t, \tau_\psi, \tau_\theta, \tau_\phi]^T$, state vector $q = [x, y, z, \psi, \theta, \phi, \alpha, \beta]^T$, and

$$b = \begin{bmatrix} s_\phi s_\psi + c_\phi c_\psi s_\theta & 0 & 0 & 0 \\ c_\phi s_\theta s_\psi - c_\psi s_\phi & 0 & 0 & 0 \\ c_\theta c_\phi & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{15}$$

Equation (14) can be represented in matrix form:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = bu \tag{16}$$

M matrix with the following coefficients:

$$M(q) = \begin{bmatrix} m_{11} & 0 & 0 & 0 & 0 & 0 & m_{17} & m_{18} \\ 0 & m_{22} & 0 & 0 & 0 & 0 & m_{27} & m_{28} \\ 0 & 0 & m_{33} & 0 & 0 & 0 & m_{37} & 0 \\ 0 & 0 & 0 & m_{44} & m_{45} & -I_\psi s_\theta & 0 & 0 \\ 0 & 0 & 0 & m_{54} & m_{55} & 0 & 0 & 0 \\ 0 & 0 & 0 & -I_\psi s_\theta & 0 & I_\psi & 0 & 0 \\ m_{71} & m_{72} & m_{73} & 0 & 0 & 0 & m_{77} & 0 \\ m_{81} & m_{82} & 0 & 0 & 0 & 0 & 0 & m_{88} \end{bmatrix} \tag{17}$$

The notations in the corresponding matrix equations are described as $m_{11} = m_{22} = m_{33} = M + m$, $m_{17} = m_{71} = m\ell c_\alpha c_\beta$, $m_{18} = m_{81} = -m\ell s_\alpha s_\beta$, $m_{27} = m_{72} = m\ell c_\alpha s_\beta$, $m_{28} = m_{82} = m\ell s_\alpha c_\beta$, $m_{37} = m_{73} = m\ell s_\alpha$, $m_{44} = I_\psi s_\theta^2 + c_\theta^2(I_\theta s_\theta^2 + I_\theta s_\theta^2)$, $m_{45} = m_{54} = (I_\theta - I\phi)(c_\theta s_\phi c_\phi)$, $m_{55} = I_\theta c_\phi^2 + I_\phi s_\phi^2$, $m_{77} = m\ell^2 + I$, $m_{88} = m\ell^2 s_\alpha^2 + I$.

Coriolis matrix (C) coefficients:

$$C(q, \dot{q}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & c_{17} & c_{18} \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{27} & c_{28} \\ 0 & 0 & 0 & 0 & 0 & 0 & c & 0 \\ 0 & 0 & 0 & c_{44} & c_{45} & c_{46} & 0 & 0 \\ 0 & 0 & 0 & c_{54} & c_{55} & c_{56} & 0 & 0 \\ 0 & 0 & 0 & c_{64} & c_{65} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -m\ell^2 s_\alpha c_\alpha \dot{\beta} \\ 0 & 0 & 0 & 0 & 0 & 0 & m\ell^2 s_\alpha c_\alpha \dot{\beta} & m\ell^2 s_\alpha c_\alpha \dot{\alpha} \end{bmatrix} \tag{18}$$

The notations in the corresponding Coriolis matrix equations are described as
$c_{17} = -m\ell(c_\alpha s_\beta \dot{\beta} + s_\alpha c_\beta \dot{\alpha})$, $c_{18} = -m\ell(c_\alpha s_\beta \dot{\alpha} + s_\alpha c_\beta \dot{\beta})$, $c_{27} = m\ell(c_\alpha c_\beta \dot{\beta} - s_\alpha s_\beta \dot{\alpha})$,
$c_{28} = m\ell(c_\alpha c_\beta \dot{\alpha} - s_\alpha s_\beta \dot{\beta})$, $c_{44} = I_\psi \dot{\theta} s_\theta c_\theta - (I_\theta + I_\phi)(\dot{\theta} s_\theta c_\theta s_\phi^2) + (I_\theta - I_\phi)\dot{\phi} c_\theta^2 s_\phi c_\phi$,
$c_{45} = I_\psi \dot{\psi} s_\theta c_\theta - (I_\theta - I_\phi)(\dot{\theta} s_\theta c_\phi s_\phi + \dot{\phi} c_\theta s_\phi^2) - (I_\theta + I_\phi)\dot{\phi} c_\theta s_\theta c_\phi^2 - \dot{\phi} c_\theta c_\phi^2)$,
$c_{46} = -(I_\psi \dot{\theta} c_\theta - (I_\theta - I_\phi)\dot{\psi} c_\theta^2 s_\phi c_\phi)$, $c_{54} = \dot{\psi} c_\theta s_\theta(-I_\psi + I_\theta s_\phi^2 + I_\phi c_\phi^2)$,
$c_{55} = -(I_\theta - I_\phi)\dot{\phi} s_\phi c_\phi)$, $c_{56} = I_\psi \dot{\psi} c_\theta + (I_\theta - I_\phi)(-\dot{\theta} s_\theta c_\phi + \dot{\psi} c_\theta c_\phi^2) - \dot{\psi} c_\theta c_\phi^2 - \dot{\phi} c_\theta s_\phi^2)$,
$c_{64} = -(I_\theta - I_\phi)\dot{\psi} c_\theta^2 s_\psi c_\psi)$, $c_{65} = -I_\psi \dot{\psi} c_\theta + (I_\theta - I_\phi)(\dot{\theta} s_\phi c_\phi + \dot{\psi} c_\theta s_\phi^2) - \dot{\psi} c_\theta c_\phi^2)$

Finally, gravity-related coefficients are given below:

$$G(q) = \begin{bmatrix} 0 & 0 & (M+m)g & 0 & 0 & 0 & mlgs_{\alpha} & 0 \end{bmatrix}^T \tag{19}$$

In order to calculate dynamics by the Euler–Lagrange method, the three-dimensional movement of the pendulum has been modeled and applied to the quadrotor system. Another unique aspect, according to the target mission, is that weight change has been included. Since the weight ratio of the payload was quite high, weight change had an important role. It is expected that a payload-delivering UAV should show acceptable performance with and without the payload. According to this aspect, due to model-changing nature of the problem, a model-free approach is needed.

3. Control

In the present study, the controller algorithm is structured in a cascaded framework. The outer control loop governs the UAV's position control, calculating the set points for angular orientation. Conversely, the inner loop orchestrates the angular control, directly influencing the actuation of the UAV's motors. Also, in order to avoid windup, a filter algorithm is used. Throughout the agile dynamics of rotorcraft UAVs, the adoption of a cascaded control strategy for this inherently underactuated system proves to be an effective approach. The visual representation of this cascade system design is depicted in Figure 2.

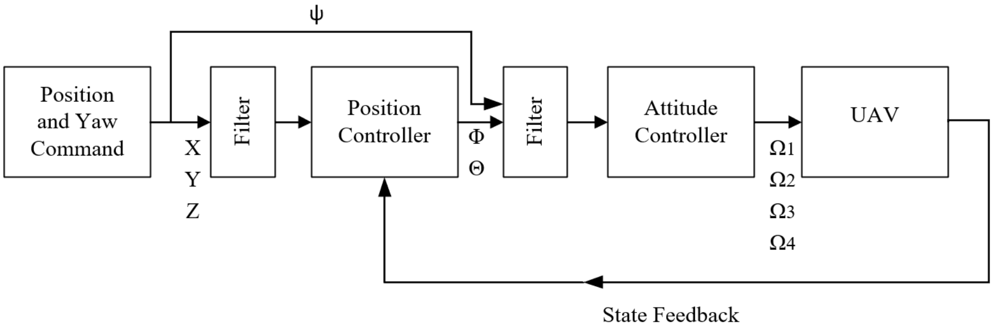

Figure 2. Proposed cascade control system diagram.

As shown in the previous section, pendulum movement affects force disturbance. Thus, the results of the disturbance show themselves on the position states. To find the best approach, linear and nonlinear control methodologies were applied and compared. One of these controllers employs a Proportional–Integral–Derivative (PID) approach, while the other controller is the integrator backstepping (IBS) approach, and last is the incremental backstepping (IBKS) controller.

For the attitude controller, the incremental backstepping (IBKS) controller has been used. Advantages of the attitude control with incremental backstepping have been discussed in a previous study [30,31]. Also, the incremental backstepping controller has been applied for position controlling.

These proposed algorithms affirmed their attributes in terms of finite-time convergence and overall stability through the Lyapunov theory. In the following discussion of the topic, the proposed controlling algorithms will be explained briefly.

3.1. *Proportional–Integral–Derivative*

The PID (Proportional–Integral–Derivative) control algorithm is widely recognized. It integrates proportional, integral, and derivative components. In our system, we implemented the PID controller in both PID and P-PID forms, with a parallel PID controller configuration shown in Figure 3.

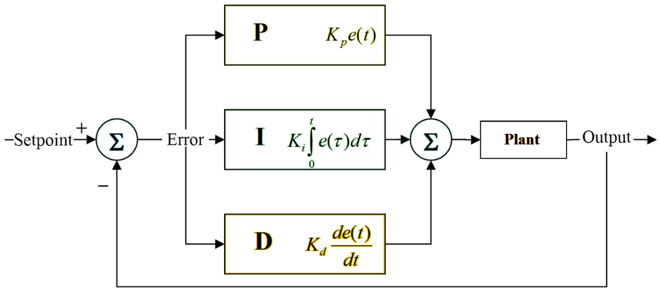

Figure 3. Proportional–Integral–Derivative(PID) controller diagram.

For position control, the PID controller is used. It can be positioned in the cascade design as an outer or higher controller, placed in the final layer of the structure. Thus, the controller structures are designed as PID-IBKS. For the X and Y positions, the Proportional–Derivative (PD) form was selected, while for the Z position, the Proportional–Integrall–Derivative (P-PID) form was chosen. The fundamental logic of the PID is detailed in Equation (20).

$$e_x = x_d - x \tag{20}$$

$$u(t) = K_{p_x} e_x + K_{i_x} \int e_x + K_{d_x} \dot{e}_x \tag{21}$$

The PID parameters are refined using an iterative optimization process that minimizes a performance index J. This index typically incorporates both time-domain specifications (e.g., rise time t_r, settling time t_s, overshoot M_p) and frequency-domain specifications (e.g., gain margin GM, phase margin PM):

$$J = w_1 t_r + w_2 t_s + w_3 M_p + w_4 |GM - GM_{\text{desired}}| + w_5 |PM - PM_{\text{desired}}| \tag{22}$$

The coefficients defined in w_i notation are weighting factors. In Table 1, the coefficients of PID controllers are given.

Table 1. PID controller coefficient table.

	P	P	I	D
X&Y	—	−2.4098	0.02374	1.11357
Z	1.61057	−3.11757	−0.4174	0

3.2. Integrator Backstepping

Integrator backstepping is a control strategy used for nonlinear systems building upon the concept of backstepping control. It involves systematically introducing integrators into the system dynamics at each step of the control design process. These integrators help in addressing steady-state errors and enhancing tracking performance.

The control law is designed gradually, with each step involving the addition of a new control term to the previous one, effectively “backstepping” through the system dynamics. In this way, translational dynamics are regulated with both position and speed state errors of the UAV. This approach allows for the creation of controllers capable of stabilizing and tracking desired trajectories for a wide range of nonlinear systems.

One of the key advantages of integrator backstepping is its robustness, as it can handle uncertainties and disturbances effectively [32]. Moreover, it offers stability guarantees and can accommodate complex system dynamics.

Proposed system equation:

$$\dot{x} = f(x) + g(x)\xi \tag{23}$$

$$\dot{\xi} = u \tag{24}$$

The integrator backstepping method proposes a virtual input ξ after putting the actual input u to an integrating function. In this way, we would be able to make modifications on the input signal. To achieve this, the desired virtual input is also defined by state feedback control:

$$\xi_{des} = \alpha(x) \Longrightarrow \dot{x} = f(x) + g(x)\alpha(x) \tag{25}$$

V is the proposed solution equation and V_a is the proposed positivity definite function

$$V(x) = \frac{1}{2}x^2 \Longrightarrow \dot{V}(x) = \dot{x}x \leq Va(x) \tag{26}$$

After the proposed process, the input signal definition according to control solution is given as below:

$$z = \xi - \alpha(x) \tag{27}$$

$$\dot{z} = u - \dot{\alpha} = \left(\frac{\partial V}{\partial x}\right)g(x) - cz \tag{28}$$

$$u = \dot{\alpha} + \left(\frac{\partial V}{\partial x}\right)g(x) - c[\xi - \alpha(x)] \tag{29}$$

In Figure 4, the placement of the desired input in the system diagram is given.

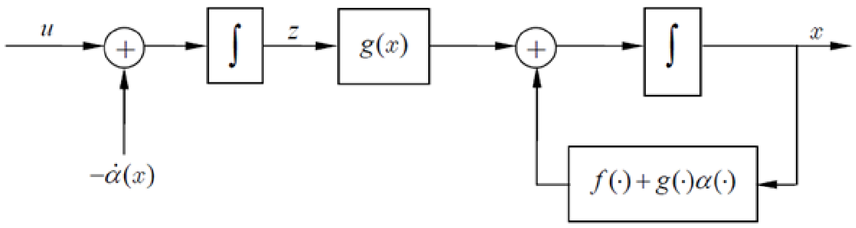

Figure 4. Integrator backstepping diagram.

With the defined backstepping dynamics, the relation between acceleration $\ddot{p}$, position error ϵ_p, and velocity error ϵ_s is given below:

$$\ddot{p} = \epsilon_p - k_p(k_p\epsilon_p + \epsilon_s) - k_s\epsilon_s \tag{30}$$

The error and acceleration variables are described as $\ddot{p} = [\ddot{p}_x, \ddot{p}_y, \ddot{p}_z]^T$, $\epsilon_p = [\epsilon_X, \epsilon_Y, \epsilon_Z]^T$, $\epsilon_s = [\epsilon_u, \epsilon_v, \epsilon_w]^T$, and k_p, k_s represents position and speed error gains. The simplified position dynamics is given below:

$$\begin{pmatrix} \ddot{p}_x \\ \ddot{p}_y \\ \ddot{p}_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} + \begin{pmatrix} -\cos\phi\sin\theta \\ \sin\phi \\ -\cos\phi\cos\theta \end{pmatrix} \frac{f_t}{m} \tag{31}$$

With the help of simplified dynamics for control, it has become possible to define the desired angle values for the defined speed.

$$u_\phi = \arcsin\left(\frac{m}{f_t}(\epsilon_Y - k_Y(k_Y\epsilon_Y + \epsilon_v) - k_v\epsilon_s)\right) \tag{32}$$

$$u_\theta = \arcsin\left(-\frac{m}{f_t(\cos\phi}\epsilon_X - k_X(k_X\epsilon_X + \epsilon_u) - k_u\epsilon_s)\right) \tag{33}$$

$$u_{f_t} = \frac{m}{\cos\phi\cos\theta}(\epsilon_Z - k_Z(k_Z\epsilon_Z + \epsilon_Z) - k_w\epsilon_w + g) \tag{34}$$

3.3. Incremental Backstepping

The incremental backstepping algorithm steps forward with its lower model dependency and effective feedback management. The improved backstepping structure has proven itself in previous studies [31]. In terms of enhancing the backstepping controller's dependency on the model, sensor-based improvements were achieved using an incremental approach. Incremental dynamics, a promising sensor-based technique, has been developed to address unmodeled system dynamics by utilizing acceleration feedback. Within this framework, two distinct control algorithms were formulated: the incremental dynamic inversion (INDI) and incremental backstepping (IBKS) algorithms [33]. The backstepping controller's advantage includes regulating rotational movement with angular state and angular rate adjustments.

However, it is essential to note that the IBKS algorithm, while beneficial, occupies a middle ground between model-based and sensor-based methodologies. Consequently, it is more vulnerable to sensor-originated disturbances such as noise, bias, and delays within the closed-loop system [34]. The block diagram illustrating the incremental backstepping approach is shown in Figure 5. The sensor based IBKS algorithm needs a sensor to acquire actuator's actuator's value. Than the actual actuator value can be reversed and used as feedback to the control algorithm.

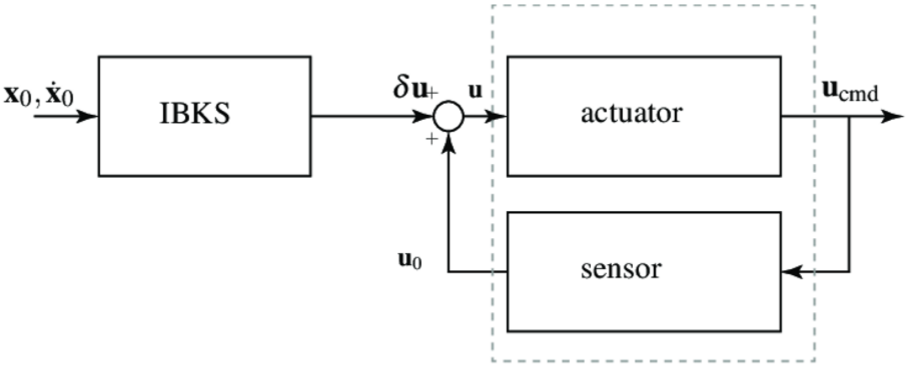

Figure 5. Incremental backstepping methodology diagram.

The key difference in the incremental approach is the combination of the controller signal Δu and the actual actuator state signal u_0, resulting in an enhanced responsive structure. As depicted in the diagrams, system feedback is of significant importance. The following section provides a mathematical explanation of the incremental backstepping algorithm.

The proposed system equation with incremental dynamics is as follows:

$$\dot{\xi} = f_k(\xi) + g_k(\xi)x \tag{35}$$

$$\dot{x} = f_d(\xi, x) + g_d(\xi, x)u \tag{36}$$

We design a backstepping controller with a control Lyapunov function (CLF) such as $V_1 = \frac{1}{2}\xi^2$ and $x_d = \alpha$

$$\dot{V_1} = \xi^T(f_k(\xi) + g_k(\xi)\alpha) \tag{37}$$

and

$$\alpha = -g_k{}^{-1}(\xi)(k_1\xi + f_k(\xi)) \tag{38}$$

α represents the desired value for the lower state. In the angular controller, this means angular rate. Repeating the same backstepping process with the included lower state, the double-layered backstepping controller is achieved. The incremental approach proposes the input signal calculated as given below:

$$u = \Delta u + u_0 \tag{39}$$

In the proposed method, u_0 is the actual input signal. And the incremental input signal and Δu becomes:

$$\Delta u = -{g_d}^{-1}(\xi, x_0)(\dot{x}_0 - \dot{\alpha} + kr + (\xi^T g_k(\xi))^T) \tag{40}$$

The final equations of input signals for the moment control become:

$$\Delta u_{\tau_\phi} = -\frac{I_x}{\ell}(\dot{p}_0 - \dot{p}_{ref} + k_p z_p + z_\phi) \tag{41}$$
$$\Delta u_{\tau_\theta} = -\frac{I_y}{\ell}(\dot{q}_0 - \dot{q}_{ref} + k_q z_q + \cos\phi z_\theta) \tag{42}$$
$$\Delta u_{\tau_\psi} = -I_z(\dot{r}_0 - \dot{r}_{ref} + k_r z_r + \frac{\cos\phi}{\cos\theta} z_\psi) \tag{43}$$

The coefficients and errors in the equations are defined as follows: $[z_\phi, z_\theta, z_\psi]$ represents angular state error, $[z_p, z_q, z_r]$ represents angular acceleration state error, and $[k_p, k_q, k_r]$ represents the gains. Also, the equations for the position control are given below:

$$\Delta u_{\phi_d} = -\frac{m}{f_t}(\dot{v}_0 - \dot{v}_{ref} + k_v z_v + \cos\phi z_y) \tag{44}$$
$$\Delta u_{\theta_d} = -\frac{m}{f_t \cos\phi}(\dot{u}_0 - \dot{u}_{ref} + k_u z_u - \cos\theta z_x) \tag{45}$$
$$\Delta u_{f_t} = \frac{m}{\cos\phi\cos\theta}(\dot{w}_0 - \dot{w}_{ref} + k_w z_w + \cos\phi\cos\theta z_z) \tag{46}$$

where $[z_x, z_y, z_z]$ represents position state error, $[z_u, z_v, z_w]$ represents speed state error, and $[k_u, k_v, k_w]$ represents the gains. The control Equation (40) indicates that the system's incremental dynamics are generated by the control input increment. Implementing this concept requires the assumption of a small sampling time. For position controlling, the actual state of the actuator has switched with the angular state of the UAV. For example, pitch angle has been chosen as the actuator value for the X position control. In this way, the cascade influence mechanism of the incremental methodology is preserved.

Furthermore, it is presumed that the actuators are highly responsive, allowing them to achieve the desired input increment within the brief sampling time. The time delay between the control command and feedback signal for u_0 has been chosen as 0.01 s. Additionally, the sensors are assumed to be ideal, delivering state derivatives without errors.

3.4. Filter

To prevent the controller from issuing infeasible commands, a command filter (CF) is integrated into the controller [35]. This filter incorporates bandwidth, magnitude, and rate limiters, as shown in Figure 6.

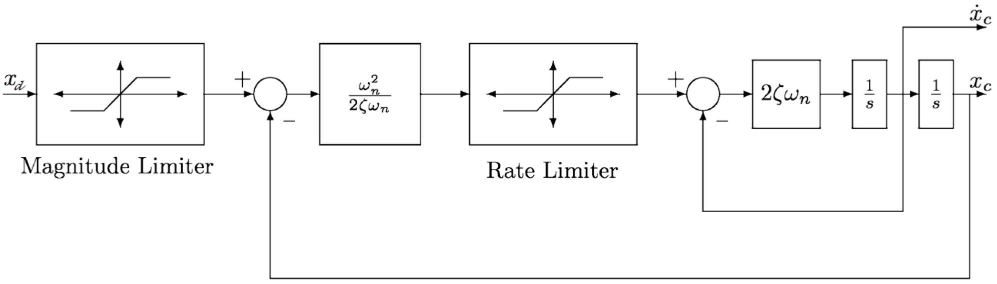

Figure 6. Anti-windup command filter diagram.

Command filters are low-pass filters that shape command inputs to align with the aircraft's dynamics. This technique, used in backstepping strategies, restricts the pseudo-control at each step. Consequently, differentiation variables must be estimated based on measurements. To estimate signal derivatives, the proposed filter also includes a differentiator.

$$\begin{bmatrix} x_c \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{47}$$

$$\begin{bmatrix} x_c \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} q_1 \\ 2\xi\omega_n \left(S_R \{ \frac{{\omega_n}^2}{2\xi\omega_n} [S_M(x_d) - q_1] \} - q_2 \right) \end{bmatrix} \tag{48}$$

The term S_M represents magnitude constraints and S_R indicates rate constraints.

4. Sparse Online Gaussian Process

Gaussian Processes (GPs) serve as potent tools for modeling continuous functions within control applications. They extend multivariate Gaussian distributions to infinite dimensions, functioning as distributions over functions. In control contexts, GP finds application in approximating intricate functions, such as uncertain disturbances in aircraft controls.

Mathematically, GP involves a collection of continuous random variables, where any finite subset follows a joint Gaussian distribution. The covariance function, dictated by a positive-definite kernel $(K_0(x, x') = Cov(\zeta, \zeta'))$, encapsulates the resemblance between inputs and guides predictions [36]. Kernel functions encapsulate crucial aspects, like the smoothness of functions, impacting predictive outcomes.

GPs exhibit remarkable adaptability and proficiency in managing stochasticity. These attributes are particularly relevant to control applications demanding real-time adjustments. GPs update their kernel dictionaries based on incoming data, ensuring relevance in evolving systems. In control algorithms, GPs bolster stability by compensating for uncertainties.

Sparse Online Gaussian Processes (SOGPs) extend GP capabilities, catering to the demands of real-time control scenarios. SOGPs optimize computational efficiency and predictive precision through sequential projections and sparse representations [37]. This framework aligns with the dynamic nature of control processes, making GP a potent choice for responsive, adaptive control systems.

The estimation of the posterior Gaussian Process approximation, along with its posterior means and covariance, involves utilizing the initial kernel $K_0(x, x')$ and the corresponding likelihoods.

$$\langle f_x \rangle_t = \alpha_t^T k_x \tag{49}$$

$$K_t(x, x') = K_0(x, x') + k_x^t C_t k_x, \tag{50}$$

Variables in the equations explained are as follows: $k_x = [K_0(x_1, x), \ldots\ldots, K_0(x_t, x)]$ is the kernel functions, $\alpha_t = [\alpha_t(1), \ldots\ldots., \alpha_t(t)]^T$ is the coefficient, and $C_t = C_t(ij)_{i,j=1.....t}$ is the coefficient matrix For dealing with regression-based problems, in the literature, Radial Basis Functions (RBFs) are widely used for kernel functions.

$$K(x, x') = exp(\frac{\left| x - x' \right|^2}{2\sigma_x^2}) \tag{51}$$

In the RBF equation, x and x' are input data points and σ represents the hyperparameter bandwidth. With the proposed framework, control efficiency is approximated using GP, characterized by its mean $\langle f_x \rangle$ and covariance $K(x, x')$. For a smooth identification process, the components of the input vector are normalized. The output observations are the instantaneous control efficiency values ζ, estimated from Equation (52) by dividing the increment of the state derivative $\Delta\dot{x}$ by the increment of the control input Δu:

$$\zeta_i = \frac{\Delta\dot{x}}{\Delta u} \tag{52}$$

To avoid singularity of the estimation due to division, an additional check of the input data is added:

$$\Delta u > \in_{u_{tol}} \tag{53}$$

Later, the achieved error measurement is added to the controller algorithm's signal as a correction. As a benefit of the model-free approach, the SOGP algorithm does not target a specific disturbance. Regardless of the source, the algorithm can neutralize any force-breaking disturbance. The problem source can be a slung payload, liquid tank, or an external issue like wind disturbance. As shown in Figure 7, the Gaussian Process output signal affects the angle command, which is calculated by position control.The disturbance caused by payload shown with the red arrow on the right side of diagram and correction signal calculated by SOGP algorithm shown with green arrow on the left side.

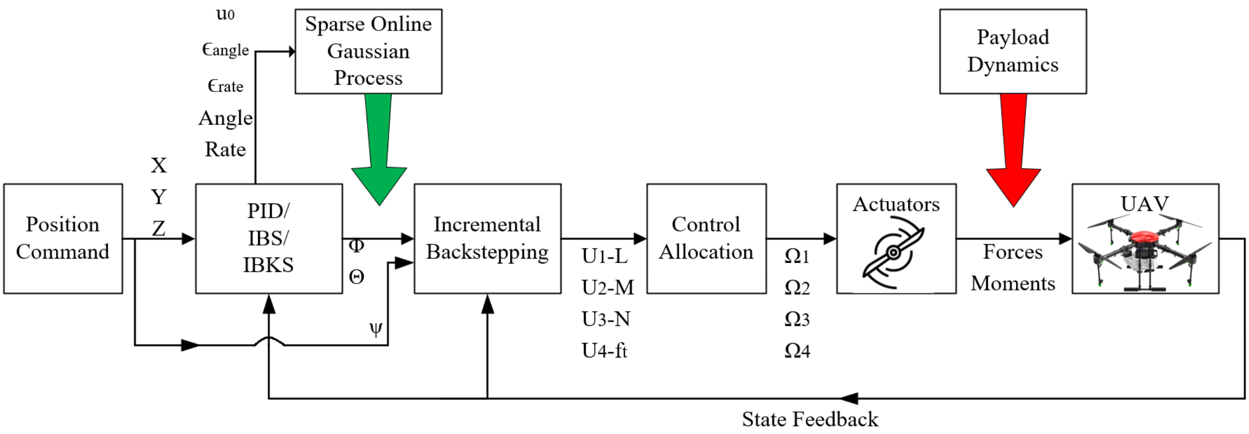

Figure 7. Proposed cascade control system diagram.

Final components of the proposed signal can be represented as below:

$$u = u_0 + y_{GP} \tag{54}$$

In this proposed methodology, u_0 represents desired angle and thrust inputs of the controllers and y_{GP} represents the SOGP algorithm signal [38]. The Gaussian Process input is notated with y_{GP} and y_{GPX}, and y_{GPY} and y_{GPZ} are the components through the axis.

$$y_{GP} = \begin{bmatrix} y_{GPX} \\ y_{GPY} \\ y_{GPZ} \end{bmatrix} \tag{55}$$

In order to achieve the most accurate values for y_{GP}, choosing correct state variables has significant importance. Variable selection for the proposed error dynamics is chosen as:

$$u_{SOGP} = [u_0, e_{angle}, e_{drate}, x_\xi, x_r]^T \tag{56}$$

Five observing state data have been chosen as follows: actual input value u_0, angular state error e_{angle}, derivative of rate state error e_{drate}, actual angular state x_ξ, and actual rate state x_r.

The GP algorithm has been designed in a model-free disturbance observer manner. By detecting unwanted changes in the angular velocity vector, or the widely used term angular body rates, it is possible to take action. From the observed status, a moment correction, the y_{GP} value, has been calculated.

Unfortunately, the performance of the proposed algorithm has been limited by the user's understanding of the state of the art. Although the proposed methodology has a clear solution to the problem, it still needs a hyper-parameter definition. According to the given parameters and gains to these parameters, the performance of the algorithm is about to change.

The validation of the proposed methodology has been tested with the regular Proportional–Integrator–Derivative (PID), integrator backstepping (IBS) and incremental backstepping control algorithms versus the Sparse Online Gaussian Process (SOGP)-implemented version of the algorithms. In this way, possible improvements and behavior analysis of the proposed methodology can be observed clearly. The results are discussed in the following section.

5. Simulation

This study explores fault recovery using the Gaussian Process in a disturbance observation manner. With a sparse online structure, the Gaussian algorithm is able to detect and reject disturbances. As explained in the slung payload model section, the movement of the slung payload affects external force to the UAV system. As a result of this, the disturbance outcome can be seen in the direct position state of the drone. Thus, the study has focused improvement on the position controlling methodology of the UAV. Then in order to achieve a deeper analysis, the proposed improvement applied two different control algorithms.

To make a comprehensive analysis, three control algorithms, PID, IBS and IBKS, have been designed and implemented. It should be noted that control algorithms are tuned according to UAVs without payload. This aspect is modeled considering operations with payload drop-off or payload weight-changing missions. The simulations have included both base control algorithms and Sparse Online Gaussian Process (SOGP)-implemented versions.

The quadrotor UAV frame model has been used with 27 kg total weight, which includes a 10 kg payload with 0.5 m rigid cable. The payload is a point mass, and considering the connecting cable between the UAV and the payload mass as zero, it is attached precisely at the UAV's center of gravity.

Additionally, the study excludes the modeling of other UAVs' pre-failure conditions and assumes they do not interfere with the single-UAV payload system under consideration. The cable is assumed to always be tight, ensuring the slung payload condition holds true, given that the UAV is not expected to execute abrupt manoeuvres causing cable tension to drop to zero. The study is set in an indoor environment, negating outdoor disturbances such as birds, wind gusts, and rotor down-wash and offering better control over disruptive variables like sudden air drafts or unexpected obstacles. Finally, during training and testing simulations, it is presumed that ideal sensor data are available without noise, and odometry is consistently accessible for use by the controllers, with no dropout. These assumptions collectively simplify the study's scope, although they may not encompass the full complexities of real-world scenarios.

Simulation studies were completed in three different approaches. Firstly, validation of all six proposed algorithms has been performed. Then, controller performance against the slung payload was analyzed. Finally, a case study that satisfies many aspects of a real-world scenario has been simulated.

5.1. Validation Simulation

Before moving through the comparison and analysis studies, validation is needed. Firstly, a nominal performance (without pendulum) of the considered controllers was considered. Secondly, the behavior of the proposed algorithms is recorded as the base point. Finally, the SOGP algorithm effects on the controller without payload were analyzed.

In Figure 8, the control algorithms' performance for the step signal command has been presented and compared. Separately, a 5 m position change in X, Y, and Z directions was simulated. The amount of the time which the command signal was applied is 7 s. This time has been measured enough to let all controllers achieve the desired command with stability. It should be remembered that the roll/pitch angle has been limited with ±30 degrees, thus achieving less than a certain amount of time is not physically possible. The black dash-dotted line shows the command signal, the red line shows the base PID algorithm, the green dashed line shows the PID algorithm with SOGP, the dotted orange line shows the base IBS algorithm, the blue solid line shows the IBS algorithm with SOGP implementation, the turquoise line shows the base IBKS algorithm, and the purple dotted line shows the IBKS algorithm with SOGP implementation.

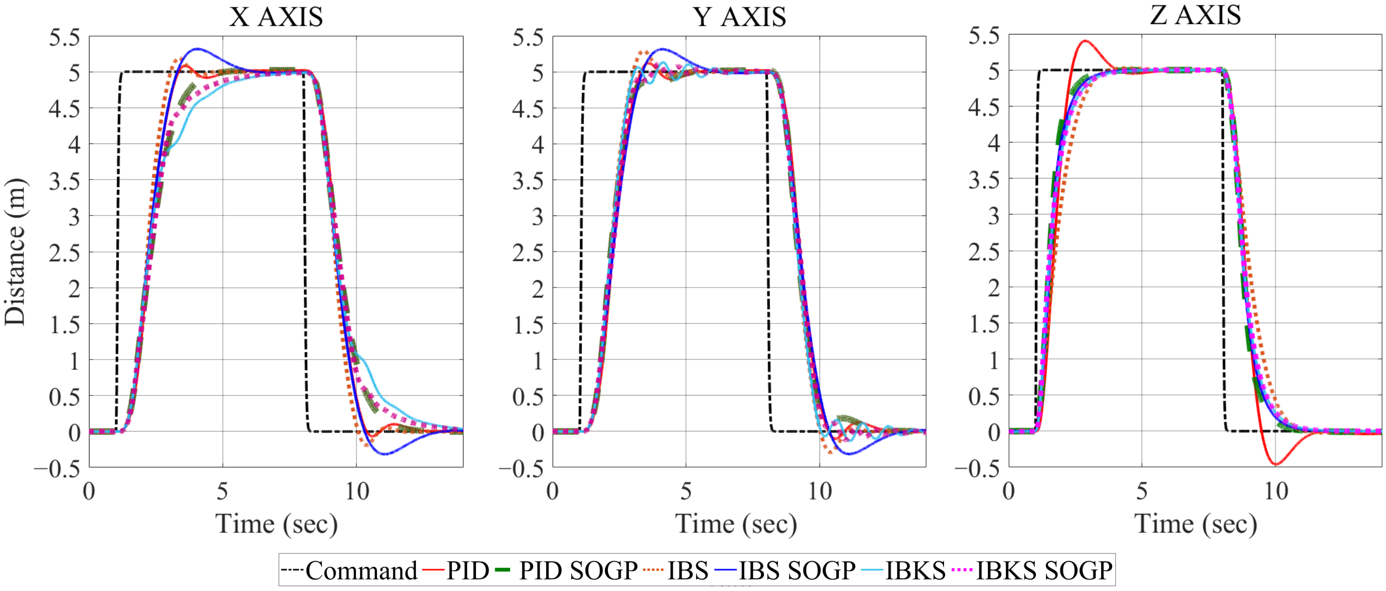

Figure 8. Step signal command for position and controller performance comparison.

All six controllers provide the desired command tracking performance. For the X-axis, PID, IBS and SOGP-implemented IBS show overshoot. The SOGP algorithm dampens the overshoot of PID, increases overshoot of IBS, and increases agility of the IBKS controller. For the Y-axis, all six controllers show overshoot except PID with SOGP. SOGP affects in a similar way the X-axis, and only the IBKS controller behaves differently since the controller has been tuned in a more agile way. Increasing the agility of the controller has caused overshoot and oscillation. However, the SOGP algorithm acts in a balanced way and decreases agility in order to avoid oscillation. In terms of Z-axis performance, all controllers shows similar behavior and change altitude with similar agility. Only the base PID controller shows an overshoot in all six controllers. However, the SOGP algorithm dampens the overshoot and PID with SOGP shows more stabilized performance. Since the study's main focus is improving the payload-carry stability, having an acceptable performance without a payload was enough.

As seen on the X-Y axis simulations, the SOGP algorithm dampens the overshoot of PID and gives more stabilized output. On the other hand, the SOGP algorithm shows a performance-increasing behavior on IBS with a shorter rise time. This also can be seen slightly in PID-based results. For the IBKS controller, although improvement has been achieved compared with the base version, it is still not as effective as other controllers. This aspect has been discussed in the conclusion section.

Validation of the proposed algorithms shows promising results for further analysis. Although SOGP implementation is designed for neutralizing slung payload effects, it is also suitable for use without payload. It is possible to claim SOGP has performance improvement.

5.2. Payload Simulation

In a similar validation study, a 5 m pulse signal on the position state was demanded from the flight controller separately. The amount of time which the command signal was applied is 7 s. For the payload, a 10 kg point mass has been selected. In this way, the controller management on the single-axis disturbance is inspected. Unlike multidimensional operations, this is a relatively simple test for algorithms. However, for close inspection purposes, it is beneficial to analyse in this way.

In Figure 9, the control algorithms' performance for the step signal command has been presented and compared. The 5 m position change in the X, Y and Z directions commanded separate simulations. The black dash-dotted line shows the command signal, the red line shows the base PID algorithm, the green dashed line shows the PID algorithm with SOGP, the dotted orange line shows the base IBS algorithm, the blue solid line shows the IBS algorithm with SOGP implementation, the turquoise line shows the base IBKS algorithm, and the purple dotted line shows the IBKS algorithm with SOGP implementation.

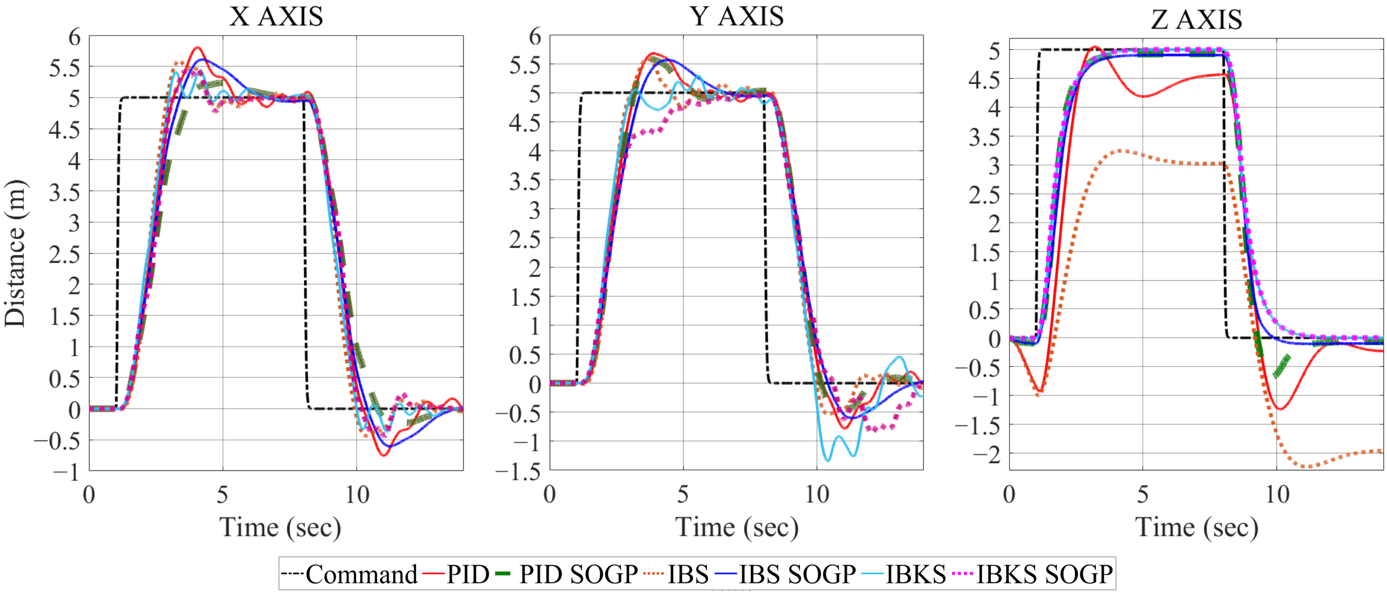

Figure 9. Step signal command for position and controller performance comparison with payload.

In X-Y axis simulations, payload swing-related position oscillation is visible on base controllers. Although IBKS has a relatively smaller amount of oscillation than PID and IBS, all controllers still fail to dampen. This should not be seen as the controllers not responding to external force disturbances at all. But, either the response time is too long or the response amount stays low compared to the slung payload effect. Also, all controllers have larger overshoots under the influence of payload.

However, on the X-axis, the backstepping algorithm responds in a different way with SOGP, and the peak value of overshoot rises with a small percentage. Although the SOGP algorithm successfully neutralizes swing effects, an increase in the overshoot seems unavoidable. Both base controllers and SOGP-integrated controllers have quite a visible overshoot, this is caused by a moment of inertia of the payload. When it comes to the Y-axis, three controllers show a similar performance with the X-axis: base PID, base IBS, and IBS with SOGp. On the other hand, PID with SOGP shows a larger overshoot. The difference in IBKS is closely related with the tuning of the controller; to achieve a more acceptable performance, and the is controller tuned in a manner with less agility. In terms of settling, the algorithms with SOGP successfully manage settling within the given time on both X-Y axes.

The behavior difference in the three controllers with SOGP shows a difference. The variety of the behavior can be related to a couple of reasons, such as the tuning of the controller and the nature of the control method. Since all controllers show similar performance without payload, all differences are accepted as differences between the method and SOGP coupling. To this extent, to provide the stability with the SOGP, the PID algorithm has to sacrifice agility and has longer rise-settling time. On the other hand, IBS also has a longer rise time and bigger overshoot. By the way, IBKS has some issues with coupling and not having smooth behavior. However IBKS also has a longer rise time a bit but gains much more stability.

The PID algorithm has the most error in the base form and the biggest change with the SOGP algorithm. This can be related with the linear control methodology since UAV dynamics move away from the linearization point. Still, rejecting the disturbance requires time. In spite of that, backstepping-based controllers tend to increase nonlinearity and lose stability. Although they remain at an acceptable level, maintaining precise stability can be a challenging issue.

During the simulations on the z-axis, the base PID and IBS controllers' vulnerability to model uncertainty has shown itself openly. A biased output has been seen on the IBS graph and also a remarkable error in PID results. Since the backstepping algorithm uses a modelled system's dynamics, any model change affects the controller performance significantly. Also, PID uses a linearized system model during the tuning and getting away from the trim point causes errors. But, especially with the help of the integration function, it covers the gap slowly over time. In contrast, the IBKS algorithm perfectly manages the error and shows the best performance. Hence, the SOGP algorithm is not able to match it.

On the other hand, the SOGP algorithm successfully manages the rejecting error and maintains stability on the PID and IBS controllers. However, on the rising part, everything seems fine; SOGP detects the error and keeps close to the starting altitude and then carries the commanded altitude with a quite small (around %2) error in PID and IBS. On the other hand, IBKS successfully manages to remove weight effect with and without SOGP. Even the SOGP algorithm successfully removes the weight effect, and the still-linear control approach still shows an overshoot with the second movement (returning the zero altitudes), which means crashing to the ground in real-life experiments. Overall, it is possible to claim that the IBS-based controller shows better performance.

Table 2 presents a comparative analysis of control performance metrics across different controller configurations for the step signal with payload, focusing on the X, Y, and Z axes. In the table, RMS (Root Mean Square), overshoot, undershoot, and rise time values have been compared. To briefly explain the purposes of these terms, the RMS value is helpful for assessing the overall tracking performance because it considers the entire signal duration, overshoot shows how much the response exceeds the target, undershoot indicates if the response falls short before reaching the steady state, and finally rise time measures the time taken to go from a low to high percentage of the final value. A lower RMS error indicates that, on average, the system is closer to the desired position, implying better tracking accuracy.

In this part of the study, the results shown in the Table 2 will be analyzed. In reference to RMS (Root Mean Square) analysis:

- X-axis: The configurations "PID w SOGP" (3.282) and "IBKS w SOGP" (3.307) achieve the lowest RMS values, suggesting superior tracking accuracy along the X-axis when SOGP is applied.
- Y-axis: The lowest RMS value is observed in the "IBKS w SOGP" configuration (3.100), indicating that this setup provides the most accurate tracking along the Y-axis.
- Z-axis: Here, the "PID" configuration stands out with the lowest RMS value (2.286), showing that, in this case, the standard PID controller is highly effective for tracking along the Z-axis.

In summary, the "IBKS w SOGP" configuration generally exhibits better tracking accuracy across the X and Y axes, while the standard PID controller performs well on the Z-axis.

Table 2. Measurements of the step signal with payload graphs.

		PID	PID w SOGP	IBS	IBS w SOGP	IBKS	IBKS w SOGP
X	RMS	3.383	3.282	3.342	3.364	3.326	3.307
	Overshoot	18%	0.526%	11.446%	13.125%	8.721	8.824%
	Undershoot	1.999%	1.997%	3.365%	1.990%	1.993%	4.504%
	Rise Time	1.235 s	1.873 s	1.076 s	1.455 s	1.236 s	1.283 s
Y	RMS	3.343	3.317	3.296	3.317	3.262	3.100
	Overshoot	13.636%	11.446%	12.963%	13.125%	4.605%	0.588%
	Undershoot	1.964%	2.188%	4.129%	1.970%	6.991%	2.792%
	Rise Time	1.212 s	1.235 s	0.982 s	1.426 s	1.152 s	2.589 s
Z	RMS	2.981	3.237	2.286	3.199	3.281	3.285
	Overshoot	9.868%	0.519%	— %	0.505%	0.503%	0.503%
	Undershoot	8.161%	1.999%	— %	1.999%	1.996%	1.997%
	Rise Time	0.885 s	1.115 s	— s	1.244 s	1.250 s	1.242 s

With respect of overshoot analysis:

- X-axis: The “PID w SOGP” configuration achieves the lowest overshoot (0.526%), demonstrating a well-controlled response compared to the other configurations.
- Y-axis: The “IBKS w SOGP” configuration also has minimal overshoot (0.588%), indicating stable control along the Y-axis.
- Z-axis: Both “PID w SOGP” (0.519%) and “IBKS w SOGP” (0.503%) configurations maintain very low overshoot, signifying controlled performance along the Z-axis.

These observations suggest that configurations with SOGP significantly reduce overshoot across all axes, making them more suitable for applications requiring precise and stable control.

With regard to undershoot analysis:

- X-axis: The “PID” and “PID w SOGP” configurations show minimal undershoot, staying below 2%, which indicates stable response without significant initial dips.
- Y-axis: “PID w SOGP” has a slightly higher undershoot (2.188%), though all configurations remain within acceptable stability limits.
- Z-axis: Undershoot is more pronounced in the “IBKS” and “PID” configurations, suggesting a slight drop in the response before stabilizing.

Overall, undershoot is generally well controlled across configurations, with slight variations but no significant impact on stability.

In terms of rise time analysis:

- X-axis: The “IBS” configuration has the shortest rise time (1.076 s), making it the fastest in responding along the X-axis.
- Y-axis: The “PID” configuration achieves the lowest rise time (1.212 s) on the Y-axis, indicating quick response.
- Z-axis: Rise times are generally consistent along the Z-axis, with “PID” and “PID w SOGP” showing times around 1.2–1.25 s.

Configurations without SOGP, specifically “PID” and “IBS”, tend to exhibit shorter rise times, suggesting they respond more quickly, although this may come at the expense of overshoot control. However, IBS was not able to catch desired command in Z axis, thus, overshoot, undershoot and rise time did not calculated.

This analysis reveals that configurations integrating SOGP generally achieve improved tracking accuracy and control, as seen in the lower RMS and overshoot values. However, they tend to have slightly longer rise times. Specifically, “PID w SOGP” and “IBKS w SOGP” configurations demonstrate superior performance in minimizing overshoot and tracking error, making them ideal for applications where stability and precision are paramount. On the other hand, the “PID” and “IBS” configurations exhibit quicker response times, which may be beneficial in scenarios where response speed is a higher priority than overshoot control.

Overall, “PID w SOGP” and “IBKS w SOGP” offer balanced performance across most metrics, providing a stable and accurate response ideal for controlled applications, while “PID” and “IBS” configurations can be considered when faster response times are essential. With the achieved results, it is possible to claim validation for rejecting the proposed disturbance type. Nevertheless, a real-life operation requires multi-dimensional movement of a UAV; moreover, complexity of movement is a crucial aspect of our study since pendulum movement can be amplified harmonically with continuous maneuvers. In the following section, a case study which fits the given points has been investigated.

5.3. Case Study: Agriculture Spraying

Within the given extent of slung payload, several different types of drone operations can be included. One of the target operations to this extent is agricultural spraying. In many studies in the literature, liquid tank sloshing has been modeled as a pendulum [14–16]. With regards to this approach, agricultural spraying drones are taken as slung payloads in terms of payload dynamics.

For realistic results, a commercially accepted spraying drone has been modeled and parameters used. Spraying drones usually have an airframe with a long diameter and big propellers in order to carry heavier payloads. To avoid dislocation and use down-wash for more precise spraying, nozzles are located under the motors, and to minimize disturbance, the pesticide tank is located in the same Z-axis direction as the CoG (center of gravity) of the drone. In Figure 10, a general spraying drone has been presented. It should be noted that, unlike the previous section, the command signal given to the controllers has been made smoother with a filter for clear presentation purposes. This can be seen in the later figures.

Figure 10. UAV spraying drone visual.

Parameters of the modeled spraying drone given in Table 3.

Table 3. Quadrotor UAV parameters [39].

Parameters	Value	Unit
Frame Weight	17	Kg
Payload Weight	10	Kg
Num of motors	4	pcs
Frame Diameter	1780	mm
Propeller	30 × 11	inch

Although the proposed modeling meets the required UAV dynamics, there are some novel attributes for the spraying. One of the special aspects is changing weight behavior; during the operation, pesticide payload slowly decreases. The other specific aspect is the operation navigation route; since the spraying needs to cover the target area iteratively, the most common operation route pattern is the Boustrophedon pattern [40]. As mentioned in the previous section, navigation pattern has a huge influence on payload dynamics.

With regard to the given information, an agricultural spraying drone with a 10 kg pesticide tank has been modeled and the spaying operation has been simulated. In Figure 11, operation results in X-Y coordinates, and in Figure 12, a three-dimensional perspective have been represented. The black dash-dotted line shows the command signal, the red line shows the base PID algorithm, the green dashed line shows the PID algorithm with SOGP, the dotted orange line shows the base IBS algorithm, the blue solid line shows the IBS algorithm with SOGP implementation, the turquoise line shows the base IBKS algorithm, and the purple dotted line shows the IBKS algorithm with SOGP implementation.

The first catch from the given graphs, similar to the previous simulations, is that the base controllers have the biggest errors. In particular, the IBS algorithm's altitude error causes big risks to the safety of the operation. On the other hand, as expected, the PID algorithm covers the altitude error within the time. Another observed pattern is the increasing instability with the change in direction. However, the payload movement behavior increases harmonically with every direction change. The linear momentum of the payload causes an increase in instability. To this extent, the number of corners should be considered during the navigation planning. On the other hand, the PID algorithm shows unstable behavior in the corners, possibly having trouble minimizing payload movement. This aspect will be investigated in further parts of the section.

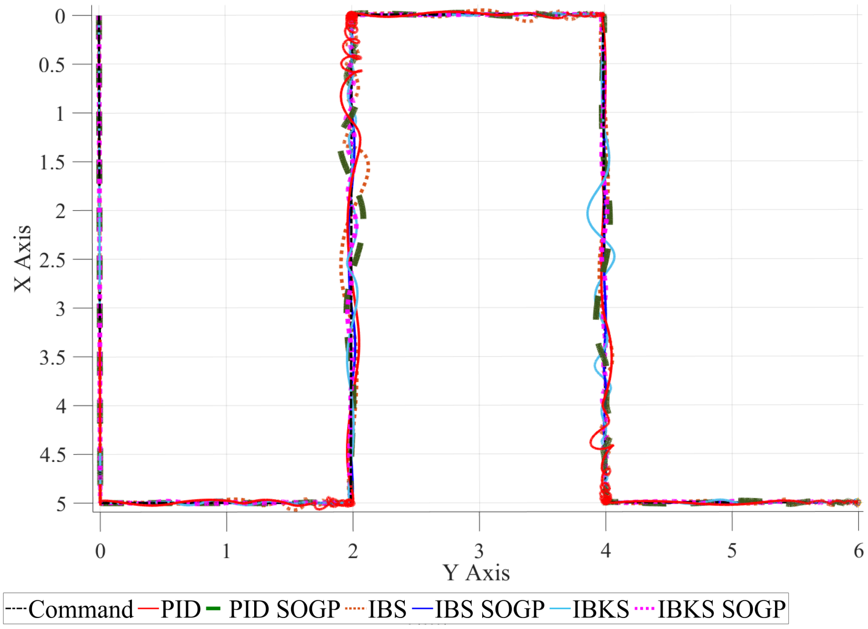

Figure 11. Spraying operation from the top.

The SOGP algorithm covers the controllers' significant error behaviors such as altitude error and instability in the corners, as well as reducing the position errors. In terms of a comparison between SOGP-implemented algorithms, it is possible to claim IBKS with SOGP has better performance. In particular, overshoots of the PID-based controller are more significant. However, it might be possible to decrease the amount of error by smoothing navigation or decreasing the speed of the drone. Despite that, increasing the agility performance of autonomous UAVs covers the main focus of this study.

In Figure 13, the error values of the controllers on the X-Y-Z position have been represented. The error graph shows the error between the desired value and the actual value. So when a command is given, it reaches the maximum. Although the peak value of the error shows controller agility and responsiveness, the proceeding way of the signals, especially, gives clues about the stability. The IBKS-based controllers' error value decrease at a slower rate. In contrast, the settling time is almost the same as the other controllers. Table 4 compares control performance metrics for different controller configurations during the spraying operation, examining the X, Y, and Z axes. Metrics include RMS value, overshoot, undershoot, and rise time, providing insights into each configuration's effectiveness in maintaining stable and accurate control.

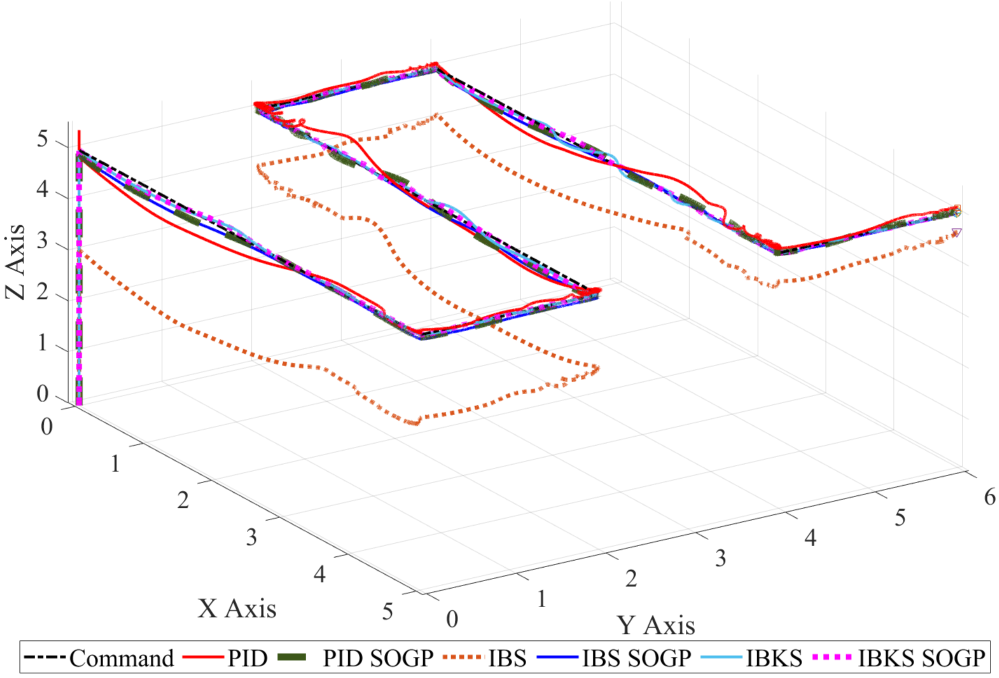

Figure 12. Spraying operation from the corner angle.

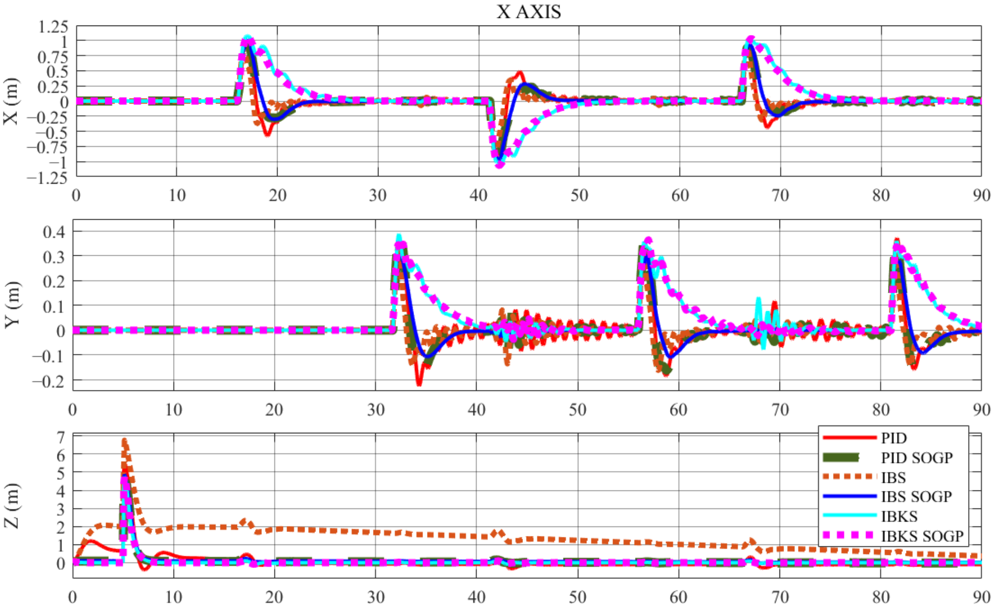

Figure 13. X, Y, Z position error values on simulation.

In this part of the study, the results shown in Table 4 will be analyzed. In terms of RMS (Root Mean Square) analysis:

- X-axis: The "IBKS" configuration yields the lowest RMS value (3.444), showing superior tracking accuracy on the X-axis without SOGp.
- Y-axis: Here, "IBKS w SOGP" has the lowest RMS (1.199), suggesting it is the most accurate configuration for tracking on the Y-axis.
- Z-axis: The "IBKS" configuration achieves a low RMS value of 3.668, indicating good performance along the Z-axis without SOGp.

Overall, "IBKS" and "IBKS w SOGP" configurations demonstrate strong tracking accuracy, particularly along the X and Y axes.

Table 4. Measurements of the spraying operation.

		PID	**PID w SOGP**	**IBS**	**IBS w SOGP**	**IBKS**	**IBKS w SOGP**
X	RMS	3.524	3.518	3.510	3.515	3.444	3.448
	Overshoot	0.690%	0.217%	0.424%	13.125%	0.505	0.487 %
	Undershoot	2.203%	1.998%	1.999%	1.999%	1.999%	1.999%
	Rise Time	3.223 s	3.218 s	3.725 s	2.982 s	4.770 s	4.919 s
Y	RMS	1.230	1.229	1.227	1.226	1.201	1.199
	Overshoot	5.851%	4.737%	6.989%	1.531%	1.531%	1.453
	Undershoot	4.123%	4.630%	4.090%	2.000%	2.624%	2.625
	Rise Time	2.659 s	2.992 s	3.866 s	3.105 s	4.769 s	4.784 s
Z	RMS	4.872	4.805	3.668	4.769	4.829	4.830
	Overshoot	5.172%	−1.322%	5.936%	−0.877%	0.473%	−0.073%
	Undershoot	11.545%	1.999%	1.981%	1.998%	1.997%	2.000%
	Rise Time	1.071 s	1.185 s	1.483 s	1.301 s	1.249 s	1.267 s

With regard to overshoot analysis:

- X-axis: The lowest overshoot is observed with "PID w SOGP" (0.217%), highlighting this configuration's stability along the X-axis.
- Y-axis: "IBKS w SOGP" has minimal overshoot (1.531%), suggesting excellent control on the Y-axis.
- Z-axis: "PID w SOGP" reduces overshoot significantly to -1.322%, indicating a controlled response with minor undershoot rather than exceeding the target.

Configurations with SOGP generally reduce overshoot, showing enhanced stability across all axes.

With respect to the undershoot analysis:

- X-axis: All configurations maintain similar undershoot levels, around 1.999%, indicating controlled response along the X-axis.
- Y-axis: "IBKS w SOGP" achieves the lowest undershoot (2.000%), supporting its stability along the Y-axis.
- Z-axis: Undershoot is consistent across configurations, with minor variations, suggesting stable control on the Z-axis.

Undershoot is generally well managed, with all configurations maintaining acceptable stability across axes.

In reference to rise time analysis:

- X-axis: The "IBS w SOGP" configuration has the shortest rise time (2.982 s), highlighting quick response on the X-axis.
- Y-axis: "PID" configuration demonstrates a relatively short rise time (2.659 s) on the Y-axis, indicating prompt response.
- Z-axis: "PID" has the shortest rise time on the Z-axis (1.071 s), showing rapid response along this axis.

The configurations without SOGP, particularly "PID" and "IBS w SOGP", tend to provide faster rise times, indicating quicker system response.

For the spraying operation, the configurations integrating SOGP (e.g., "PID w SOGP" and "IBKS w SOGP") generally achieve reduced overshoot and strong tracking accuracy, with slightly longer rise times compared to configurations without SOGp. Specifically, "IBKS w SOGP" and "PID w SOGP" perform well in controlling overshoot and undershoot, making them ideal for applications that prioritize stability and precision.

In scenarios where rapid response is critical, "PID" and "IBS w SOGP" provide shorter rise times and good overall performance, making them suitable for applications demanding quick system responsiveness.

For three-axis combined motion, the IBKS with SOGP controller shows the most stabilized output. In terms of X-Y plane, IBS with SOGP has the best performance. Nevertheless, altitude management is worse than IBKS-based controllers. Even though SOGP has damped most of the instability from the PID-only controller, still oscillations are shown in the route. On the Z-axis, SOGP-implemented PID and IBS show quite similar performance. However, for providing altitude stability also, it is possible to claim the SOGP algorithm rejects the disturbance.

Also, on the X-axis route, visible fluctuation in altitude is shown in all controllers, unlike the Y-axis. Another aspect with a similar pattern is the amount of error value on the X- and Y-axes; the average error value on the X-axis is almost three times the error on the Y-axis. This behavior is caused because by the amount of commanded signal; the command given is on the axis at 5 m, whereas the command on the Y-axis is only 2 m. The amount of the time tilted in maximum value (±30°) affects the altitude. With a lower maximum tilting value, this affect can be minimized. However, in the missions that operate in narrow places, this aspect should be considered.

As given in the modeling section, the angle between the Z-axis and pendulum payload has been named alpha (α) and the pendulum angle on the X-Y coordinate is named beta (β). Changes in these angles during the simulation give important clues about controller management and dominance in the system. In Figure 14, the relation between position change, α angle change, β angle change, and weight change has been presented. Each group of lines is defined in the legends of graphs.

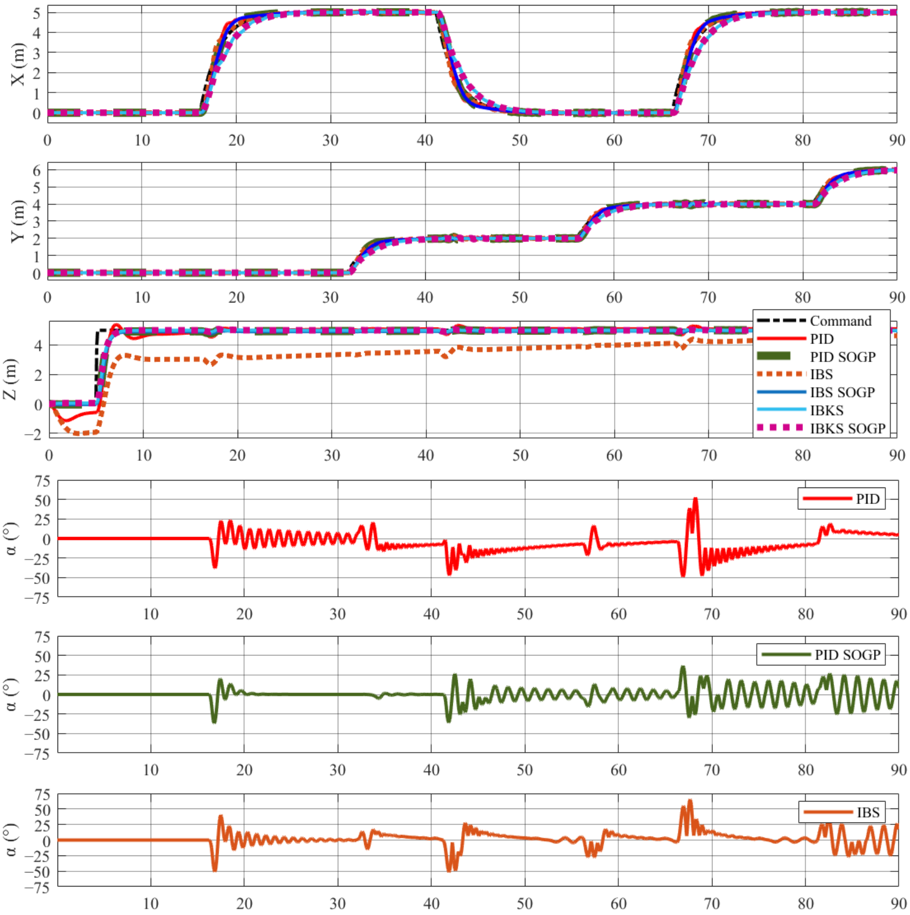

Figure 14. *Cont.*

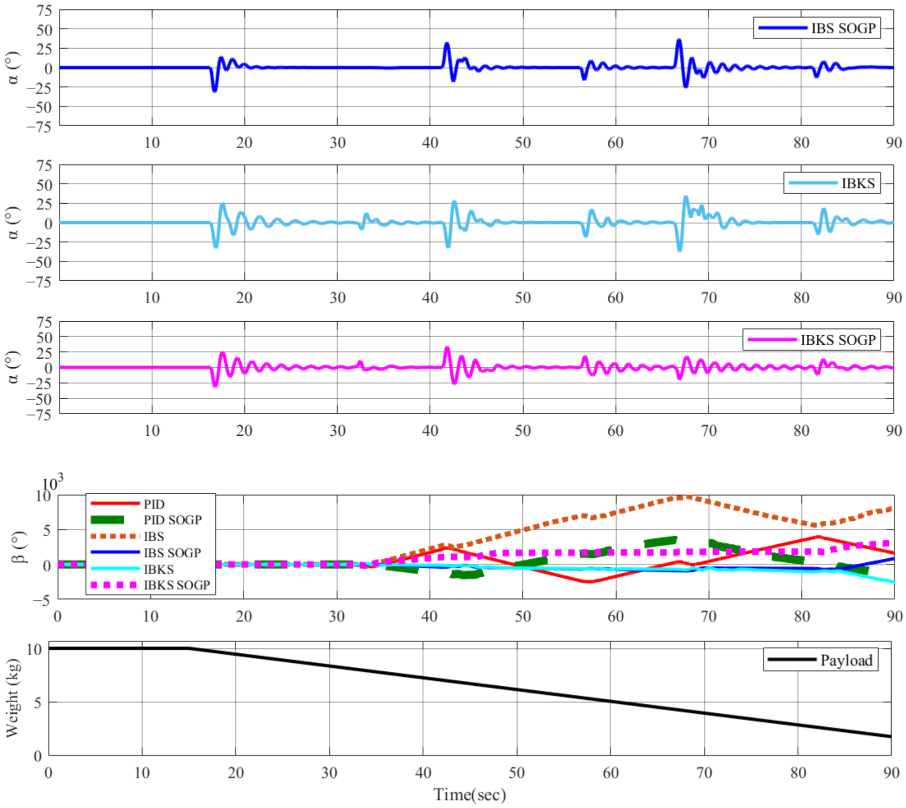

Figure 14. X-Y-Z position, alpha angle, beta angle and weight change relation.

The controllers' performance difference also shows itself in the α angle. While base controllers require more time to dampen pendulum movement, SOGP-implemented controllers seem more stable. The only exception to this matter is IBKS-based controllers. SOGP algorithms have issues with integrating incremental dynamics and instead of damping the pendulum completely only reduce it to a certain level. However, this allows the controller to keep the UAV in the desired position. Furthermore, there is a visible difference between IBS and PID-based controllers on both sides. IBS-based controllers (with and without SOGP) are more successful at damping pendulum movement. This behavior should affect the outcome of the position graph. Therefore, the force disturbance is closely related to the angular rate of the pendulum.

Another aspect is increasing swing angles and speed with time; with the reduction in weight, the UAV's movement force causes faster swinging. Moreover, the payload movement behavior increases harmonically with every direction change and decreased weight helps increase the swing angle. Following the tradition of swinging in a direction on the X-Y coordinate, the β angle keeps increasing during the operation. The more damped movements are seen in the β angle also. Beneficial to stabilizing, navigation planning can be planned to neutralize the beta angle movement.

6. Conclusions

In this paper, disturbances which swing a slung payload carried by a quadcopter have been inspected and the solution has been proposed with the Sparse Online Gaussian Process. Firstly, the modeling of the quadcopter and slung payload is explained. Then, control algorithms were explained briefly. Due to the highly nonlinear nature of the quadrotor, a cascaded structure of control has been used. Also, to achieve a more comprehensive result, PID and integrator backstepping controllers have been used in position control. Finally, Sparse Online Gaussian Process algorithms are explained briefly.

With the gathered modeling and control methods, a series of numerical simulations has been performed. The SOGP algorithm has been implemented with the PID, IBS, and IBKS algorithms. With these implementations, a total of four controllers have been included in the simulations. Firstly, validation of the proposed four controllers was performed with pulse command on each position axis separately. Secondly, controllers were compared with the same commands as the payload. Thirdly, a payload carrying case, an agricultural spraying operation, was simulated.

It is possible to claim that base PID and IBS controllers are far away from satisfying results with stability. However, the SOGP algorithm successfully covers this aspect and gives an acceptable outcome. Another noticeable aspect is the base IBKS algorithm's performance and cooperation with SOGP. With the support of incremental design, base IBKS more successfully manages the drone than the other base controllers. And with SOGP implementation, the IBKS algorithm shows the best performance in the case study. Typically, the slower behavior of the IBKS is a disadvantage for use in position controlling. However, in terms of maintaining payload stability, this aspect is beneficial.

Although the IBKS algorithm was successful, a groundbreaking improvement with the SOGP was not observed. Still, small oscillations show themselves. Regardless of the attempt to improve the tuning of both controllers, any better result was not achieved. This result shows more than one feed-forward algorithm might have integration problems. Results of the simulations were discussed with consideration of the nature of the control methodologies and slung payload behavior. In general, implementation of the Sparse Online Gaussian Process provided better results.

The topic of slung payload transport presents numerous avenues for further research, particularly in enhancing navigation accuracy, addressing shifts in the center of gravity (CoG) during flight, enabling efficient multi-UAV collaboration, and implementing dynamic gain scheduling for improved adaptability. Although the proposed algorithms—IBKS and SOGP—have shown promising results, certain limitations remain due to their reliance on sensor data. Both algorithms are sensitive to the quality and accuracy of feedback information, meaning that any distortion or noise in sensor readings can degrade their performance and stability.

Future work will prioritize the development of robust strategies to mitigate the impact of sensor inaccuracies, possibly through sensor fusion techniques or fault-tolerant control mechanisms, to ensure more reliable feedback data. Additionally, efforts will focus on optimizing the gain parameters dynamically within the control algorithms to improve adaptability across varying payload conditions and operational environments. By enhancing the resilience of IBKS and SOGP to sensor errors and fine-tuning gain scheduling, this research aims to advance the reliability and versatility of UAVs in slung payload transport, paving the way for broader practical applications and increased operational efficiency.

Author Contributions: Conceptualization, M.R.K. and D.I.I.; methodology, M.R.K.; software, M.R.K.; validation, M.R.K.; formal analysis, M.R.K.; investigation, M.R.K.; resources, M.R.K.; data curation, M.R.K.; writing—original draft preparation, M.R.K.; writing—review and editing, D.I.I.; visualization, M.R.K.; supervision, D.I.I. and A.Z.; project administration, D.I.I. and A.Z.; funding acquisition, M.R.K. and A.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to express very great appreciation to the Ministry of National Education of the Republic of Türkiye for funding this project.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
PID	Proportional–Integral–Derivative
IBS	Integrator Backstepping
IBKS	Incremental Backstepping
SOGP	Sparse Online Gaussian Process
6DoF	6-degree-of-freedom
GM	Gain Margin
PM	Phase Margin
INDI	Incremental Dynamic Inversion
CLF	Control Lyapunov Function
GP	Gaussian Process
CoG	Center of Gravity

References

1. Kumar, S.S.; Gudipalli, A. A comprehensive review on payloads of unmanned aerial vehicle. *Egypt. J. Remote. Sens. Space Sci.* **2024**, *27*, 637–644. [CrossRef]
2. Askarzadeh, T.; Bridgelall, R.; Tolliver, D. Monitoring Nodal Transportation Assets with Uncrewed Aerial Vehicles: A Comprehensive Review. *Drones* **2024**, *8*, 233. [CrossRef]
3. Yüksel, Z.; Epcim, D.E.; Mete, S. Multi-Depot vehicle routing problem with drone collaboration in humanitarian logistic. *J. Optim. Decis. Mak.* **2024**, *3*, 438–448.
4. Imran, I.H.; Wood, K.; Montazeri, A. Adaptive Control of Unmanned Aerial Vehicles with Varying Payload and Full Parametric Uncertainties. *Electronics* **2024**, *13*, 347. [CrossRef]
5. Zolotas, A. Simple Internal Model-Based Robust Control Design for a Non-Minimum Phase Unmanned Aerial Vehicle. *Machines* **2023**, *11*, 498. [CrossRef]
6. Geronel, R.S.; Bueno, D.D. Adaptive sliding mode control for vibration reduction on UAV carrying a payload. *J. Vib. Control* **2024**, *30*, 459–925.
7. Ullah, N.; Sami, I.; Shaoping, W.; Mukhtar, H.; Wang, X.; Shahariar Chowdhury, M.; Techato, K. A computationally efficient adaptive robust control scheme for a quad-rotor transporting ca-ble-suspended payloads. *Proc. Inst. Mech. Eng. Part J. Aero-Space Eng.* **2022**, *236*, 379–395. [CrossRef]
8. Bingöl, Ö.; Güzey, H.M. Neuro sliding mode control of quadrotor UAVs carrying sus-pended payload. *Adv. Robot.* **2021**, *35*, 255–266. [CrossRef]
9. Ergezer, H.; Leblebicioğlu, K. Control structure design with constraints for a slung load quadrotor system. *Meas. Control* **2024**, *57*, 16–29. [CrossRef]
10. Arizaga, J.M.; Miranda-Moya, A.; Castañeda, H.; Castillo, p. Observer-based adaptive control for slung payload stabilization with a fully-actuated multirotor UAV. *ISA Trans.* **2024**, *147*, 109–117. [CrossRef]
11. Grehan, J.; Ignatyev, D.; Zolotas, A. Fault detection in aircraft flight control actuators using support vector machines. *Machines* **2023**, *11*, 211. [CrossRef]
12. Krstic, M. Nonlinear and Adaptive Control Design. *Wiley* **1995**, *2*, 69–79.
13. Bacon, B.; Ostroff, A. Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Denver, CO, USA, 14–17 August 2000; p. 4565.
14. Feddema, J.T.; Dohrmann, C.R.; Parker, G.G.; Robinett, R.D.; Romero, V.J.; Schmitt, D.J. Control for slosh-free motion of an open container. *IEEE Control Syst. Mag.* **1997**, *17*, 29–36. [CrossRef]
15. Yano, K.; Terashima, K. Sloshing suppression control of liquid transfer systems consid-ering a 3-D transfer path. *IEEE ASME Trans. Mechatron.* **2005**, *10*, 8–16. [CrossRef]
16. Zang, Q.; Huang, J.; Liang, Z. Slosh suppression for infinite modes in a moving liquid container. *IEEE ASME Trans. Mechatron.* **2014**, *20*, 217–225. [CrossRef]
17. Kurode, S.; Spurgeon, S.K.; Bandyopadhyay, B.; Gandhi, P.S. Sliding mode control for slosh-free motion using a nonlinear sliding surface. *IEEE ASME Trans. Mechatron.* **2012**, *18*, 714–724. [CrossRef]
18. Toumi, M.; Bouazara, M.; Richard, M.J. Impact of liquid sloshing on the behaviour of ve-hicles carrying liquid cargo. *Eur. J. Mech. A Solids* **2009**, *28*, 1026–1034. [CrossRef]
19. Navabi, M.; Davoodi, A. Fuzzy control of fuel sloshing in a spacecraft. In Proceedings of the 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Kerman, Iran, 28 February–2 March 2018; pp. 123–126. [CrossRef]
20. Reyhanoglu, M.; Hervas, J.R. Robotically controlled sloshing suppression in point-to-point liquid container transfer. *J. Vib. Control* **2013**, *19*, 2137–2144. [CrossRef]

21. Dong, K.; Qi, N.M.; Guo, J.J.; Li, Y.Q. An estimation approach for propellant sloshing ef-fect on spacecraft gnc. In Proceedings of the 2008 2nd International Symposium on Systems and Control in Aerospace and As-tronautics, Shenzhen, China, 10–12 December 2008; pp. 1–6. [CrossRef]
22. Nichkawde, C.; Harish, P.M.; Ananthkrishnan, N. Stability analysis of a multibody system model for coupled slosh—Vehicle dynamics. *J. Sound Vib.* **2004**, *275*, 1069–1083. [CrossRef]
23. Sopasakis, P.; Bernardini, D.; Strauch, H.; Bennani, S.; Bemporad, A. Sloshing-aware at-titude control of impulsively actuated spacecraft. In Proceedings of the 2015 European Control Conference (ECC), Linz, Austria, 15–17 July 2015; pp. 1376–1381. [CrossRef]
24. Constantin, L.; De Courcy, J.; Titurus, B.; Rendall, T.C.; Cooper, J.E. Analysis of damping from vertical sloshing in a SDOF system. *Mech. Syst. Signal Process.* **2021**, *152*, 107452. [CrossRef]
25. Carton, P.; Chrétien, J.P.; Maurette, M. Simulation and control of space manipulators bearing complex payloads. *IFAC Proc. Vol.* **1989**, *22*, 127–132. [CrossRef]
26. Wu, C.H.; Chen, B.F. Sloshing waves and resonance modes of fluid in a 3D tank by a time-independent finite difference method. *Ocean. Eng.* **2009**, *36*, 500–510. [CrossRef]
27. Robu, B.; Baudouin, L.; Prieur, C.; Arzelier, D. Simultaneous H_∞ Vibration Control of Fluid/Plate System via Reduced-Order Controller. *IEEE Trans. Control Syst. Technol.* **2011**, *20*, 700–711. [CrossRef]
28. Robu, B.; Baudouin, L.; Prieur, C. A controlled distributed parameter model for a flu-id-flexible structure system: Numerical simulations and experiment validations. In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15 December 2009; pp. 5532–5537.
29. Guerrero, M.E.; Mercado, D.A.; Lozano, R.; García, C.D. IDA-PBC methodology for a quadrotor UAV transporting a cable-suspended payload. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 470–476. [CrossRef]
30. van Gils, P.; Van Kampen, E.J.; de Visser, C.C.; Chu, Q.p. Adaptive incremental backstepping flight control for a high-performance aircraft with uncertainties. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, USA, 4–8 January 2016; p. 1380. [CrossRef]
31. Kartal, M.R.; Ignatyev, D.; Zolotas, A. Non-linear Control of a Quadrotor with Actuator Delay. In Proceedings of the AIAA SCITECH, Orlando, FL, USA, 8–12 January 2024; p. 0569.
32. Abbas, N.; Abbas, Z.; Zafar, S.; Ahmad, N.; Liu, X.; Khan, S.S.; Foster, E.D.; Larkin, S. Survey of Advanced Nonlinear Control Strategies for UAVs: Integration of Sensors and Hybrid Techniques. *Sensors* **2024**, *24*, 3286. [CrossRef]
33. Pollini, L.; Metrangolo, A. Simulation and robust backstepping control of a quadrotor aircraft. In Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Honolulu, HI, USA, 18 August 2008; p. 6363.
34. Ignatyev, D.; Shin, H.S.; Tsourdos, A. Gaussian process adaptive incremental backstepping flight control. In Proceedings of the AIAA SCITECH, Forum 2022, San Diego, CA, USA, 3–7 January 2022; p. 2032. [CrossRef]
35. Farrell, J.; Polycarpou, M.; Sharma, M. On-line approximation based control of uncertain nonlinear systems with magnitude, rate and bandwidth constraints on the states and actuators. In Proceedings of the 2004 American Control Conference, Boston, MA, USA, 30 June 2004; Volume 3, pp. 2557–2562. [CrossRef]
36. Ignatyev, D.I.; Shin, H.S.; Tsourdos, A. Sparse online Gaussian process adaptation for in-cremental backstepping flight control. *Aerosp. Sci. Technol.* **2023**, *136*, 108157. [CrossRef]
37. Csató, L.; Opper, M. Sparse on-line Gaussian processes. *Neural Comput.* **2002**, *14*, 641–668. [CrossRef]
38. Capone, A.; Hirche, S. Backstepping for partially unknown nonlinear systems using Gaussian processes. *IEEE Control Syst. Lett.* **2019**, *3*, 416–421. [CrossRef]
39. Eft E410p 4 Axis 10L Farm Drone Agriculture Spraying Drone DIY Complete Version. ARRISHOBBY. (n.d.). 2024. Available online: https://www.arrishobby.com/products/arris-e410p-4-axis-10l-farm-drone-agriculture-spraying-drone-complete-version?VariantsId=15920 (accessed on 1 September 2024).
40. Choset, H.; Pignon, p. Coverage path planning: The boustrophedon cellular de-composition. In *Field and Service Robotics*; Springer: London, UK, 1998; pp. 203–209.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.