Robotica (2025), **43**, pp. 2376–2397
doi:10.1017/S0263574725101719

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Design and experimental validation of IDA-PBC-based flight control for quadrotors

Jesús-Renato Montoya-Morales[1], María-Eusebia Guerrero-Sánchez[1,2], Guillermo Valencia-Palomo[1], Omar Hernández-González[1], Francisco-Ronay López-Estrada[3] and Luis C. Félix-Herrán[4]

[1]Tecnológico Nacional de México, IT de Hermosillo, Hermosillo 83170, Mexico
[2]IIxM SECIHTI-Tecnológico Nacional de México, IT Hermosillo, Hermosillo 83170, Mexico
[3]TURIX-Dynamics Diagnosis and Control Group, Tecnológico Nacional de México, IT Tuxtla Gutiérrez, Tuxtla Gutierrez 29050, Mexico
[4]School of Engineering and Sciences, Tecnológico de Monterrey, Hermosillo 83000, Mexico
Corresponding author: Guillermo Valencia-Palomo; Email: gvalencia@hermosillo.tecnm.mx

Received: 15 October 2024; **Revised:** 14 February 2025; **Accepted:** 2 May 2025; **First published online:** 18 June 2025

Keywords: IDA-PBC; UAV; port-Hamilton controller; Parrot Bebop 2; experimental tests

Abstract

This paper presents the design and experimental validation of a robust flight control strategy for quadrotor unmanned aerial vehicles (UAVs) based on the Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) methodology. The proposed approach is specifically tailored to the Parrot Bebop 2, a commercial UAV. The IDA-PBC control law is derived using the Hamiltonian model of the UAV dynamics obtained from experimental data to represent the dynamics of all six degrees of freedom, including translational and rotational motions. The control strategy was validated through numerical simulations and experimental tests conducted in an indoor flight setup using MATLAB, Robot Operating System, and an OptiTrack motion capture system. Numerical and experimental results demonstrate that the controller effectively tracks desired flight trajectories, ensuring stable and robust performance.

1. Introduction

Unmanned aerial vehicles (UAVs) have spread to various areas of society, and their use has been intensified, adding value to production chains and services. From UAVs assisting in search and rescue tasks to those used in light shows, it is common to find these devices in daily life [27]. Whether deployed for load transport, security tasks, or other applications, effective modeling and control strategies play a critical role in enhancing the performance of UAV operations.

Most reported results tackle the control problem using inner-outer hierarchical schemes based on first-principle mathematical models. This approach is based on deriving differential equations to represent the system's dynamic behavior using the knowledge of the physical laws governing the system's dynamics. Subsequently, the resulting differential equations are transformed to establish an exploitable version of the mathematical model, followed by the control design stage. Here, the design is divided into two control loops (layers). On the one hand, the inner loop concerns the UAV rotation, and on the other, the outer loop concerns the UAV translation, and the controllers are designed sequentially. A wide variety of control algorithms have been developed for UAVs using this approach. For instance, a modified robust integral of signum error nonlinear control method for quadrotor trajectory tracking and control is presented in [18]. Another approach, described in [1], introduces a fault-tolerant nonlinear observer controller based on sliding mode control (SMC). The solution includes a nonlinear observer to diagnose faults, and its fault estimates are integrated into the controller to adjust the control law. Furthermore,

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

[6] presents an SMC-based strategy applying metaheuristic optimization techniques such as particle swarm optimization, gray wolf optimizer, and equilibrium optimizer. Additional contributions include the model predictive control (MPC) with fault-tolerance [17], a neural network-based optimal mixed H_2/H_∞ control system designed for specialized UAVs for precise trajectory tracking missions [4], and a proportional integral derivative (PID) with model-based optimization approach alongside internal mode control [5]. However, PID controllers are simple to implement and effective for basic stabilization tasks, but they typically lack robustness to nonlinearities and unmodeled dynamics. Similarly, SMC provides robustness to uncertainties and external disturbances but may introduce chattering effects, which can degrade performance in UAV applications. While MPC is particularly effective in handling multivariable systems and constraints, it is computationally intensive, making real-time implementation on UAVs challenging, especially for systems with fast dynamics like quadrotors. Moreover, innovative concepts like a hybrid intelligent agent controller for aerial vehicle formation [31] and the use of A^* and linear quadratic regulator (LQR) algorithms for efficient path planning [26] have also been explored. However, these strategies lack robustness against unmodeled dynamics and parametric uncertainty, although an extension to a Gaussian quadratic regulator (LQG) may improve the performance to some extent [3]. Thus, there is a need to explore alternative paradigms to enhance UAV flight performance indices. In contrast, the energy approach combines the strengths of the previously discussed approaches, providing robustness concerning unmodeled dynamics and parametric uncertainty in the sense that stability is preserved in the energy approach. This concept is based on the kinetic and potential energies of the UAV and their variations during flight. Within this framework, the system's total energy is modeled [20]. Then, sets of partial differential equations (PDEs) are obtained whose solution provides controllers that couple the dynamics of the states, which is extremely useful in underactuated systems, such as UAVs. This advantage is part of the motivation for this work. Some successful energy-based control strategies for UAVs with outstanding results are presented in [7, 8].

Within the set of energy-based control techniques, the Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) strategy arises as a promising approach for UAV control. Initial applications of this method demonstrate its efficacy in achieving desired outcomes for path tracking in quadrotor UAVs, for both high and low-level control (position and orientation) [25]. The IDA-PBC controller could be further refined to include the port-controlled Hamiltonian (PCH) approach. This modular controller framework needs the development of an energy-based model tailored to the specific application and its underlying physical dynamics. The synthesis of the controller involves defining a minimum energy function [23]. In recent advancements, the authors in [10] developed an IDA-PBC algorithm to stabilize an unmanned powered parachute aerial vehicle. Also, [15] introduced innovative elements into the control formulation to address challenges posed by external disturbances, thereby enhancing system robustness. Additionally, integrating filtered observers into the IDA-PBC architecture has been instrumental in estimating the state vector from noisy signals, thereby augmenting UAV performance during trajectory tracking tasks [14]. These developments underscore the continuous refinement and adaptation of IDA-PBC strategies to meet the evolving demands of UAV control.

The recent results of IDA-PBC applied to UAVs are promising; however, the mathematical model of the UAV used to design the robust control system is based on first principles, and the validation was carried out at the simulation level. Also, the inner loop (rotation dynamics) is considered controlled. Furthermore, it is assumed that the state vector measurements are available; thus, in practice, a state observer is needed. In this context, previous works overlook the practical considerations in obtaining the mathematical model used in the controller design and implementation.

On the other hand, the UAVs commercialized by the company Parrot Drone SAS have served as experimental platforms for validating autonomous control and navigation strategies in recent years. In [9], the development of an autonomous visual navigation system for the AR.Drone 2.0 is described based on the implementation of PID controllers and the parallel tracking and mapping technique. Similarly, [11] discusses the utilization of ORB-SLAM for the same vehicle. Other examples include [16], where the AR.Drone 2.0 is employed to validate a fractional SMC strategy, and in [12] and [13], the UAV is used to validate energy-based techniques, incorporating the use of quaternions. In [24], the UAV is applied

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

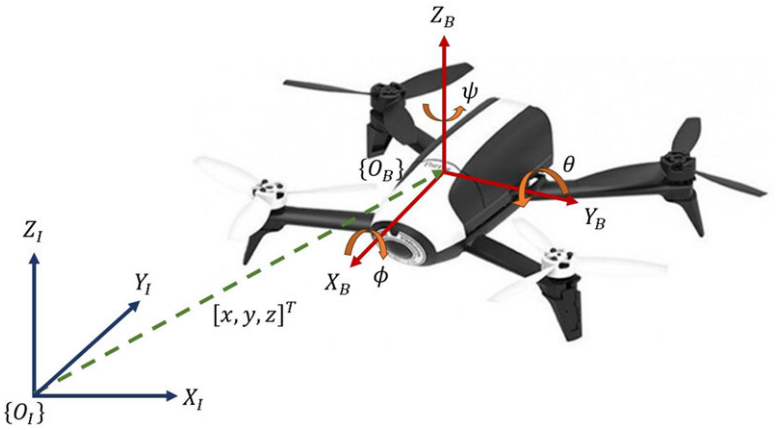


Figure 1. *Quadrotor degrees of freedom and reference frames.*

to validate a backstepping controller. In [29], the use of the LSD-SLAM technique is documented for the autonomous navigation of the Parrot Bebop 2. Additionally, in [30], it plays a role in designing and validating a fault-tolerant incremental SMC. Furthermore, in [28], a pair of these UAVs is used for the cooperative transport of a load. Also, [2] presents a case where the Parrot Mambo is stabilized using the adaptive super-twisting reaching law. The authors in [21] and [22] implement backstepping control techniques that take into account the dynamics of the internal orientation controller of the AR.Drone 2.0 and Bebop 2.0 quadrotors, respectively. Finally, [19] shows the results of experimental tests based on sliding mode control algorithms to stabilize an AR.Drone 2.0 aerial vehicle.

Therefore, this work aims to experimentally validate the literature's most recent energy-based control method on a commercially available quadrotor-type UAV, using flight data to characterize the mathematical model. Specifically, the contribution of this work consists of the application of an IDA-PBC approach tailored for a Parrot Beebop 2 vehicle. Unlike traditional methods relying on a predefined first-principle mathematical model, the IDA-PBC control law is derived via the Hamiltonian model of the UAV dynamics obtained from experimental data.

The rest of the manuscript is organized as follows. Section 2 develops the data-driven UAV identification and mathematical modeling based on the Hamiltonian. Section 3 is dedicated to the IDA-PBC formulation. Section 4 describes numerical results that demonstrate the performance of the control strategy, and Section 5 presents the experimental validation using the Parrot Beebop 2. Conclusions and future work close the manuscript in Section 6.

2. Identification and mathematical modeling

Quadrotors are a type of UAV designed for vertical take-off and landing. They possess six degrees of freedom (DOF), corresponding to three translational coordinates (x, y, z) and three rotational motions around their principal axes (ϕ, θ, ψ), as depicted in Figure 1. Additionally, quadrotors are underactuated systems: while rotational dynamics and altitude can be directly controlled, translational dynamic depends on the vehicle's orientation.

As shown in Figure 1, two reference frames are used for control and analysis. The first is the global or inertial frame $\{O_I\} = \{X_I, Y_I, Z_I\}$, typically used in ground control stations. The second is the local body-fixed frame $\{O_B\} = \{X_B, Y_B, Z_B\}$, attached to the UAV's center of mass. The coordinates in the inertial frame $\{O_I\}$ are considered absolute.

Parrot UAVs are commercial flight systems where the rotational dynamic is particularly well-controlled. Specifically, the roll and pitch angles have angular position control, while altitude and yaw angles are managed through velocity control. The Parrot Bebop 2 is controlled through four input signals:

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

u_x and u_y are lateral control inputs associated with the roll and pitch torques, respectively; u_ψ represents the yaw control input, which affects the yaw torque; and u_z is the vertical thrust input, influencing the upward or downward force. The Software Development Kit (SDK) allows the manipulation of these control signals. The control actions u_x, u_y, u_ϕ, and u_z are dimensionless, normalized inputs defined within the range $[-1, 1]$, as per the specifications of the Parrot Bebop 2's SDK. These normalized values correspond to scaled control signals for the UAV's physical actuators. The UAV's onboard systems translate these normalized inputs into actual thrust (n) and torques (N·m) based on its internal control architecture and motor dynamics. Throughout the manuscript, the control actions are presented in normalized form for consistency with the SDK and ease of analysis, while the physical interpretation aligns with the UAV's actuation mechanisms. Additionally, using its onboard sensors, it transmits the velocities $(\dot{x}_B, \dot{y}_B, \dot{z}_B)$, position z_B, and angular positions (ϕ, θ, ψ), all relative to the body-fixed frame. To complement the measurements, the global position coordinates (x, y, z) of the UAV referenced to a fixed point can be obtained using the internal GPS, which operates at 1 Hz, or a motion capture system in indoor flights. However, there is no access to low-level variables such as the individual angular velocity or power consumption of each motor.

In [19] and [21], a model of the internal dynamics for the Parrot AR.Drone is presented, implementing a first-order identification of the internal controller dynamics. As described in [22], both the Parrot AR.Drone, using the SDK ARDroneSDK2, and the Parrot Bebop 2, equipped with the SDK ARDroneSDK3, feature internal control systems built on similar frameworks. The SDK for the Bebop 2 was developed from the AR.Drone's SDK, preserving the same functions and data handling processes. As a result, the structure of input and output messages is consistent between the two models, enabling the internal dynamics of the Parrot Bebop 2 to be modeled using the same terminology and variables as those used for the AR.Drone. Inspired by these works, a model identification is performed to obtain the mathematical model of the Parrot Bebop 2.

To identify the UAV's model parameters, each control input was individually excited using a step signal while the UAV was in its equilibrium position during hover, with the resulting responses captured and recorded through a motion capture system. This data was then used to perform an offline identification of the UAV's dynamics for subsequent control purposes. The objective is to identify the dominant input-output relationships, that is, the effects of u_x on θ, u_y on ϕ, u_ψ on ψ, and u_z on z, which are critical for the UAV's control. The effects of cross-couplings, such as the effect of u_y on θ or u_z on ϕ, may exist. However, they were assumed to have minimal influence within the operating range tested and were not explicitly modeled. Finally, the model assumes constant system parameters, such as consistent motor efficiency and battery performance, without accounting for potential variations due to wear, temperature changes, or battery consumption over time.

From the step response data, a second-order approximation of the system dynamics is derived. The parameter identification is performed using nonlinear least squares optimization, specifically the Levenberg–Marquardt algorithm. This algorithm considers a set of N observations $(\tilde{x}_i, \tilde{y}_i)$ and a model $f(\tilde{x}, \Theta)$ that depends on parameters Θ. The goal is to find the parameter vector $\Theta \in \mathbb{R}^j$ that minimizes the sum of squared errors (residuals):

$$S(\Theta) = \sum_{i=1}^{N} \left[\tilde{y}_i - f(\tilde{x}_i, \Theta)\right]^2. \tag{1}$$

The Levenberg–Marquardt algorithm updates the parameter vector Θ iteratively to reduce $S(\Theta)$ until convergence. The algorithm begins by initializing the parameter vector with an initial guess Θ_0 and setting an initial damping parameter λ_0, typically a small positive value that balances the need for fast convergence with the need for stability. In each iteration, the algorithm computes the residual vector r and the Jacobian matrix J at the current parameter vector,

$$r_i = y_i - f(\tilde{x}_i, \Theta); \qquad J_{ij} = \frac{\partial f(\tilde{x}_i, \Theta)}{\partial \Theta_j}. \tag{2}$$

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

Using these, the algorithm solves a set of modified normal equations to determine the parameter update $\Delta\Theta$

$$\left(J^T J + \lambda I\right) \Delta\Theta = J^T r. \tag{3}$$

After computing $\Delta\Theta$, the parameters are updated as follows:

$$\Theta_{k+1} = \Theta_k + \Delta\Theta_k. \tag{4}$$

Next, the algorithm evaluates the sum of squared errors $S(\Theta)$ using the updated parameters. If S has decreased, the algorithm accepts $\Delta\Theta$ and reduces λ to allow for potentially larger steps in the next iteration. If S has not decreased, it rejects $\Delta\Theta$, and λ is increased to make the algorithm more cautious in subsequent iterations. This process of updating and evaluating the parameters continues iteratively. The algorithm checks for convergence in different ways. For example, it looks at how much the sum of squared errors decreases with each iteration, how much the parameters change, and whether a maximum number of iterations has been reached. When any of these criteria are met, the algorithm concludes, and the current parameter values are considered the solution.

The identification algorithm is supplied with the value of $\lambda_0 = 1 \times 10^{-2}$, and it stops when the change in $S(\Theta)$ between iterations is smaller than 1×10^{-6} or when the maximum number of iterations, set to 400, is reached. Figure 2 compares the outputs of the obtained mathematical model and the Parrot Bebop 2 in response to step inputs. The solid lines in the figure represent the outputs generated by the mathematical model, while the dashed lines correspond to the experimental measurements obtained from the Parrot Bebop 2. This comparison highlights the alignment between the modeled dynamics and the real system behavior, demonstrating the accuracy of the system identification process. The identified dynamics have a mean square error lower than 20% for each output. For instance, θ and u_x have 4.52%; ϕ and u_y, 7.74%; $\dot{\psi}$ and u_ψ, 19%; z and u_z, 16%; x and θ, 6%; and y and ϕ, 3%. For these relationships, the following state-space model has been derived:

$$\ddot{\theta} = -259.1\dot{\theta} - 9042\theta + 1.848 \times 10^5 u_x; \tag{5}$$

$$\ddot{\phi} = -247.9\dot{\phi} - 1.456 \times 10^4 \phi + 4.026 \times 10^5 u_y; \tag{6}$$

$$\ddot{\psi} = -28.95\dot{\psi} + 19.17u_\psi; \tag{7}$$

$$\ddot{z} = -51.19\dot{z} + 50.26u_z; \tag{8}$$

$$\ddot{x} = -0.753\dot{x} + 0.7\theta; \tag{9}$$

$$\ddot{y} = -0.478\dot{y} + 0.463\phi; \tag{10}$$

where u_x, u_y, u_z, and u_ψ are the control signals applied to the UAV to control the movements of their respective DOF in the O_B reference frame.

By defining $\bar{X} = [\theta, \phi, \psi, z, x, y]^T$ as $\bar{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^T$ and $\bar{u} = [u_x, u_y, u_\psi, u_z]^T$, the state-space model (5) – (10) can be expressed in compact form by

$$M\ddot{\bar{q}} + C\dot{\bar{q}} + D\bar{q} = B\bar{u}. \tag{11}$$

Considering that the vehicle can rotate around the Z_B-axis in the ψ angle through the matrix B_1, the B matrix is defined as

$$B = \begin{bmatrix} B_1 \\ 0_{2\times 4} \end{bmatrix} \quad \text{with} \quad B_1 = \begin{bmatrix} C_\psi & S_\psi & 0 & 0 \\ -S_\psi & C_\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{12}$$

and the other matrices as

$$M = \text{diag}\left(m_1, m_2, m_3, m_4, m_5, m_6\right),$$

with $m_1 = 0.54 \times 10^{-5}$, $m_2 = 0.248 \times 10^{-5}$, $m_3 = 0.0521$, $m_4 = 0.02$, $m_5 = m_6 = 1$;

$$C = \text{diag}\left(c_1, c_2, c_3, c_4, c_5, c_6\right),$$

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

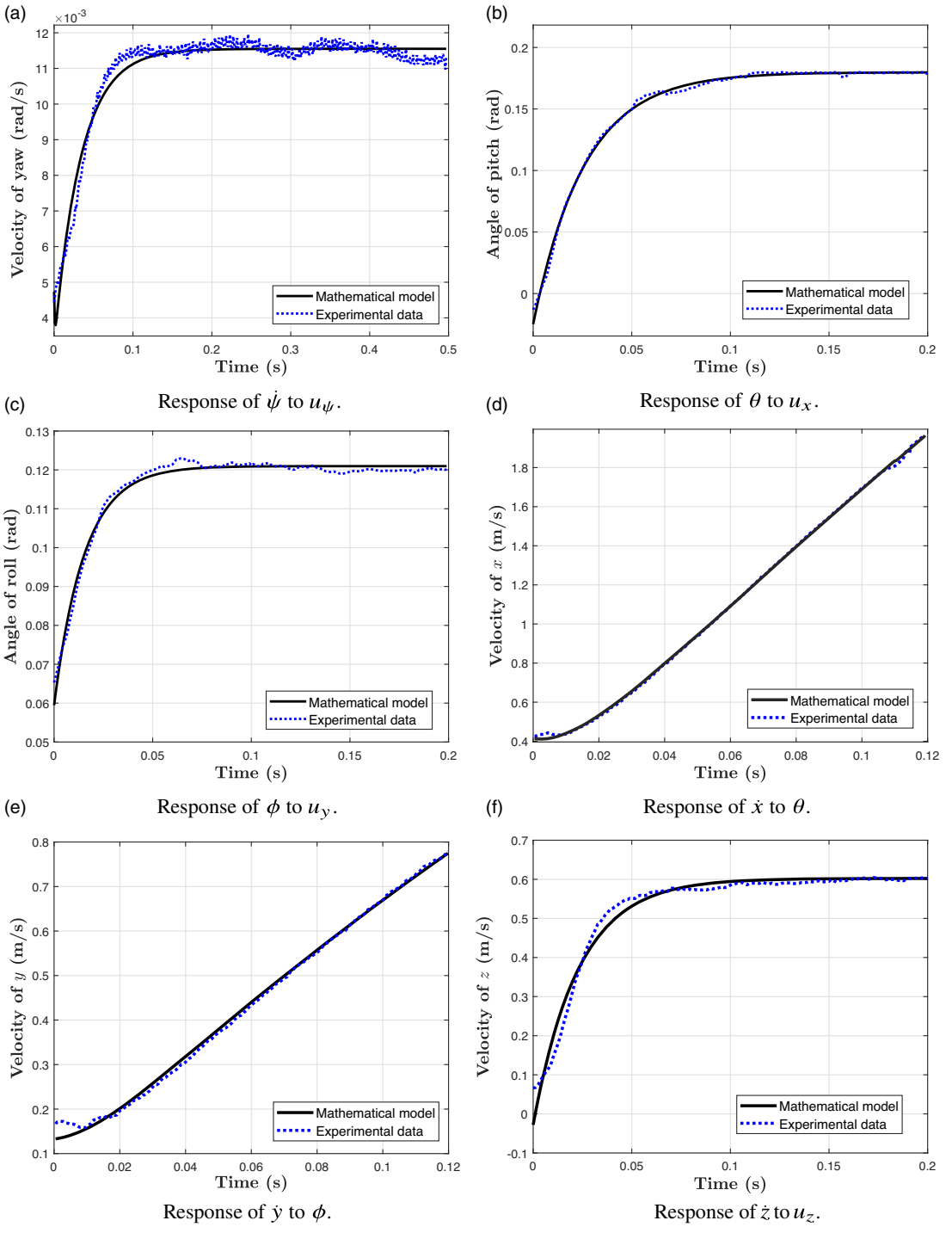


Figure 2. *Identification results: relationship between control inputs and both orientation and translation states.*

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

with $c_1 = 0.14 \times 10^{-2}$, $c_2 = 0.616 \times 10^{-3}$, $c_3 = 1$, $c_4 = 1.085$, $c_5 = 0.753$, $c_6 = 0.478$; finally,

$$D = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ d_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_4 & 0 & 0 & 0 & 0 \end{bmatrix},$$

with $d_1 = 0.048$, $d_2 = 0.036$, $d_3 = -0.7$, $d_4 = -0.0463$.

To apply the IDA-PBC design methodology, the structure of a PCH considers:

$$\bar{p} = \frac{\partial L}{\partial \dot{\bar{q}}} = M\dot{\bar{q}}, \tag{13}$$

where L represents the Lagrangian, which is obtained as the difference between the kinetic and potential energies. Let $K(\bar{p}) = \frac{1}{2}\bar{p}^T M^{-1}\bar{p}$ represent the system's kinetic energy, and $V(\bar{q})$ denote the potential energy. Specifically, in this case, $V(\bar{q}) = mgz$.

The matrix D is related to the potential energy $V(\bar{q})$ through the relationship:

$$D\bar{q} = \frac{\partial V(\bar{q})}{\partial \bar{q}}.$$

The vector $\bar{p}$ is then calculated through (13), resulting in

$$\bar{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} m_1\dot{q}_1 \\ m_2\dot{q}_2 \\ m_3\dot{q}_3 \\ m_4\dot{q}_4 \\ m_5\dot{q}_5 \\ m_6\dot{q}_6 \end{bmatrix} = \begin{bmatrix} m_1\dot{\theta} \\ m_2\dot{\phi} \\ m_3\dot{\psi} \\ m_4\dot{z} \\ m_5\dot{x} \\ m_6\dot{y} \end{bmatrix}.$$

Here, $p_1, p_2, \ldots, p_6$ represent the components of the generalized momenta associated with each coordinate $q_1, q_2, \ldots, q_6$. Thus, the open-loop PCH system is represented as follows:

$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{\bar{p}} \end{bmatrix} = \begin{bmatrix} 0_{6\times6} & I_{6\times6} \\ -I_{6\times6} & 0_{6\times6} \end{bmatrix} \begin{bmatrix} \nabla_{\bar{q}} H \\ \nabla_{\bar{p}} H \end{bmatrix} + \begin{bmatrix} 0_{6\times4} \\ B(\bar{q}) \end{bmatrix} \bar{u} \tag{14}$$

The Hamiltonian H represents the total energy of the system, which is obtained by adding the kinetic energy and the potential energy, that is, $H(\bar{q}, \bar{p}) = K(\bar{p}) + V(\bar{q})$. The terms $\nabla_{\bar{q}} H$ and $\nabla_{\bar{p}} H$ represent the partial derivatives of H with respect to $\bar{q}$ and $\bar{p}$, respectively, that is,

$$\nabla_{\bar{q}} H = \frac{\partial H}{\partial \bar{q}} \quad \text{and} \quad \nabla_{\bar{p}} H = \frac{\partial H}{\partial \bar{p}}.$$

3. Control law design

This section presents the IDA-PBC design methodology.

3.1. IDA-PBC theory

The closed-loop system with the IDA-PBC is represented as [20]:

$$\begin{bmatrix} \dot{\bar{q}} \\ \dot{\bar{p}} \end{bmatrix} = \left[J_d(\bar{q}, \bar{p}) - R_d(\bar{q}, \bar{p}) \right] \begin{bmatrix} \nabla_{\bar{q}} H_d \\ \nabla_{\bar{p}} H_d \end{bmatrix}, \tag{15}$$

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

where the desired interconnection matrix J_d and the desired damping matrix R_d are elements of the structure of the desired PCH dynamics. These matrices are defined as

$$R_d = R_d^T = \begin{bmatrix} 0 & 0 \\ 0 & R_2(\bar{q}, \bar{p}) \end{bmatrix},$$
$$J_d = -J_d^T = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_d M^{-1} & J_2(\bar{q}, \bar{p}) \end{bmatrix},$$

with the antisymmetric matrix $J_2(\bar{q}, \bar{p})$ and $R_2(\bar{q}, \bar{p}) = BK_vB^T$ where $K_v = K_v{}^T > 0$.

Additionally, the desired Hamiltonian energy function H_d is given by

$$H_d = \frac{1}{2}\bar{p}^T M_d^{-1}\bar{p} + V_d(\bar{q}), \tag{16}$$

where $M_d = M_d^T$ represents the desired closed-loop inertia matrix, and V_d is the desired potential energy function, which has an isolated minimum at $\bar{q}_* = \arg\min V_d(\bar{q})$.

The control law that modifies the system from the open-loop structure (14) to the closed-loop structure (15) is expressed as

$$\bar{u} = u_{es}(\bar{q}, \bar{p}) + u_{di}(\bar{q}, \bar{p}), \tag{17}$$

where u_{es} is the energy shaping component and u_{di} corresponds to the damping injection component. Then, in order to add damping into the system, we select the second term of (17) as

$$u_{di}(\bar{q}, \bar{p}) = -K_v B^T \nabla_{\bar{p}} H_d. \tag{18}$$

Now, to obtain u_{es}, we replace (17) and (18) in (14) and equate it with (15):

$$\begin{bmatrix} 0_{6\times6} & I_{6\times6} \\ -I_{6\times6} & 0_{6\times6} \end{bmatrix} \begin{bmatrix} \nabla_{\bar{q}} H \\ \nabla_{\bar{p}} H \end{bmatrix} + \begin{bmatrix} 0_{6\times4} \\ B(\bar{q}) \end{bmatrix} u_{es} = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_d M^{-1} & J_2(\bar{q}, \bar{p}) \end{bmatrix} \begin{bmatrix} \nabla_{\bar{q}} H_d \\ \nabla_{\bar{p}} H_d \end{bmatrix}$$

Thus, the first row of the aforementioned equations is clearly satisfied, and the second row can be expressed as

$$\begin{aligned} Bu_{es}(\bar{q}, \bar{p}) &= \nabla_{\bar{q}} H - M_d M^{-1} \nabla_{\bar{q}} H_d + J_2 M_d^{-1} \bar{p} \\ u_{es}(\bar{q}, \bar{p}) &= (B^T B)^{-1} B^T (\nabla_{\bar{q}} H - M_d M^{-1} \nabla_{\bar{q}} H_d + J_2 M_d^{-1} \bar{p}). \end{aligned} \tag{19}$$

The matrix B is not invertible, since the system is underactuated. Then, u_{es} can only affect the terms in the range space of matrix B. Thus, this causes the following set of constraint equations, which must be satisfied for any selection of energy shaping component u_{es}:

$$B^{\perp} \left\{ \nabla_{\bar{q}} H - M_d M^{-1} \nabla_{\bar{q}} H_d + J_2 M_d^{-1} \bar{p} \right\} = 0, \tag{20}$$

where $B^{\perp}$ is the full-rank annihilator satisfying $B^{\perp}B = 0$. The expression (20) is essentially a PDE in terms of the generalized coordinates and momenta $\bar{q}$ and $\bar{p}$. Thus, introducing $H = \frac{1}{2}\bar{p}^T M^{-1}\bar{p} + V(\bar{q})$ and $H_d = \frac{1}{2}\bar{p}^T M_d^{-1}\bar{p} + V_d(\bar{q})$, we obtain

$$\begin{aligned} &B^{\perp} \left(\nabla_{\bar{q}} V(\bar{q}) - M_d M^{-1} \nabla_{\bar{q}} V_d(\bar{q}) \right) \\ &\quad + B^{\perp} \left(\nabla_{\bar{q}} \left(\frac{1}{2}\bar{p}^T M^{-1} \bar{p} \right) - M_d M^{-1} \nabla_{\bar{q}} \left(\frac{1}{2}\bar{p}^T M_d^{-1} \bar{p} \right) + J_2 M_d^{-1} \bar{p} \right) = 0 \end{aligned}$$

which can be separated into

$$\begin{aligned} B^{\perp} \left[\nabla_{\bar{q}} \left(\bar{p}^T M^{-1} \bar{p} \right) - M_d M^{-1} \nabla_{\bar{q}} \left(\bar{p}^T M_d^{-1} \bar{p} \right) + 2 J_2 M_d^{-1} \bar{p} \right] = 0, \\ B^{\perp} \left[\nabla_{\bar{q}} V - M_d M^{-1} \nabla_{\bar{q}} V_d \right] = 0. \end{aligned} \tag{21}$$

The stability of our algorithm is established by the following theorem:

Theorem 3.1. *The system (14) in closed-loop with the control law (17) with (19) and (18), has an asymptotic stable equilibrium point at* $(\bar{q}_*, 0)$ *with the desired Hamiltonian function given by (16).*

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

Proof. The desired total energy function (16) is considered a Lyapunov candidate function for system (15). Then, differentiating (16) along the trajectories of the system (15):

$$\begin{aligned}\dot{H}_d &= [\nabla_{\bar{q}} H_d]^T \dot{\bar{q}} + [\nabla_{\bar{p}} H_d]^T \dot{\bar{p}} \\ &= [\nabla_{\bar{q}} H_d]^T M^{-1} M_d [\nabla_{\bar{p}} H_d] - [\nabla_{\bar{p}} H_d]^T (M_d M^{-1} [\nabla_{\bar{q}} H_d] + B K_v B^T [\nabla_{\bar{p}} H_d]) \\ &= -[\nabla_{\bar{p}} H_d]^T B K_v B^T [\nabla_{\bar{p}} H_d]\end{aligned}$$

since J_d is skew-symmetric and K_v is positive-definite. From the above expression, it can be observed that $\dot{H}_d \leqslant 0$; that is, the stability is preserved for this type of system.

Using LaSalle's invariance principle, all the trajectories of the system (14) converge to the equilibrium point $(\bar{q}_*, 0)$, which implies asymptotic stability.

Let Ω be the set of all points where $\dot{H}_d = 0$. If $\dot{H}_d = 0$, then, $\Omega = \{(\bar{q}, \bar{p}) | \dot{H}_d = 0\}$. Therefore, H_d is constant. Also, $\dot{H}_d = 0$ implies that $\nabla_{\bar{p}} H_d \in (span(B^{\perp})^T)$. If the equilibrium point is $(\bar{q}_*, 0) \in \Omega$ and $V_d(\bar{q}_*) = 0$, then $\dot{H}_d = [\nabla_{\bar{q}} H_d]^T \dot{\bar{q}} = 0$. Thus, $\dot{\bar{q}} = [\nabla_{\bar{p}} H_d] = 0$, which implies $\bar{p} = 0$ or $\nabla_{\bar{p}} H_d = 0$. From the above, it can be concluded that the states remain at equilibrium in both cases. □

3.2. IDA-PBC algorithm

To apply the IDA-PBC methodology, it is necessary to propose the design matrices $B^{\perp}$, M_d, K_v, and J_2. The matrices $B^{\perp}$ and M_d must be introduced to Eq. (21) to obtain the desired potential energy function. Thus, $B^{\perp}$ is proposed as

$$B^{\perp} = \begin{bmatrix} 0_{2\times 4} & I_{2\times 2} \end{bmatrix}. \tag{22}$$

In order to consider that the translational dynamics is related to the attitude dynamics, we propose the following design matrices M_d and J_2:

$$M_d = \text{diag}(m_{d_1}, m_{d_2}, m_{d_3}, m_{d_4}, m_{d_5}, m_{d_6}), \tag{23}$$

where $M_d = M_d^T$ and $m_{d_1} m_{d_2} m_{d_3} m_{d4} m_{d_5} m_{d_6} > 0$.

$$J_2 = \begin{bmatrix} 0_{3\times 3} & A \\ -A^T & 0_{3\times 3} \end{bmatrix}, \quad \text{with} \quad A = \begin{bmatrix} 0 & -a_1 & 0 \\ 0 & 0 & -a_2 \\ 0 & 0 & 0 \end{bmatrix}.$$

where $a_1 \geq 0$ and $a_2 \geq 0$. Finally, the matrix K_v is a diagonal matrix containing the tuning constants, such that

$$K_v = \text{diag}(k_{d_1}, k_{d_2}, k_{d_3}, k_{d_4}) \tag{24}$$

where the tuning constants k_{d1}, k_{d2}, k_{d3} and k_{d4} strictly positive. Introducing (22) and (23) in (21), the following PDEs are obtained:

$$d_3\theta - \frac{m_{d_5}}{m_5}\frac{\partial V_d}{\partial x} = 0; \quad d_4\theta - \frac{m_{d_6}}{m_6}\frac{\partial V_d}{\partial y} = 0.$$

The desired potential energy is obtained by solving the PDEs and has the following form:

$$V_d = \frac{d_3 m_5}{m_{d_5}}\theta\bar{x} + \frac{d_4 m_6}{m_{d_6}}\phi\bar{y} + \frac{1}{2}k_1\bar{\psi}^2 + \frac{1}{2}k_2\bar{z}^2 \tag{25}$$

where the tuning constants k_1 and k_2 are strictly positive and the terms $\bar{x}$, $\bar{y}$, $\bar{\psi}$, and $\bar{z}$ represent the error in their respective coordinates. In this context, the error is calculated as the difference between the desired and current states.

The desired Hamiltonian is composed of the desired kinetic and potential energies. Using Eqs. (16) and (25), the desired Hamiltonian takes the form:

$$H_d = \frac{1}{2}\left(\frac{m_1^2\dot{\theta}^2}{m_{d_1}} + \frac{m_2^2\dot{\phi}^2}{m_{d_2}} + \frac{m_3^2\dot{\psi}^2}{m_{d_3}} + \frac{m_4^2\dot{z}^2}{m_{d_4}} + \frac{m_5^2\dot{x}^2}{m_{d_5}} + \frac{m_6^2\dot{z}^2}{m_{d_6}}\right) + \frac{d_3 m_5\theta\bar{x}}{m_{d_5}} + \frac{d_4 m_6\phi\bar{y}}{m_{d_6}} + \frac{1}{2}k_1\bar{\psi}^2 + \frac{1}{2}k_2\bar{z}^2 \tag{26}$$

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

To compute the final control law (17), the energy-shaping term (19) and the damping injection term (18) are determined:

$$u_{es} = \begin{bmatrix} u_{es_x} \\ u_{es_y} \\ u_{es_\psi} \\ u_{es_z} \end{bmatrix} = B_1^T \begin{bmatrix} d_1\theta + \frac{d_3 m_5 m_{d_1}}{m_1 m_{d_5}} \bar{x} - a_1 m_5 \dot{x} \\ d_2\phi + \frac{d_4 m_6 m_{d_2}}{m_2 m_{d_6}} \bar{y} - a_2 m_6 \dot{y} \\ \frac{k_1 m_{d_3}}{m_3} \bar{\psi} \\ \frac{k_2 m_{d_4}}{m_4} \bar{z} \end{bmatrix}; \tag{27}$$

$$u_{di} = \begin{bmatrix} u_{di_x} \\ u_{di_y} \\ u_{di_\psi} \\ u_{di_z} \end{bmatrix} = B_1^T \begin{bmatrix} -k_{d_1} \frac{m_1}{m_{d_1}} \dot{\theta} \\ -k_{d_2} \frac{m_2}{m_{d_2}} \dot{\phi} \\ -k_{d_3} \frac{m_3}{m_{d_3}} \dot{\psi} \\ -k_{d_4} \frac{m_4}{m_{d_4}} \dot{z} \end{bmatrix}. \tag{28}$$

Finally, the total control input $\bar{u}$ is given by:

$$\bar{u} = \left[u_x, u_y, u_\psi, u_z\right]^T = u_{es} + u_{di} \tag{29}$$

We can observe from (27) and (28) that by means of the appropriate choice of M_d and J_2, the inputs u_ψ and u_z control the variables ψ and z, respectively, while u_x and u_y control the variables θ, x and ϕ, y, respectively. This design links the translational dynamics to the attitude dynamics, providing an effective control scheme tailored for the underactuated nature of the system.

4. Numerical simulations

This section presents the validation of the controller described by (27)–(29) through numerical simulations applied to the mathematical model (11). The control objective is to stabilize the vehicle during trajectory tracking tasks, with the tuning parameters provided in Section 5. The reference positions are defined as $r_d = [x_d, y_d, \psi_d, z_d]^T$ for both the experimental validation test and the numerical simulations. Likewise, the numerical tests begin with the UAV at position (0, 0, 1) m in hover to match the conditions under which the mathematical model was obtained. An important consideration is that the range of the input values to the Parrot Bebop 2 through the SDK is $[-1, 1]$, which is the fraction of the maximum value configured for that variable. Also, it is important to mention that the IDA-PBC control technique does not provide a systematic procedure for tuning the free design parameters and control gains. The gains in Eq. (27) were iteratively adjusted to ensure that the closed-loop system accurately followed the desired trajectory while preserving the PCH structure. Then, the gains in Eq. (28) were iteratively adjusted to minimize steady-state error and overshoot as much as possible, also with careful verification to maintain the integrity of the PCH structure.

4.1. First numerical test

The reference trajectory for this first test is:

$$r_d = \begin{cases} \begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}^T, & t < 4 \\ \begin{bmatrix} 0.5 & 0.0 & 0.0 & 1.0 \end{bmatrix}^T, & 4 \leq t < 14 \\ \begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}^T, & 14 \leq t < 24 \\ \begin{bmatrix} 0.0 & 0.5 & 0.0 & 1.0 \end{bmatrix}^T, & 24 \leq t < 34 \\ \begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}^T, & 34 \leq t < 44 \\ \begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.5 \end{bmatrix}^T, & 44 \leq t < 54 \\ \begin{bmatrix} 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}^T, & 54 \leq t < 64 \end{cases} \tag{30}$$

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

Table I. *Performance indices for the numerical tests.*

Test	**Index**	x	y	z	ψ
1	ISE	0.3533	0.3328	0.1697	0.0
1	IAE	1.1960	1.0426	0.6609	0.0
2	ISE	0.9307	0.7727	0.0018	0.0
2	IAE	6.7142	6.1261	0.3269	0.0
3	ISE	0.2592	0.7453	0.0003	0.0
3	IAE	3.5492	6.0105	0.1308	0.0

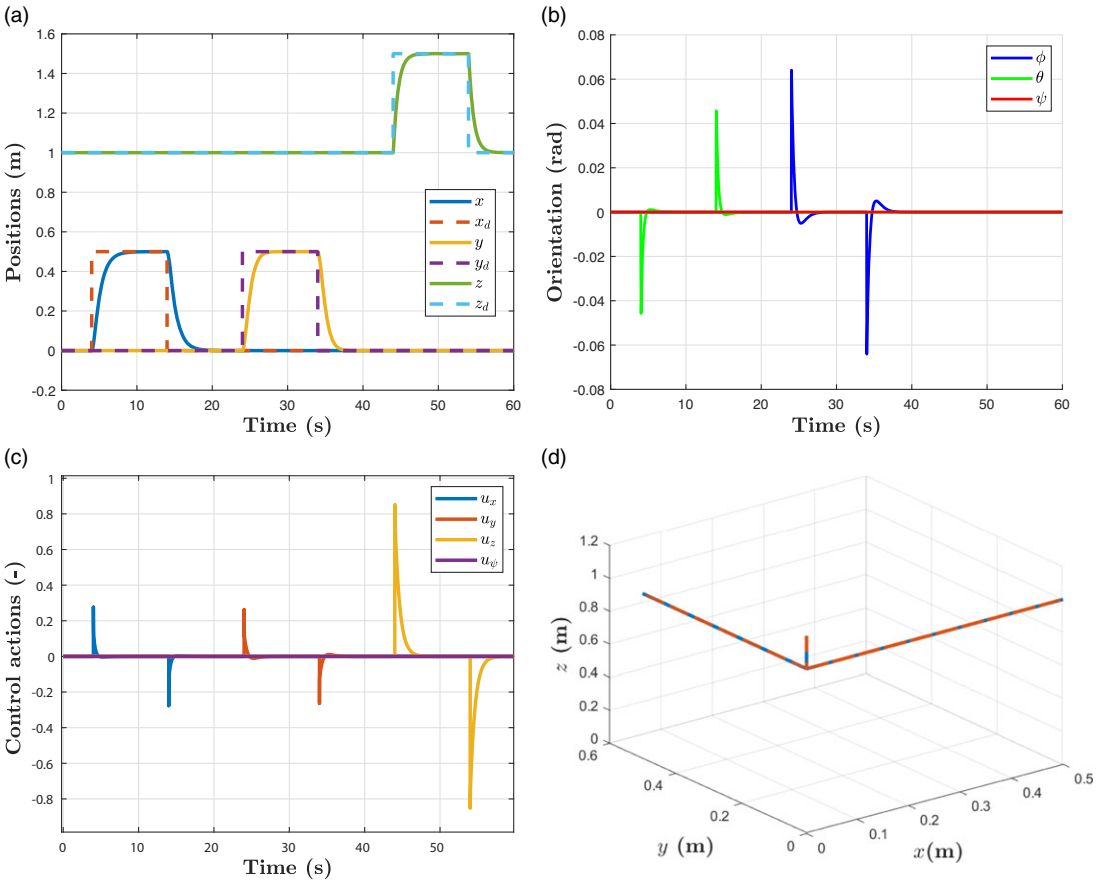


Figure 3. *Simulation results for the first numerical test.*

Figure 3 shows the simulation results for the first test. As observed in Figure 3(c), the control signals u_x, u_y, and u_z reach their maximum values of 0.274 (-), 0.262 (-), and 0.850 (-), respectively, while u_ψ remains at zero (-). The stabilization times for x, y, and z are 4.97 s, 4.1 s, and 3.6 s, respectively, as illustrated in Figure 3(a). The rotation dynamics in Figure 3(b) show small peaks during the reference change. Table I summarizes the performance indices, including the integral square error (ISE) and the integral absolute error (IAE). The IAE values for x and y are slightly above one, indicating that only these indices exceeded the threshold while the others remained relatively low.

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

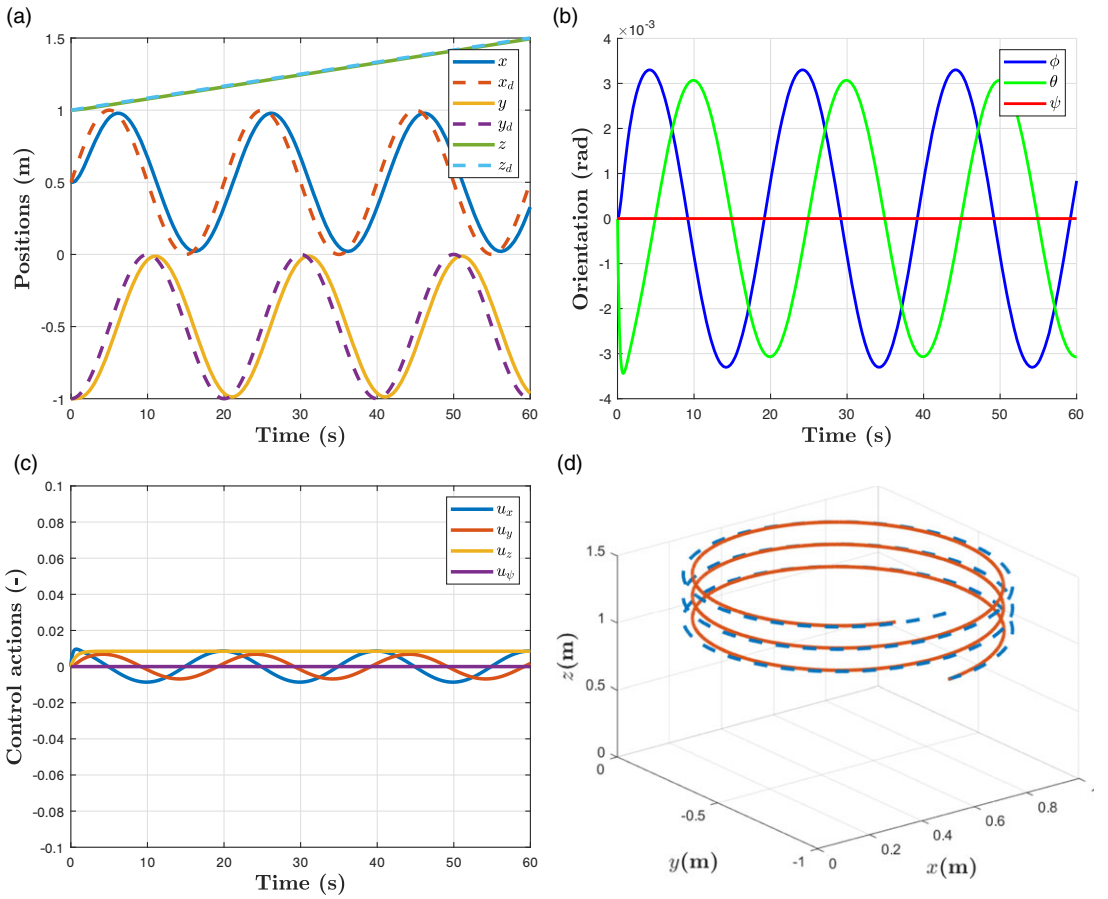


Figure 4. *Simulation results for the second numerical test.*

4.2. Second numerical test

For the second test, the considered reference trajectory is:

$$r_d = \begin{bmatrix} A_x(1+\sin(\omega t)) \\ A_y(1+\cos(\omega t)) \\ \psi_d \\ z_0 + m_z t \end{bmatrix} \tag{31}$$

where $\omega = 0.1\pi$ rad/s, $A_x = 0.5$ m, $A_y = -0.5$ m, $z_0 = 1$ m, $\psi_d = 0$ rad, and $m_z = 0.0083$ m/s.

Figure 4 displays the simulation results for the second test. In Figure 4(c), the control signals u_x and u_y demonstrate smooth behavior, with maximum values of 0.86 (-) and 0.68 (-), respectively, while u_z converges to 0.85 (-) within 3 s. There is a noticeable delay in the tracking of approximately 1 s. The UAV rotation is presented in Figure 4(b). Table I shows the performance indices ISE and IAE.

4.3. Third numerical test

For the third test, the reference trajectory is defined as:

$$r_d = \begin{bmatrix} A_x\left(1+\cos\left(\omega_x t\right)\right) \\ A_y\left(1+\sin\left(\omega_y t\right)\right) \\ \psi_d \\ z_0 + m_z t \end{bmatrix} \tag{32}$$

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

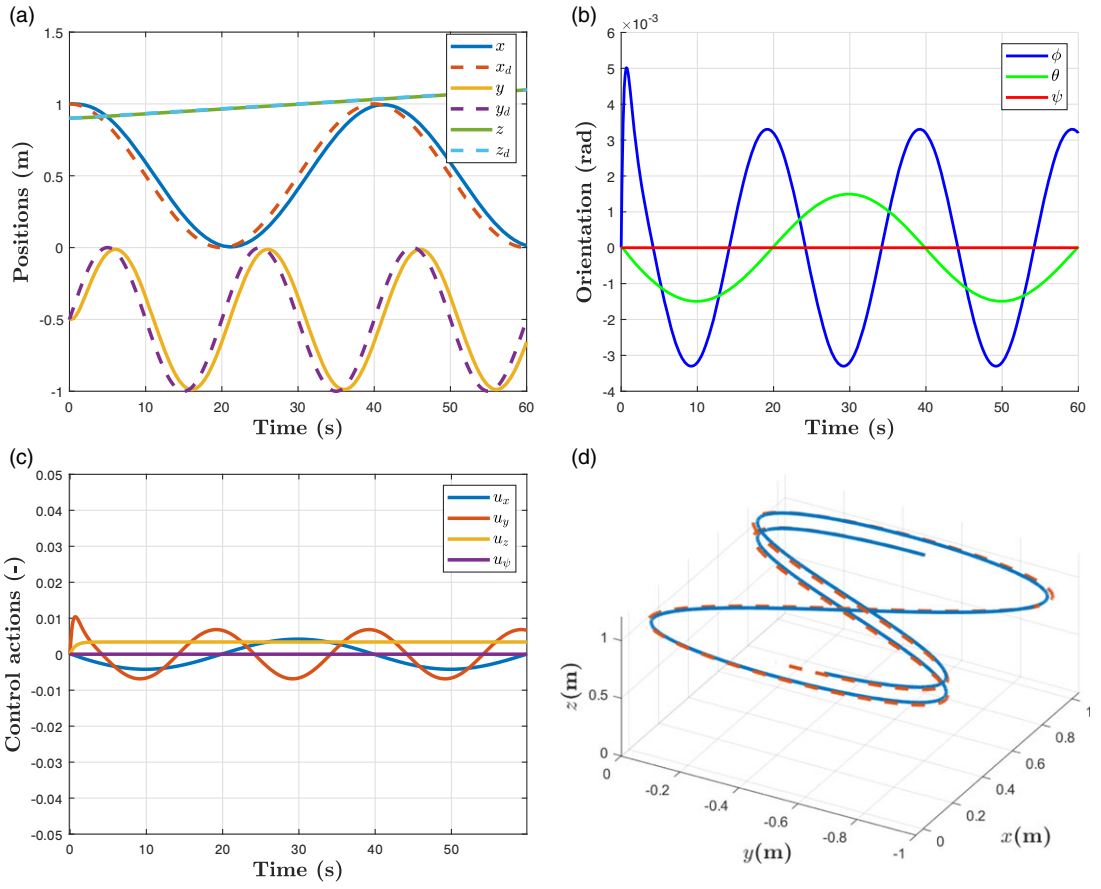


Figure 5. *Simulation results for the third numerical test.*

where $\omega_x = 0.1\pi$ rad/s, $\omega_y = 0.05\pi$ rad/s, $A_x = 0.5$ m, $A_y = 0.5$ m, $z_0 = 0.9$ m, $\psi_d = 0$ rad, and $m_z = 0.0033$ m/s. Similar to the previous test, Figure 5 shows that the UAV rotation and the control actions u_x and u_y exhibit sinusoidal behavior, while the control action for z converges to a fixed value. The maximum values of u_x and u_y are 0.41 (-) and 0 (-). The control action u_z converges to 0.3395 (-) within 2.8 s. The performance indices are shown in Table I.

In the context of these tests, it is clear that the control law successfully stabilizes vehicle dynamics and demonstrates effective performance in trajectory tracking. The performance indices in Table I reveal small values relative to the flight duration, indicating efficient control. Additionally, the vehicle dynamics exhibit rapid yet smooth responses, a result attributed to the damping injection of the IDA-PBC method.

5. Experimental validation

In this section, the UAV's behavior is validated through experimental testing using the designed control law.

5.1. Platform and equipment

The experiments were conducted at the Mobile Robotics Laboratory of the Tecnologico Nacional de Mexico, IT Hermosillo. The flight area and equipment used are shown in Figure 6.

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

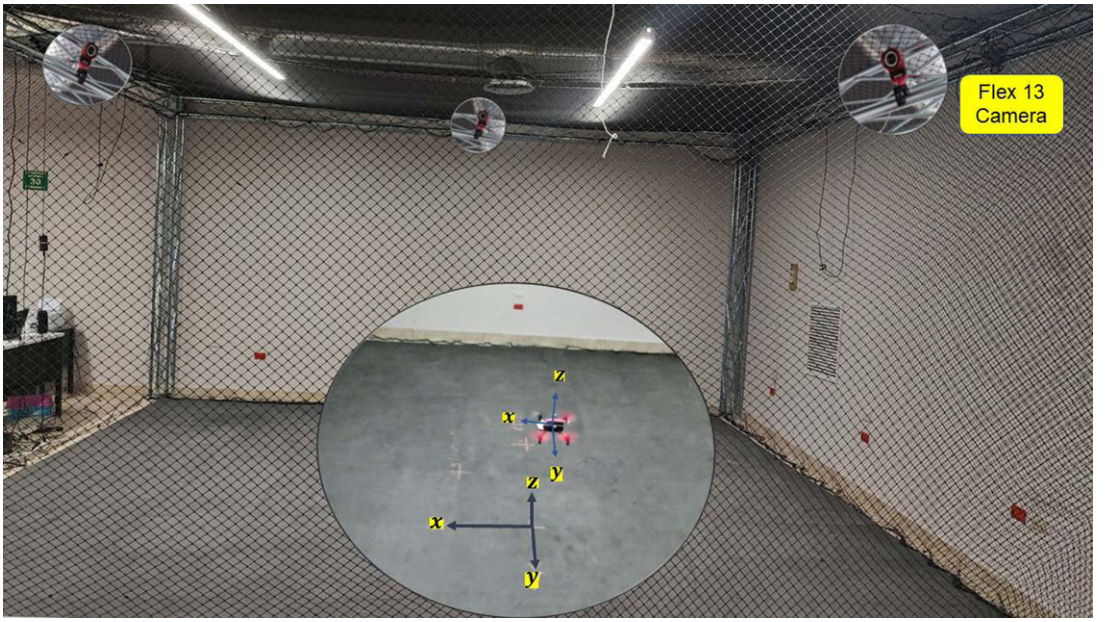


Figure 6. *Flight area and equipment.*

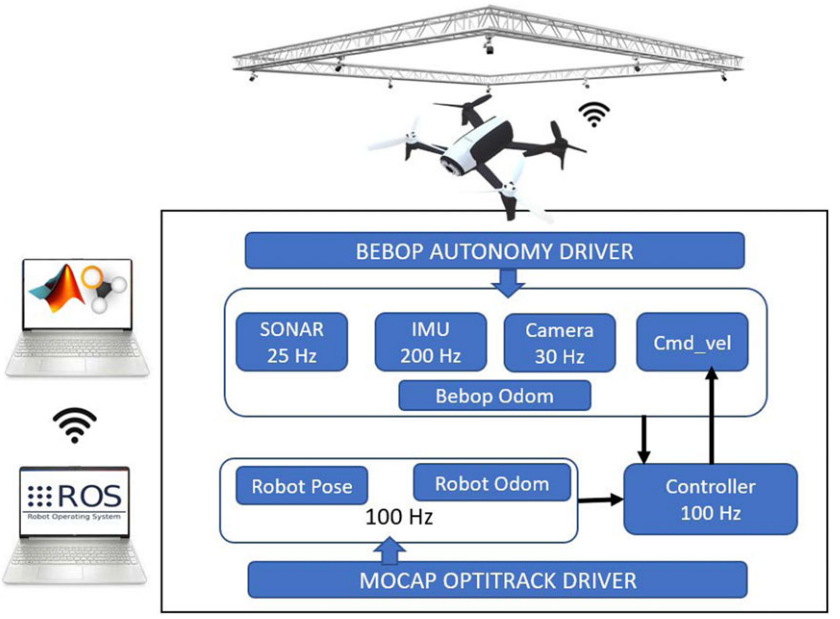


Figure 7. *Experimental platform integration.*

Figure 7 illustrates the equipment integration used to accomplish the required tasks. The diagram highlights the interaction between the environment and the UAV subsystem, with the Robot Operating System (ROS) serving as the integration framework. The experimental platform was built around the commercial UAV Bebop 2, manufactured by Parrot Drone SAS. The motion capture system utilized was the OptiTrack platform, shown in Figure 6. This system comprises eight Flex 13 cameras capable of tracking the pose of objects marked with reflective spheres within a 36 m^2 flight area. Six markers were placed on the UAV for accurate tracking.

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

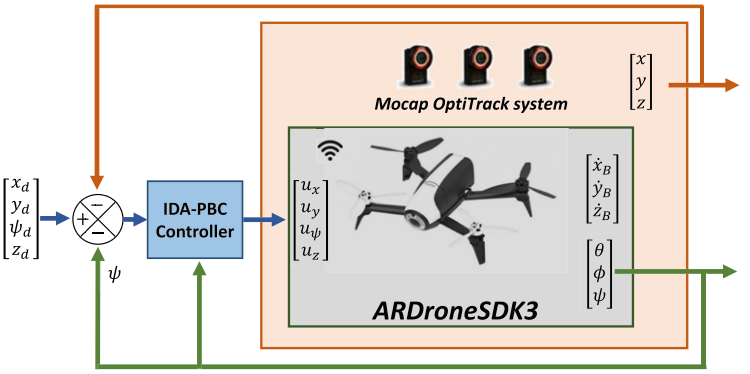


Figure 8. *Experimental platform in closed-loop.*

ROS was installed on a computer running Linux Ubuntu 18.04 to implement and execute the monitoring and control algorithms. The Motive software was used to configure the motion capture system, and the FreeFly Pro app was employed to set the UAV's flight parameters. The main parameters configured for the UAV were a maximum tilt angle of 0.14 rad (8 deg), maximum tilt velocity around the X_B-axis and Y_B- axis of 1.4 rad/s (80 deg/s), maximum vertical velocity of 1 m/s, and maximum turn velocity of 0.227 rad/s (13 deg/s) around the Z_B-axis.

For simplicity, no filters were applied to the sensor data during the experimental tests; all measurements were taken directly from the UAV's onboard sensors and the motion capture system without preprocessing. Once again, the control actions u_x, u_y, u_z, and u_ψ are constrained within the range $[-1, 1]$, where 1 represents 100% of the corresponding configuration parameter and -1 represents -100% of the parameter. The experiment begins with the UAV at position (0, 0, 1) m in hover to match the conditions under which the mathematical model was obtained.

Figure 8 shows the experimental platform operating in closed-loop, displaying both global and local coordinates. The local coordinates are generated and published by the ARDroneSDK3, while the global coordinates are provided by the Mocap OptiTrack system.

5.2. First experimental test

The controller was implemented in ROS, operating at a rate of 100 Hz. A MATLAB/ROS script running at 50 Hz was developed to send reference signals and plot the results. This test aimed to track the desired position as defined in (30).

The experimental results for this test are presented in Figure 9. In Figure 9(a), the control law stabilizes the x and y positions at their desired references within approximately 6 s, while the z position stabilizes in 2 s. To achieve this, the control actions, shown in Figure 9(c), for u_x and u_y reach 0.32 (-), while the control action u_z reaches a maximum of 0.91 (-). Table II shows the performance indices ISE and IAE, which are significantly lower compared to the flight duration.

With this result, it can be validated that a controller under the IDA-PBC methodology presents robustness for unmodeled dynamics since it is clear from Figure 2 that not all the system's dynamic characteristics are captured in the identification process. Also, robustness due to parametric uncertainty is achieved, as the system's real parameters vary based on UAV characteristics such as battery percentage, propeller condition, and additional in-flight load, among others. Despite these variations, the dynamics maintain a stable and smooth behavior.

5.3. Second experimental test

The same reference signals from the second numerical simulation test were used for this case. That is, the reference described in (31) was aimed.

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

Table II. *Performance indices for the experimental tests.*

Test	**Index**	x	y	z	ψ
1	ISE	0.3852	0.3843	0.2424	0.00044
1	IAE	1.6781	1.9173	1.3739	0.13363
2	ISE	0.1765	0.2474	0.0142	0.00023
2	IAE	2.8956	3.4219	0.7531	0.08554
3	ISE	0.0557	0.2469	0.0084	0.00062
3	IAE	1.5406	3.4379	0.5813	0.16791

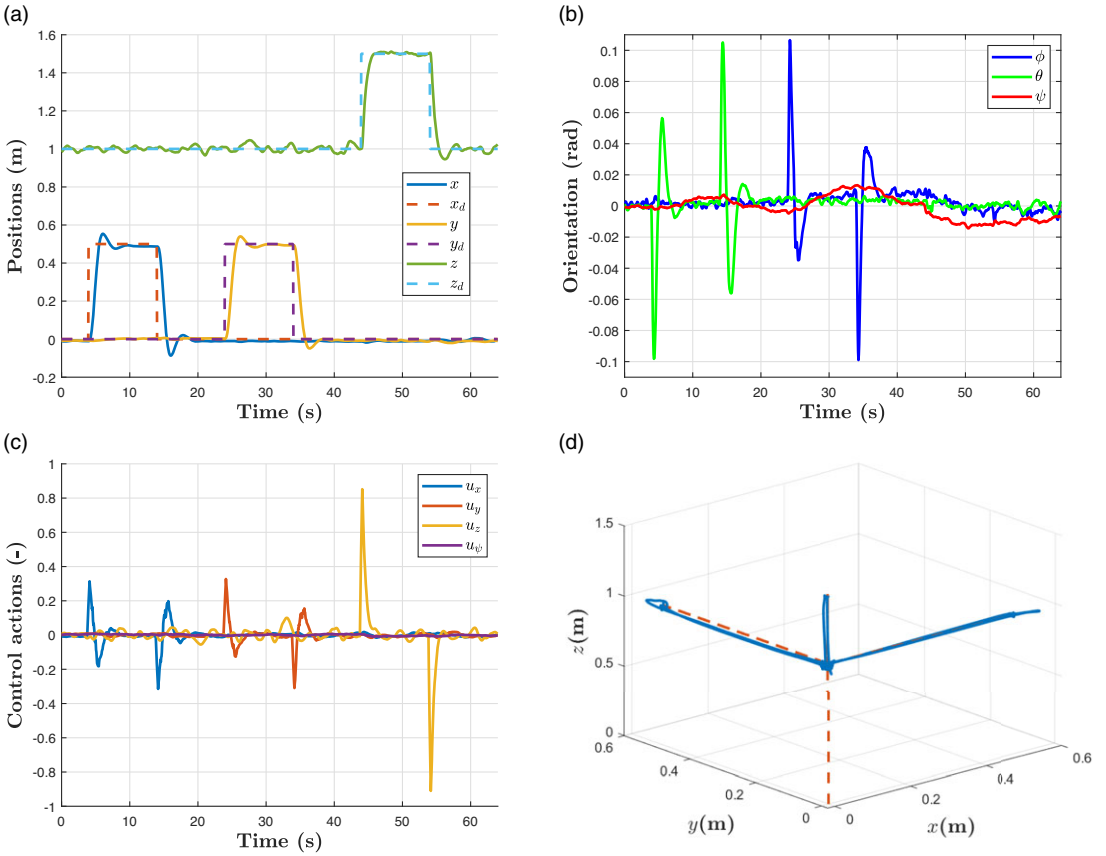


Figure 9. *Results for first experimental test.*

The results of this second test are shown in Figure 10, where it can be seen that the UAV tracks the desired trajectory with a maximum delay of 0.45 s for x and y. Table II presents the performance indices ISE and IAE for this test, which are maintained at an acceptable level.

5.4. Third experimental test

A third experiment was performed, where the vector of the desired states was represented by (32). Figure 11 shows the results of this experiment. As seen in Figure 11(c), the maximum values for the control actions were $u_x = 0.0437$ (-), $u_y = 0.0953$ (-), $u_\psi = 0.48$ (-), and $u_z = 0.0549$ (-). The performance indices for this experiment can be seen in Table II.

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

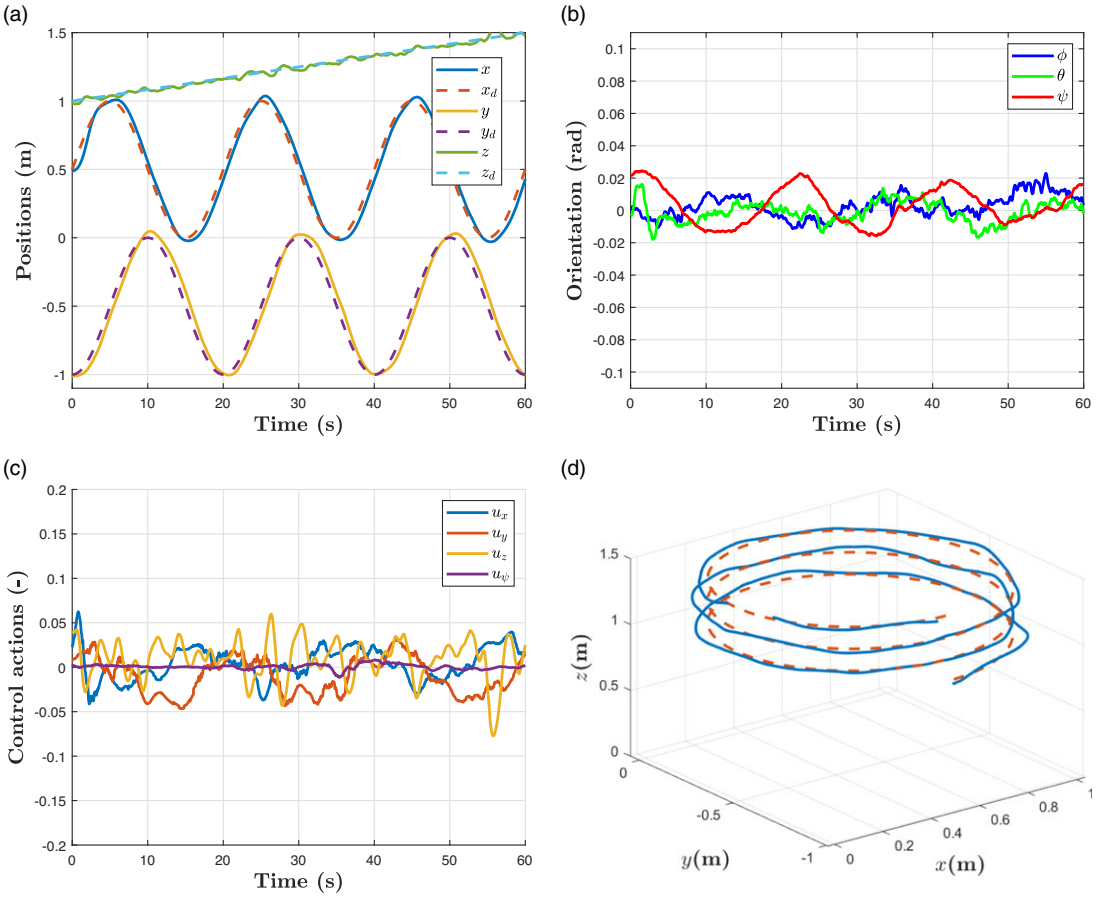


Figure 10. *Results for second experimental test.*

In summary, these experiments demonstrate that the closed-loop system dynamics obtained in the numerical simulation closely match the actual closed-loop dynamics of the vehicle, which validates the identification performed. However, it is important to note that, as shown in Figure 2, the identified model does not fully capture the behavior of the UAV, although it achieves a mean square error lower than 20% for all DOF. Finally, you can watch an example of the experimental validation at the following link: https://www.youtube.com/watch?v=a4jZ_DJcpls.

5.5. Comparison of experimental results with numerical simulation

In the experimental tests, the UAV exhibited stable behavior under the designed control laws, with control actions smoothly converging to their final values and stabilization times closely aligning with theoretical expectations. For example, in the first test, where the UAV tracked trajectory (30), the stabilization times in the experiments were very close to those obtained through simulations. The performance indices also showed good agreement, with the primary difference observed in ψ. In numerical tests, the values of performance indices for ψ were negligible, while in the experiments, they increased to 0.00044 for ISE and 0.13363 for IAE, respectively.

In the second test, where a helical trajectory (31) was tracked, the experimental performance indices were slightly larger compared to the first test. In both cases, the output exhibited a similar delay relative to the reference. However, a notable difference was the presence of oscillations in the experimental results, which were absent in the numerical tests.

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

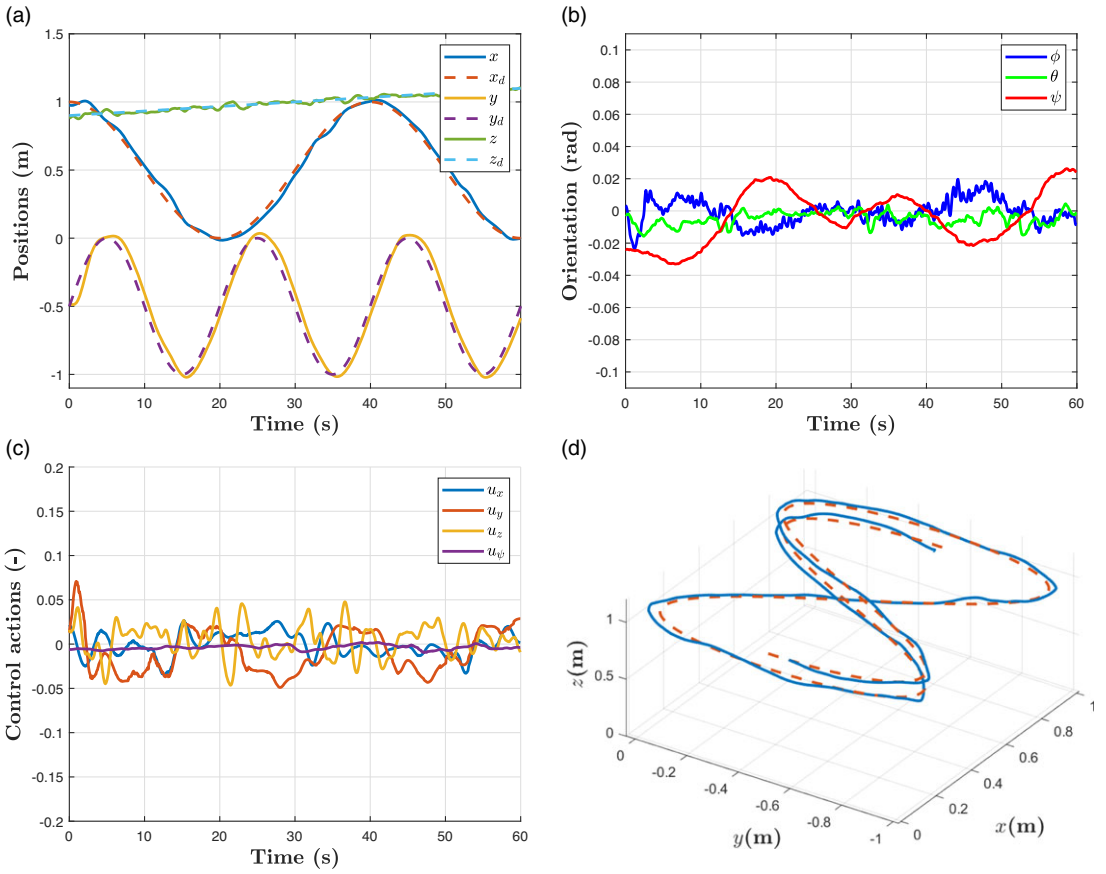


Figure 11. *Results for third experimental test.*

Finally, in the third test, while tracking the Lemniscata trajectory from eq. (32), a significant difference was observed in the behavior of the UAV in z. Figures 5 and 11 reveal small oscillations in the experimental results, leading to a higher IAE performance index. In the numerical tests, the IAE for z was 0.1308, whereas it increased to 0.5816 in the experiments.

Thus, while the numerical tests offered a close approximation of the UAV's behavior, experiments revealed additional complexities, such as unmodeled dynamics and parameter variability that affected performance.

5.6. Comparison with other control techniques

The proposed IDA-PBC algorithm was compared with respect to a sliding mode controller from [19]. The comparison was performed exclusively through experimental tests. Both controllers were evaluated according to the trajectory described in (30). The responses of both controllers are shown in Figures 12 and 13.

In Figure 12, it can be seen that the IDA-PBC algorithm presents shorter stabilization times for the translational position compared to the responses of the SMC strategy. For the position in x, the IDA-PBC controller stabilizes in 5.4 s, while the SMC requires 6.2 s. For the position in y, the IDA-PBC stabilizes in 5.35 s, whereas the SMC stabilizes in 6.7 s. For the position in z, both controllers achieve stabilization in approximately 2 s. Also, we can observe that the IDA-PBC algorithm shows a maximum overshoot of 0.05 m for x and 0.03 m for y, while the SMC does not exhibit overshoot but presents a steady-state error of 0.01 m in both x and y. Thus, the ISE indices for the translational position of the IDA-PBC controller

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

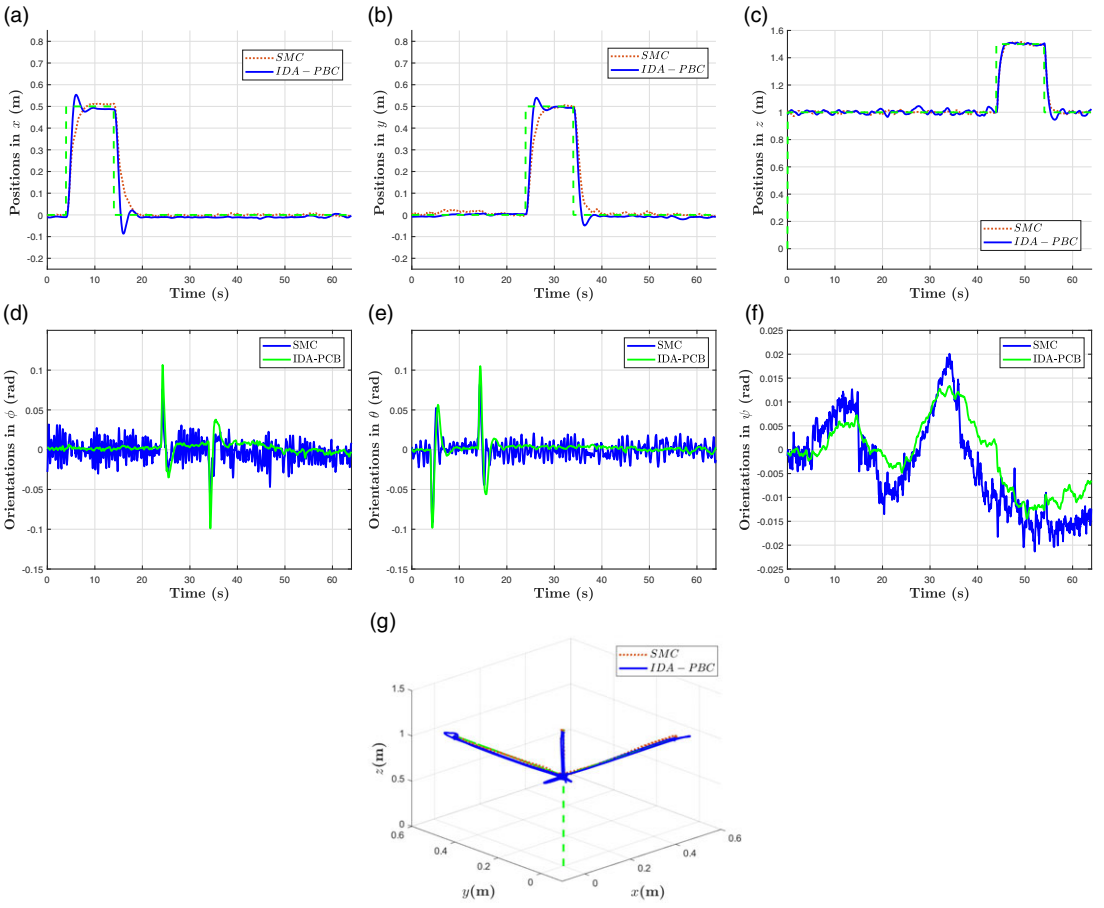


Figure 12. *Comparison results of the translational and rotational motion of the proposed IDA-PBC with the controller of [19].*

are 0.3852, 0.384, and 0.2398, respectively. In contrast, the SMC shows ISE indices of 0.4757, 0.4864, and 0.2373. Regarding vehicle orientation, the IDA-PBC scheme exhibits smooth behavior, while the SMC displays sustained oscillations due to chattering, a well-known characteristic of the SMC strategy.

In Figure 13, it can be observed that the control action of the IDA-PBC presents peaks of up to 0.233 (-) for u_x, 0.2015 (-) for u_y, and 0.6137 (-) for u_z. In contrast, the SMC shows peaks in control actions of up to 0.338 (-) for u_x, 0.3345 (-) for u_y, and 0.8553 (-) for u_z. Additionally, both controllers maintain u_ψ close to 0 (-). The main difference in these responses is the sustained oscillations in the control actions of the SMC for u_x and u_y, with an amplitude of approximately 0.2 (-), leading to increased energy consumption and reduced flight time.

Based on the above, we can mention that the proposed IDA-PBC scheme presents a slightly superior performance over the one in [19].

5.7. Practical implementation challenges

During the physical implementation of the IDA-PBC controller, several practical challenges were encountered that were not present in the numerical simulations. One significant issue was sensor noise, which introduced inaccuracies in state measurements. Additionally, unmodeled disturbances, such as minor vibrations, environmental variations, and platform-specific dynamics, impacted system

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

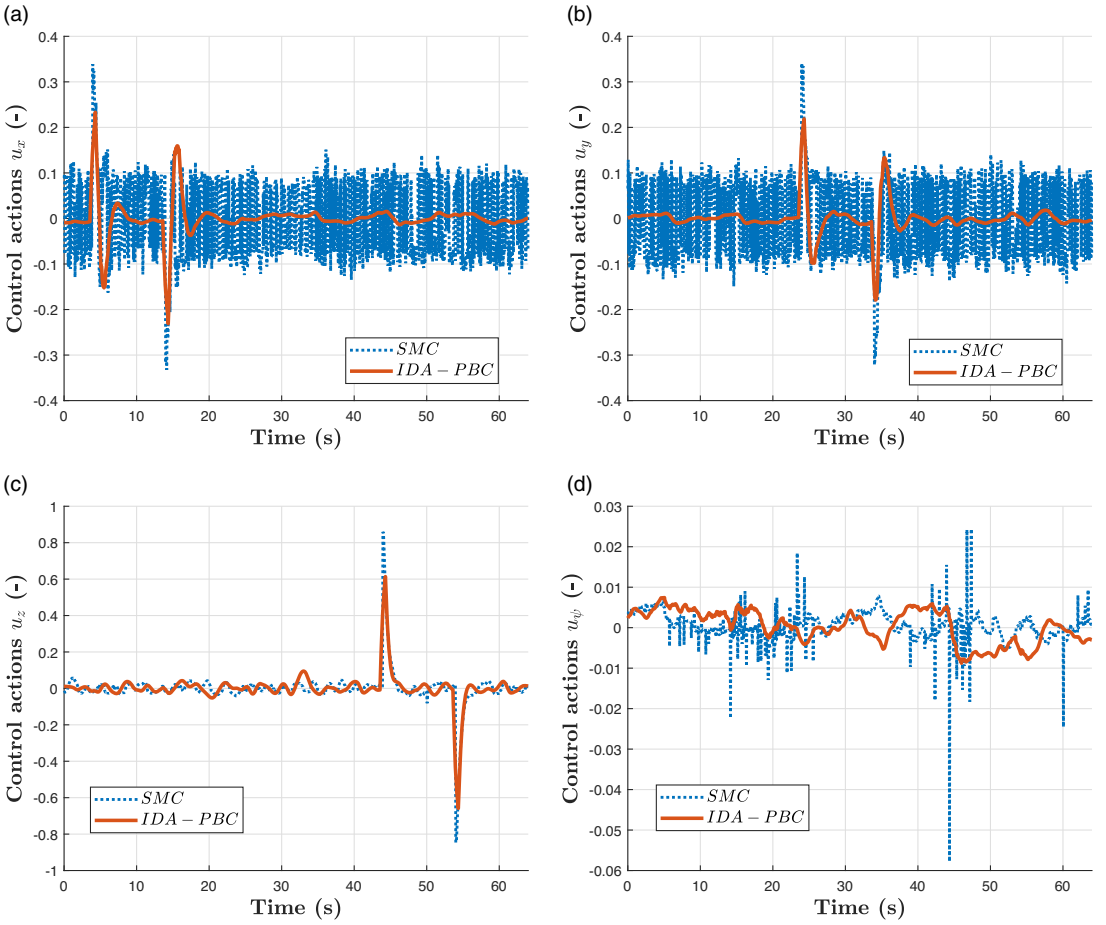


Figure 13. *Comparison results of the control laws of the proposed IDA-PBC with the controller of [19].*

performance. Parametric uncertainties, caused by factors like battery consumption, motor wear, and unaccounted payload variations, further influenced the UAV's behavior.

Despite these real-world challenges, the IDA-PBC controller demonstrated robust performance. Its energy-based design, particularly the damping injection mechanism, helped mitigate the effects of disturbances by ensuring asymptotic stability. Furthermore, the passivity-based framework provided robustness to parametric uncertainties, preserving the stability of the system even when deviations from the nominal model occurred.

In comparison to simulations, the experimental results showed slightly larger trajectory tracking errors and occasional oscillations. These deviations were primarily due to sensor noise and the practical limitations of the UAV's onboard actuators. However, the controller consistently maintained stability and achieved accurate trajectory tracking, validating its effectiveness in handling real-world conditions.

In summary, the experiments highlighted additional complexities, such as unmodeled dynamics and parameter variability, that affected performance. Nevertheless, the robustness observed in the experiments demonstrates the practical applicability of the IDA-PBC framework for UAV control.

6. Conclusions

This work presented the design and experimental validation of an IDA-PBC-based flight control strategy for the Parrot Bebop 2 quadrotor UAV using a Hamiltonian model derived from experimental data.

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

The proposed controller effectively addressed the challenges associated with modeling and controlling UAVs, demonstrating robust stabilization and precise trajectory tracking. Numerical simulations and experimental testing in an indoor flight setup demonstrated that the control approach achieved smooth and stable performance, even in the presence of unmodeled dynamics and parametric uncertainties. Moreover, the proposed controller's structure is relatively simple. For its implementation, it is required to supply the controller with a number of offline-designed constant gains that result in a fixed number of floating-point operations to compute the control law. Thus, the controller has no significant limitations from the computational efficiency perspective. The proposed IDA-PBC control was compared with respect to a sliding mode controller from [19]; in this comparison, the IDA-PBC presented a slightly superior performance over the one in [19]. Future work will focus on extending the validation to outdoor environments with more complex disturbances and to scenarios involving multi-agent UAV systems, which pose scalability and communication challenges. Additionally, integrating advanced state estimation techniques for handling noisy sensor data is a promising direction for enhancing the scalability and reliability of the proposed methodology. Furthermore, advanced modeling approaches, such as stochastic or robust identification techniques, to incorporate uncertainty and disturbances during system identification will be explored.

Acknowledgments. The first author thanks A.M. Bonilla-Huesca, J.A. Zepeda-Bojórquez, and A. Corral-Sánchez for their support.

Author contributions. JRMM, OHG, and LCFH performed the experiments and wrote the first draft of the article. MEGS and GVP conceived and designed the study. OHG and FRLE analyzed the results. MEGS, GVP, and FRLE wrote and edited the final version of the paper.

Financial support. This research was funded by Tecnológico Nacional de México under the program *Proyectos de Investigación Cientfica, Desarrollo Tecnológico e Innovación* and the international network *Red Internacional de Control y Cómputo Aplicado*. Additional funding was provided by SECIHTI under the program *Investigadoras e Investigadores por México*, Project 88. The work of J.R. Montoya-Morales was supported by SECIHTI through a doctoral fellowship.

Competing interests. The authors declare no conflicts of interest.

References

[1] K. Ahmadi, D. Asadi, A. Merheb, S.-Y. Nabavi-Chashmi and O. Tutsoy, "Active fault-tolerant control of quadrotor uavs with nonlinear observer-based sliding mode control validated through hardware in the loop experiments," *Control Eng. Pract.* **137**, 105557 (2023).

[2] H. Ahn, M. Hu, Y. Chung and K. You, "Sliding-mode control for flight stability of quadrotor drone using adaptive super-twisting reaching law," *Drones*. **7**(8), 522 (2023).

[3] Z. N. A. Mardlijah, "Control design of quadcopter using linear quadratic gaussian (LQG)," **In:** 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), IEEE (2022) pp. 192–196.

[4] F. W. Alsaade, H. Jahanshahi, Q. Yao, M. S. Al-zahrani and A. S. Alzahrani, "A new neural network-based optimal mixed H_2/H_∞ control for a modified unmanned aerial vehicle subject to control input constraints," *Adv. Space Res.* **71**(9), 3631–3643 (2023).

[5] O. Arrieta, D. Campos, J. Rico-Azagra, M. Gil-Martínez, J. D. Rojas and R. Vilanova, "Model-based optimization approach for PID control of pitch–roll UAV orientation," *Mathematics*. **11**(15), 3390 (2023).

[6] N. Boualem, M. Lotfi, G. Abderrezak, B. Abdelhak, Z. M. Kadda and T. Mohammed, "Metaheuristic optimization techniques of a sliding-mode controller gains applied to quadrotor," **In:** AIP Conference Proceedings, Vol. 2536, AIP Publishing (2023).

[7] J. Cariño, H. Abaunza and P. Castillo, "A fully-actuated quadcopter representation using quaternions," *Int. J. Control*, **96**(12), 3132–3154 (2022). https://doi.org/10.1080/00207179.2022.2129789

[8] A. A. El-Badawy and M. A. Bakr, "Quadcopter aggressive maneuvers along singular configurations: An energy-quaternion based approach," *J. Control Sci. Eng*. **2016**, 1–10 (2016).

[9] J. Engel, J. Sturm and D. Cremers, "Scale-aware navigation of a low-cost quadrocopter with a monocular camera," *Robot. Auton. Syst (RAS)*. **62**(11), 1646–1656 (2014).

[10] C. D. García-Beltrán, E. M. Miranda-Araujo, M. E. Guerrero-Sánchez, G. Valencia-Palomo, O. Hernández-González and S. Gómez-Peñate, "Passivity-based control laws for an unmanned powered parachute aircraft," *Asian J. Control*. **23**(5), 2087–2096 (2021).

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press

[11] D. Gehrig, M. Göttgens, B. Paden and E. Frazzoli, *Scale-Corrected Monocular-Slam for the AR.Drone 2.0* (ETH Zürich, 2017).
[12] M. E. Guerrero-Sanchez, H. Abaunza, P. Castillo, R. Lozano, C. Garcia-Beltran and A. Rodriguez-Palacios, "Passivity-based control for a micro air vehicle using unit quaternions," *Appl. Sci*. **7**(1), 13 (2016).
[13] M. E. Guerrero-Sanchez, H. Abaunza, P. Castillo, R. Lozano and C. D. García-Beltrán, "Quadrotor energy-based control laws: A unit-quaternion approach," *J. Intell. Robot. Syst*. **88**(2–4), 347–377 (2017).
[14] M.-E. Guerrero-Sánchez, O. Hernández-González, G. Valencia-Palomo, F.-R. López-Estrada, A.-E. Rodríguez-Mata and J. Garrido, "Filtered observer-based IDA-PBC control for trajectory tracking of a quadrotor," *IEEE Access*. **9**, 114821–114835 (2021).
[15] M. E. Guerrero-Sánchez, O. Hernández-González, G. Valencia-Palomo, D. A. Mercado-Ravell, F. R. López-Estrada and J. A. Hoyo-Montaño, "Robust IDA-PBC for under-actuated systems with inertia matrix dependent of the unactuated coordinates: Application to a UAV carrying a load," *Nonlinear Dynam*. **105**(4), 3225–3238 (2021).
[16] C. Izaguirre-Espinosa, A. J. Muñoz-Vázquez, A. Sánchez-Orta, V. Parra-Vega and P. Castillo, "Attitude control of quadrotors based on fractional sliding modes: Theory and experiments," *IET Control Theory Appl*. **10**(7), 825–832 (2016).
[17] W. Jung and H. Bang, "Fault and failure tolerant model predictive control of quadrotor UAV," *Int. J. Aeronaut. Space*. **22**(3), 663–675 (2021).
[18] K. B. Kidambi, C. Fermüller, Y. Aloimonos and H. Xu, "Robust nonlinear control-based trajectory tracking for quadrotors under uncertainty," *IEEE Control Syst. Lett*. **5**(6), 2042–2047 (2020).
[19] J. R. Montoya-Morales, M. E. Guerrero-Sánchez, G. Valencia-Palomo, O. Hernández-González, F. R. López-Estrada and J. A. Hoyo-Montaño, "Real-time robust tracking control for a quadrotor using monocular vision," **In:** Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering (2023) pp. 09544100231158265.
[20] R. Ortega, M. W. Spong, F. Gómez-Estern and G. Blankenstein, "Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment," *IEEE Trans. Autom. Control*. **47**(8), 1218–1233 (2002).
[21] M. A. Rosaldo-Serrano and E. Aranda-Bricaire, "Trajectory tracking for a commercial quadrotor via time-varying backstepping," *IFAC-PapersOnLine*. **51**(13), 532–536 (2018).
[22] M. A. Rosaldo-Serrano and E. Aranda-Bricaire, "Trajectory tracking for a commercial quadrotor," **In:** 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), IEEE (2019) pp. 1616–1621.
[23] M. Ryalat, D. S. Laila and M. M. Torbati, "Integral IDA-PBC and PID-like control for port-controlled Hamiltonian systems," **In:** 2015 American Control Conference (ACC), IEEE (2015) pp. 5365–5370.
[24] J. Santiaguillo-Salinas, M. A. Rosaldo-Serrano and E. Aranda-Bricaire, "Observer-based time-varying backstepping control for parrot's AR.Drone 2.0," *IFAC-PapersOnLine*. **50**(1), 10305–10310 (2017).
[25] C. Souza, G. V. Raffo and E. B. Castelan, "Passivity based control of a quadrotor UAV," *IFAC Proceedings*. **47**(3), 3196–3201 (2014).
[26] R. Thusoo, S. Jain and S. Bangia, "Path planning of quadrotor using A$*$ and LQR," **In:** Recent Developments in Electrical and Electronics Engineering: Select Proceedings of ICRDEEE 2022, Springer (2023) pp. 227–238.
[27] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles* (Vol. **1**, Springer, 2015).
[28] D. K. D. Villa, A. S. Brandão, R. Carelli and M. Sarcinelli-Filho, "Cooperative load transportation with two quadrotors using adaptive control," *IEEE Access* **9**, 129148–129160 (2021).
[29] L. von Stumberg, V. Usenko, J. Engel, J. Stückler and D. Cremers, "From monocular slam to autonomous drone exploration," **In:** 2017 European Conference on Mobile Robots (ECMR), IEEE (2017) pp. 1–8.
[30] X. Wang, S. Sun, E.-J. van Kampen and Q. Chu, "Quadrotor fault tolerant incremental sliding mode control driven by sliding mode disturbance observers," *Aerosp. Sci. Technol*. **87**, 417–430 (2019).
[31] L. Zhao, Y. Liu, Q. Peng and L. Zhao, "A dual aircraft maneuver formation controller for MAV/UAV based on the hybrid intelligent agent," *Drones*. **7**(5), 282 (2023).

Cite this article: J.-R. Montoya-Morales, M.-E. Guerrero-Sánchez, G. Valencia-Palomo, O. Hernández-González, F.-R. López-Estrada and L. C. Félix-Herrán (2025). "Design and experimental validation of IDA-PBC-based flight control for quadrotors", Robotica **43**, 2376–2397. https://doi.org/10.1017/S0263574725101719

https://doi.org/10.1017/S0263574725101719 Published online by Cambridge University Press