Fast Aerial Manipulation of a Varying Payload

Tariel Simonyan * and Oleg Gasparyan

Posted Date: 12 May 2026

doi: 10.20944/preprints202605.0750.v1

Keywords: gain-scheduling; unmanned aerial vehicles; aerial manipulation; intelligent control system

 doi:10.20944/preprints202605.0750.v1

Article

Fast Aerial Manipulation of a Varying Payload

Tariel Simonyan * and Oleg Gasparyan

National Polytechnic University of Armenia, Armenia

* Correspondence: tariel.simonyan@polytechnic.am

Abstract

This paper addresses the robust trajectory tracking problem of an Unmanned Aerial Vehicle (UAV) equipped with a 2-DOF manipulator, designed for fast aerial manipulation of varying payloads. To overcome the high computational cost and adaptability limitations of traditional model-based controllers, this work introduces a novel hybrid gain-scheduling framework that shifts the computational complexity to the pre-flight phase. The approach utilizes an approximate inverse dynamics linearization, based on fixed nominal models, which transforms the complex nonlinear system into a simple linear plant with bounded, structured uncertainties. The entire configuration space, including manipulator states and a range of payload properties, is partitioned into dynamically similar regions using K-Means clustering. For each local region, a dedicated robust PD controller is designed using a multi-objective Genetic Algorithm (GA). This framework also successfully implements a gain interpolation technique to mitigate the potential for abrupt control actions. Simulation results validate the controller's ability to maintain high-precision tracking during fast maneuvers and payload switching, confirming the robustness and adaptability of the offline-tuned design.

Keywords: gain-scheduling; unmanned aerial vehicles; aerial manipulation; intelligent control system

1. Introduction

The aerial manipulator is a new type of aerial system that can both fly and physically interact with the world. Aerial robots get so much interest for their increased mobility compared to ground robots as they are not restricted by terrains and can navigate in hard-to-access locations [1]. These systems offer solutions in the environment which can be too dangerous for a human operator. Aerial manipulators are used in diverse fields like military, inspection, transportation, architecture, building, and construction.

Nevertheless, the integration of a robotic arm with a floating base presents significant control difficulties. The system is subject to complex, time-varying dynamics coupling that results in strong nonlinearities. The motion of the manipulator and the interaction with varying payload continuously alter the system's inertia tensor and shifts the center of mass (CoM) during flight. Consequently, achieving robustly stable and precise control remains a complex problem.

Current control literature is full of different control methods for those systems, proposed to manage the coupled dynamics include nonlinear, adaptive controllers and sophisticated compensation schemes. Despite these advances, existing control systems are still facing some limitations. To achieve robust stability, many designs, including robust optimal methods like the classic nonlinear H∞ controller [2], rely on the explicit, continuous real-time calculation and compensation for the reaction forces induced by the robotic arm [3,4]. The model-based solutions, derived from advanced techniques using numerical methods for dynamic compensation [5] are increasing the computational cost and restricting the performance of those platforms [6]. Complex solutions, such as combining sliding mode control with adaptive laws and disturbance observers for simultaneous internal and external disturbance compensation [7,8], or adaptive control methods

 doi:10.20944/preprints202605.0750.v1

combined with neural networks to manage high modeling errors [8,9], are promising but result in a large computational burden that can challenge the system's stability during fast aerial manipulation maneuvers.

Control stability is often dependent on strictly defined mechanical parameters. For example, stability conditions often rely on critical assumption such as keeping the mass of the manipulator sufficiently small with respect to the aerial vehicle [4] or require complex algorithms to estimate time-varying inertia when manipulating objects [10]. These inherent structures are not robust to the changes introduced by grasping different payloads [8,11]. While robust stability can be analytically proved using complex frameworks like nested saturation control (achieving input-to-state stability) [3,4], these systems are designed with controller structures that are difficult to tune across significantly varying flight conditions.

Based on preceding analysis, the central research problem involves the trade-off between the high computational cost of real-time nonlinear compensation and the need for robust stability during fast maneuvers with varying payloads. While current control methods for aerial manipulation can achieve stability, they typically rely on real-time calculations that create a significant computational burden, or they lack the necessary robustness to handle rapid payload changes and fast maneuvers simultaneously.

To address this problem, the research objectives are as follows:

i. To minimize the real-time computational cost associated with nonlinear dynamic compensation.
ii. To enhance the adaptability of the control system to rapid shifts in payload characteristics and system inertia during fast maneuvers.
iii. To guarantee high-precision trajectory tracking and robust stability throughout the whole operational envelope.

The proposed solution is based on shifting the computational complexity to the pre-flight phase. By employing an inverse dynamics linearization [12] based on a fixed, pre-computed nominal models, the complex nonlinear system was transformed into a simple linear system with residual nonlinearities captured as bounded uncertainties. Those structured uncertainties are quantified using Linear Fractional Transformation (LFT) framework [13]. The entire configuration space, including the manipulator state, the payload mass and the object dimensions, is divided into clusters, using K-Means algorithm optimized via the Elbow Method [14]. For each local region, a PD controller is designed, and its gains are tuned using a multi-objective GA [15]. The reward function is tasked to minimize the tracking error and the control effort, while maximizing the local robust stability margin, using the Small Gain Theorem [13]. This method yields a control system that can tolerate grasping of various payloads in the defined range of object masses and dimensions.

2. System Dynamics

The aerial platform consists of a quadrotor base carrying a 2-DOF manipulator with two revolute joints as depicted in Figure 1. The kinematic description of the system is based on two coordinate systems: $\{I\}$ Inertial frame and $\{B\}$ body-fixed frame aligned with the principal axes of inertia. The transition from $\{I\}$ to $\{B\}$ is given by the orthogonal rotation matrix constructed from the UAV's Euler angles in the ZYX convention. The system state is defined using a generalized coordinate vector $q \in \mathrm{R}^6$, which consists of UAV's height z in $\{I\}$, its attitude vector (roll, pitch, yaw) $\Omega = [\phi, \theta, \psi]^T$ in $\{B\}$, and the manipulator's joint angles $\Upsilon = [\varepsilon_1, \varepsilon_2]^T$:

$$q = [z, \phi, \theta, \psi, \varepsilon_1, \varepsilon_2]^T, \quad (1)$$

Unlike conventional UAVs, the dynamics of an aerial manipulator are intrinsically coupled to the arm's motion, which displaces the CoM and alters the vehicle's inertia tensor during the flight. The inputs driving the system are denoted as τ and combine the quadrotor's aerodynamics with the servo torques:

 doi:10.20944/preprints202605.0750.v1

$$\tau = [T_\Sigma, \tau_\phi, \tau_\theta, \tau_\psi, \tau_{j1}, \tau_{j2}]^T, \quad (2)$$

where T_Σ represents the collective thrust of the four rotors:

$$T_\Sigma = \sum_{i=1}^{4} T_i, \quad (3)$$

where each i-th rotor generates a thrust T_i which is proportional to the square of angular velocity of rotors σ_i (i.e. $T_i = c_T \sigma_i^2$, $c_T > 0$) and acts along the body-fixed axis Z_B [16,17]. The $\tau_\phi, \tau_\theta, \tau_\psi$ are the quadrotor torques and τ_{j1}, τ_{j2} are the torques of each joint of the manipulator. The thrust to torque mapping for the quadrotor can be expressed using the 4-dimensional vector of thrusts T_i, ($\bar{T}_{UAV} = [T_1, T_2, T_3, T_4]^T$), and the quadrotor configuration matrix B_{UAV}:

$$\begin{bmatrix} T_\Sigma \\ \tau_{UAV} \end{bmatrix} = B_{UAV} \bar{T}_{UAV}. \quad (4)$$

Given the needed controls T_Σ and τ, the equation (4) allows computing the required thrusts T_i (or, which is equivalent, the velocities σ_i) of rotors. System's torque to input mapping can be described as $\tau = B_{SYS} \bar{T}_{SYS}$, where $\bar{T}_{SYS}$ is the combined vector of the quadrotor motors thrusts and the manipulator joint torques. B_{SYS} for the X configuration quadrotor with 2-DOF robotic arm will have the form:

$$B_{SYS} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ \sqrt{2}L/2 & \sqrt{2}L/2 & -\sqrt{2}L/2 & -\sqrt{2}L/2 & 0 & 0 \\ -\sqrt{2}L/2 & \sqrt{2}L/2 & \sqrt{2}L/2 & -\sqrt{2}L/2 & 0 & 0 \\ -k_\psi & k_\psi & -k_\psi & k_\psi & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

where L is the length of the quadrotor arms, and k_ψ ($k_\psi > 0$) is the drag coefficient.

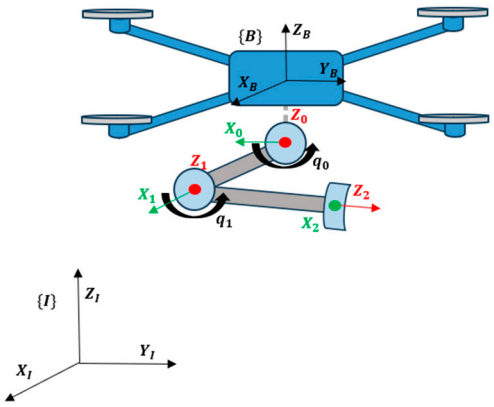

Figure 1. UAV-manipulator system

The equations of motion of the system are derived using Euler-Lagrange method [18], have the standard form:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_D. \quad (6)$$

where $M(q)$ is R^{6x6} symmetric, positive-definite inertia matrix, $C(q, \dot{q})$ contains Coriolis and centrifugal terms, $G(q)$ is the gravity vector and τ_D is the disturbance forces.

To facilitate the control design, the Coriolis, centrifugal, gravitational and disturbance terms are combined into a single nonlinear vector $N(q, \dot{q})$. Thus, equation (6) can be rewritten as

$$M(q)\ddot{q} + N(q, \dot{q}) = \tau. \quad (7)$$

This equation is linear in the control τ, and has full rank $M(q)$, which can be inverted for any valid configuration. Taking the control τ as a function of the manipulator states in the form:

$$\tau = M(q)y + N(q, \dot{q}), \quad (8)$$

 doi:10.20944/preprints202605.0750.v1

leads to the system described by

$$\ddot{q} = y \tag{9}$$

where y represents a virtual input vector:

$$y = K_D(\dot{q}_{des} - \dot{q}) + K_P(q_{des} - q) + r. \tag{10}$$

where r is the reference component [12,19].

This ideal linearization approach relies on exact cancellation of nonlinearities in the system equations of motion. Its practical implementation requires consideration of various sources of uncertainties such as modeling errors, unknown payloads, and computation errors [20]. Also, it requires online computation of the exact time-varying $M(q)$ and $N(q, \dot{q})$, which is computationally expensive, impractical, and in some cases impossible for real-time aerial manipulation tasks. To overcome this, it is proposed to use an offline computed fixed nominal model ($\widehat{M}$, $\widehat{N}$) in the inverse dynamics controller (Figure 2):

$$\tau = \widehat{M}(q)y + \widehat{N}(q, \dot{q}). \tag{11}$$

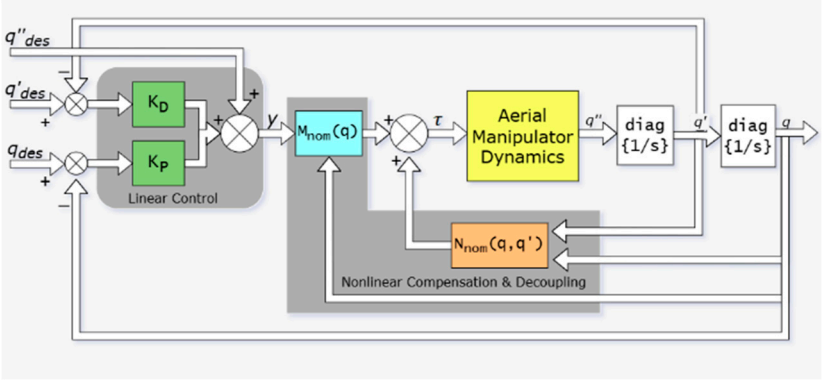

Figure 2. Block diagram of the UAV-manipulator system.

This approach does not result in perfect cancellation of nonlinearities. Instead, applying (11) to the actual plant (7) results in a perturbed linear system. The difference between the fixed nominal model and the actual varying dynamics is isolated into structured uncertainty blocks. The system dynamics become:

$$\ddot{q} = y + \triangle_m y + \triangle_a, \tag{12}$$

where $\triangle_m$, $\triangle_a$ are multiplicative and additive uncertainties:

$$\triangle_m = M^{-1}(q)\widehat{M}(q) - I, \tag{13}$$

$$\triangle_a = M^{-1}(q)\widetilde{N}(q, \dot{q}), \tag{14}$$

$$\widetilde{N}(q, \dot{q}) = \widehat{N}(q, \dot{q}) - N(q, \dot{q}). \tag{15}$$

This formulation, shown in Figure 3, allows treating the nonlinear complex aerial manipulator dynamics as a linear plant subjected to bounded perturbations, which can be managed using robust control techniques.

The robust control input τ for the resulting system is expressed by:

$$\tau = \widehat{M}(q)\left[\ddot{q}_{des} + K_D\dot{\breve{q}} + K_P\breve{q}\right] + \widehat{N}(q, \dot{q}), \tag{16}$$

where $\breve{q}$ and $\dot{\breve{q}}$ are the tracking error and differential error terms.

The robust stability of this control law against the uncertainties $\triangle_m$ and $\triangle_a$ is assessed through the μ-analysis and Small Gain procedure detailed in the next chapter.

 doi:10.20944/preprints202605.0750.v1

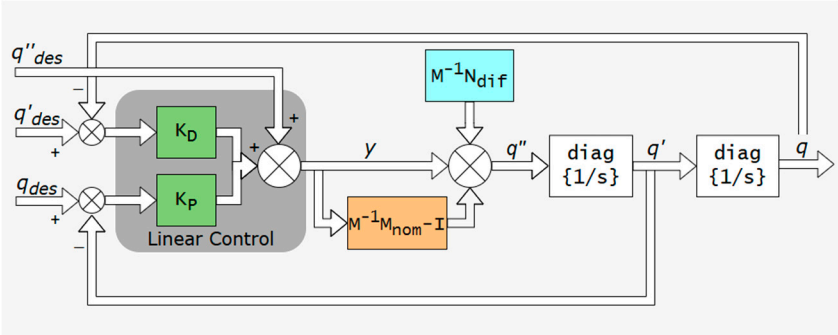

Figure 3. Block diagram of the linearized system.

3. Control System Design

The dynamic model derived in the previous section, particularly the feedback-linearized form in Eq. (12), isolates the system's complex, time-varying, and payload-dependent dynamics into a structured uncertainty term $\Delta = diag(\triangle_m, \triangle_a)$. For a global controller, the bounds on this uncertainty must encompass all possible manipulator configurations, velocities, and payload properties. Such a global bound is necessarily large, forcing a monolithic robust controller to be highly conservative, which severely degrades tracking performance and agility.

In this paper, a hybrid gain-scheduling framework is introduced to minimize the uncertainty bounds. The main strategy is to partition the system's operating space into a finite K number of smaller, dynamically-similar regions. By designing a dedicated controller for each region, it is only necessary to guarantee stability against a much smaller local uncertainty bounds, enabling high-performance control in every region.

3.1. Dynamic-Based State Space Clustering

To avoid the conservatism of a single robust controller, a comprehensive sampling space was defined that includes not only the manipulator kinematics but also the full range of expected payload masses and object dimensions. The dataset, generated by sampling the manipulator's kinematics and potential payload characteristics, is subjected to unsupervised learning. The K-Means algorithm is employed to minimize the within cluster variance of the dynamic parameters.

An important design parameter is the cluster cardinality K. Rather than arbitrarily selecting this value, the distortion curve was analyzed, plotting the within cluster variance against an increasing number of centroids. By identifying the point of rate decrease, the geometric "elbow" of the curve, the optimal K number is determined, that balances model fidelity against controller complexity. This granularity ensures that within any specific cluster k, the deviation between the local nominal model and the actual dynamics remains within the predictable bounds required for robust stability synthesis. The distortion curve for the system is shown in Figure 4, which represents the Sum of Squared Errors (SSE) against the number of clusters.

 doi:10.20944/preprints202605.0750.v1

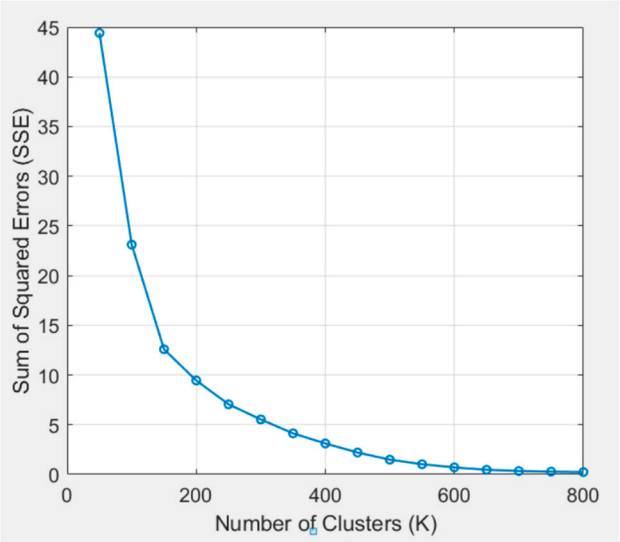

Figure 4. Elbow method for optimal K.

3.2. Robust Controller Synthesis

For each of the K regions, a dedicated robust controller is designed. The system, linearized by approximate inverse dynamics, is analyzed using the LFT framework, as shown in Figure 5. This structure isolates the nominal plant $P(s)$ from the local, bounded uncertainty $\Delta_k = diag(\triangle_m^k, \triangle_a^k)$.

The interconnection matrix $G(s)$ maps the uncertainty blocks (y from $\triangle_m$, w from $\triangle_a$) to their respective inputs (y, z) through the nominal closed-loop system. For the considered 6-DOF system with a diagonal plant $P_{plant}(s) = diag\{\frac{1}{s^2}, \dots, \frac{1}{s^2}\}$ and a diagonal PD controller $K(s) = diag\{K_1(s), \dots, K_6(s)\}$, where each $K_k(s) = K_p^k + K_d^k s/(\mathrm{T}s + 1)$, where T is the time constant for the derivative filter, interconnection matrix $G(s)$ is a 12x12 block matrix given by:

$$G(s) = \begin{pmatrix} -diag\left\{\frac{K(s)}{s^2+K(s)}\right\} & -diag\left\{\frac{K(s)}{s^2+K(s)}\right\} \\ diag\left\{\frac{1}{s^2+K(s)}\right\} & diag\left\{\frac{1}{s^2+K(s)}\right\} \end{pmatrix}. \quad (17)$$

According to the Small Gain Theorem, robust stability for cluster k is guaranteed if the supremum of the structured singular value $\mu_\Delta(P(jw))$ is less than 1 for all frequencies w [13]:

$$\sup_w \mu_{\Delta_k}(P(C_k, jw)) < 1. \quad (18)$$

where C_k is the local controller for cluster k. This condition can be expressed as ensuring the robust stability margin, $\beta_k = 1/\mu_{\Delta_k}$, is greater than 1.

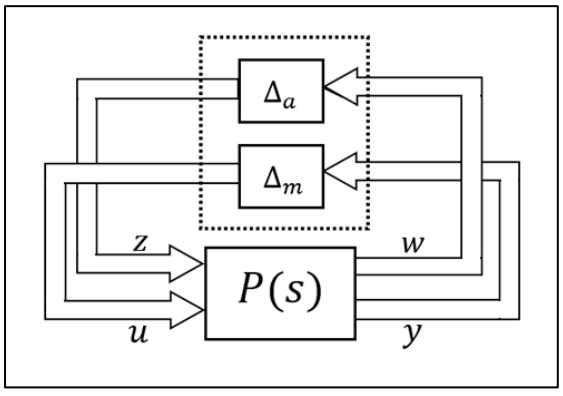

Figure 5. LFT model of the system.

 doi:10.20944/preprints202605.0750.v1

Calculating the optimal feedback K_p^k and K_d^k that simultaneously satisfy the Small Gain condition $\beta_k > 1$ and minimize tracking error is a non-convex optimization problem. To navigate this complex search space, a GA is employed to evolve a population of candidate controllers for each cluster. The fitness of each candidate solution is evaluated using a composite cost function that penalizes instability while rewarding precision. The evaluation criteria J balances 3 objectives:

$$J = -\left[\alpha_1 e + \alpha_2 \|\tau\|^2 + \alpha_3 \max\left(0, \mu_{\Delta_k}(P(jw)) - 1\right)\right], \quad (19)$$

where $\alpha_1, \alpha_2, \alpha_3$ are tuning weights. The first term minimizes the Root Mean Square Error (RMSE) of the step response and the second term penalizes excessive control effort. The third term returns zero if the stability margin $\beta_k > 1$, but applies a heavy penalty if the robust stability constraint is violated. Consequently, the optimization process selects a set of controller parameters that maximize the precision without compromising the system's stability margins. As can be observed from the relationship between the linearized dynamics (9) and the control law (10), the selection of positive-definite gain matrices ensures the asymptotic convergence of the tracking error to zero. This structure guarantees that the system accurately follows the desired trajectory even when subjected to the bounded model uncertainties defined in (12).

3.3. Gain-Scheduling Specification

The gain-scheduling framework functions as an adaptive mechanism that bridges real-time system dynamics with a library of robust controllers. During operation, the system calculates the Euclidean distance between the current dynamic state and the pre-computed operating points, or cluster centroids, which represent the partitioned workspace. By identifying the most relevant clusters in this space, the system retrieves the corresponding control gains that were specifically optimized during the offline tuning phase.

4. Simulation and Results

To validate the performance and robustness of the proposed control framework, a series of simulations were conducted. The primary goal of the simulations is to demonstrate that the offline-tuned controller can robustly manage the dynamic variations introduced by fast manipulator motion and grasping a variety of payloads.

4.1. Simulation Setup

Simulations were conducted in the MATLAB software to validate the performance and robustness of the UAV-manipulator system. The configuration space was sampled by focusing on the dominant variables: the manipulator joint angles, the payload mass, and the payload dimensions. The grid of manipulator positions was combined with a set of payload masses and 3 distinct object dimension sets (small, medium, big). For each combination, a set of 5 fast velocity vectors (up to $\pi/2$ rad/s) were applied to capture the full dynamics, including Coriolis and Gravitational effects.

This process generated a comprehensive dataset of [~30000] valid dynamic samples $d_i = [M, N]$. The K-Means algorithm, using $K = 300$ (determined from the Elbow Method, Figure 4), was used to partition this data, yielding 300 local nominal models $(\widehat{M}_k, \widehat{N}_k)$.

The 99th percentile H_∞ of the local uncertainty $\Delta_k = diag(\triangle_m^k, \triangle_a^k)$, was calculated for each cluster. This resulted in significantly small bounds for both multiplicative (γ_m^k) and additive (γ_a^k) uncertainties (Figure 6), which were then used as constraints for the controller synthesis.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 May 2026 doi:10.20944/preprints202605.0750.v1

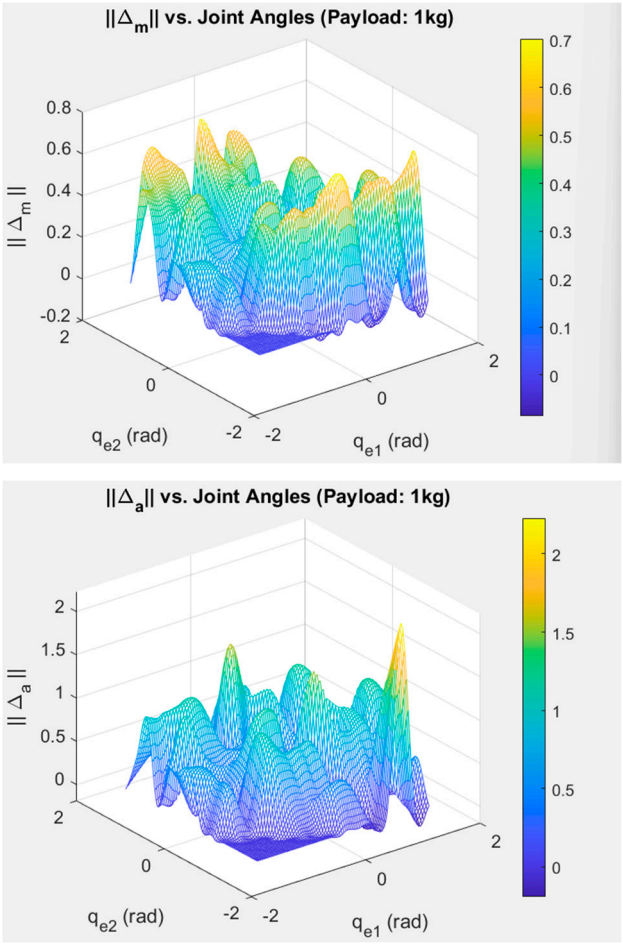

Figure 6. Uncertainty norms in clusters.

For each of the K=300 clusters, a local diagonal PD controller was tuned using the GA as described in the previous section.

The dynamic performance of the proposed system depends on the behavior of the entire setup, including the masses and dimensions of the UAV and manipulator. The physical parameters used in the simulations are summarized in Table 1.

Table 1. UAV and manipulator characteristics.

Parameter	Symbol	Value	Unit
UAV mass	m_{UAV}	4.5	kg
UAV arm length	L_{UAV}	0.5	m
UAV Moment of Inertia (X,Y axes)	I_X, I_Y	0.075	kgm^2
UAV Moment of Inertia (Z axis)	I_Z	0.12	kgm^2
Length of link 1	L_1	0.2	m
Length of link 2	L_2	0.3	m
Mass of link 1	m_{L1}	0.3	kg
Mass of link 2	m_{L2}	0.45	kg

4.2. Robust Stability Validation

The first validation test was conducted to confirm that the GA successfully synthesized controllers that meet the robust stability conditions. The robust stability margin, $\beta_k = 1/\mu_{\Delta_k}$, was

 doi:10.20944/preprints202605.0750.v1

calculated using the final tuned controller in each of the 300 clusters. As shown in Figure 7 the stability margin for all 300 clusters is greater than the robust stability boundary of $\beta = 1$.

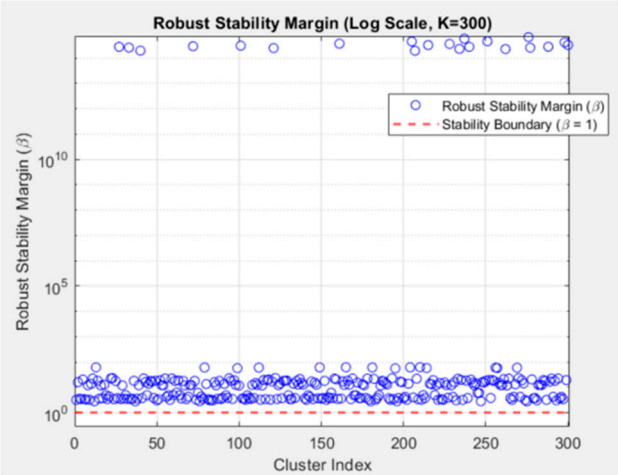

Figure 7. Robust stability margin (β) on Logarithmic Scale.

This result confirmed that the offline GA tuning, constrained by the local bounds, successfully generated a set of controllers that are guaranteed to be robustly stable within their respective operating regions.

The second simulation was implemented to validate the adaptability and precision of the control framework when confronted with the simultaneous changes in both manipulator configuration and payload properties. This simulation was executed over a 30-second span, forcing the system through critical operational states. The UAV's translational and attitude states were commanded to hold a still position of $Z = 1m$ and $\phi, \theta, \psi = 0$, while manipulator joints were commanded to perform a sinusoidal tracking motion from - $\pi/2$ to $\pi/2$ radians, and actively switching the payload every 10 seconds to stress the controller's adaptability. The payload characteristics and corresponding grasping times are given below in Table 2.

Table 2. Payload characteristics.

Object	Time	Mass (kg)	Dimensions(m)
Small	0-10 sec	0.1	0.1, 0.02, 0.02
Medium	10-20 sec	0.5	0.4, 0.05, 0.05
Big	20-30 sec	1	0.6, 0.08, 0.08

The resulting plots are depicted in Figure 8. The plots show that the actual trajectory remains tightly coupled with the desired trajectory for all controlled states. Only minimal tracking errors were observed during the manipulation of the biggest payload configuration, and these deviations are quantitatively negligible.

 doi:10.20944/preprints202605.0750.v1

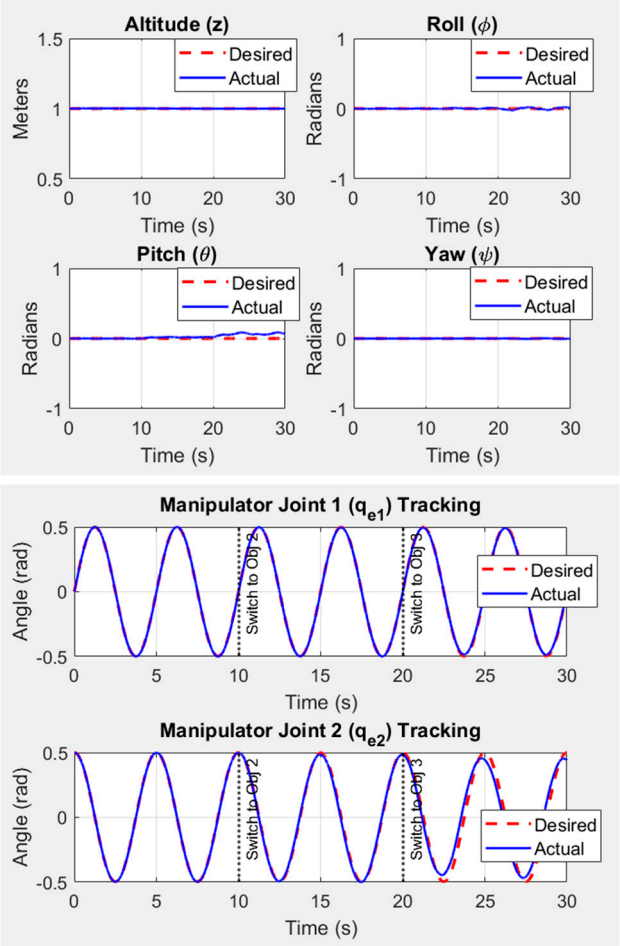

Figure 8. Trajectory tracking of the system.

Gain switching was logged along the simulations to analyze the control stability at transition points. To mitigate the potential for abrupt control actions, the gain interpolation technique was used to ensure a smooth transition between scheduled gains. Instead of selecting a single best controller, this method blends the control laws from multiple adjacent clusters. Using the k-nearest neighbors algorithm, the system identifies the 4 closest cluster centroids to the current dynamic state. This provides a set of indices for the 4 most relevant local controllers. For each of the neighboring clusters, the weight component w_i is calculated based on its Euclidean distance D_i from the current state. The weight is computed as the inverse of the distance, with a small ε added to avoid division by zero

$$w_i = 1/D_i + \varepsilon, \tag{20}$$

the weights are then normalized to sum to one, esuring a stable weighted average:

$$w_i^* = \frac{w_i}{\sum_{j=1}^{k} w_j}, \tag{21}$$

$$K_{blend} = \sum_{i=1}^{k} w_i^* \cdot K_i. \tag{22}$$

The resulting gain-switch plot is presented in Figure 9. As a result total of 176 gain switches occurred with 48 unique cluster gains used, indicating continuous adaptation.

 doi:10.20944/preprints202605.0750.v1

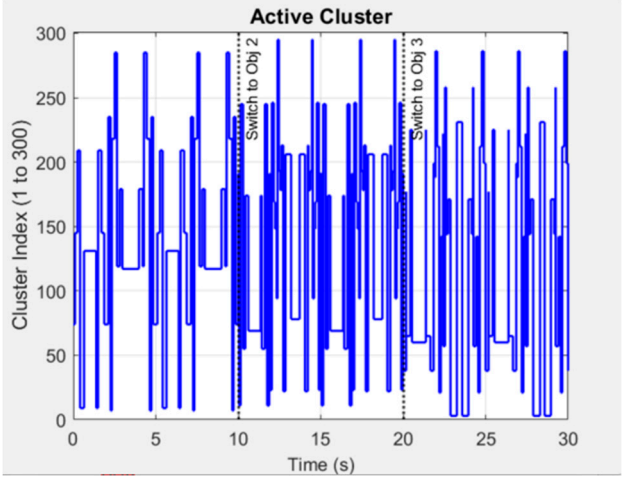

Figure 9. Active Cluster Index over Time.

The integrated gain interpolation successfully mitigated the discontinuities of hard-switching. This is quantitatively supported by the RMSE plot in Figure 10, which confirms that the smooth switching mechanism helped maintain a consistently low tracking error across the entire mission, even when controlling the system under the biggest payload (Obj. 3).

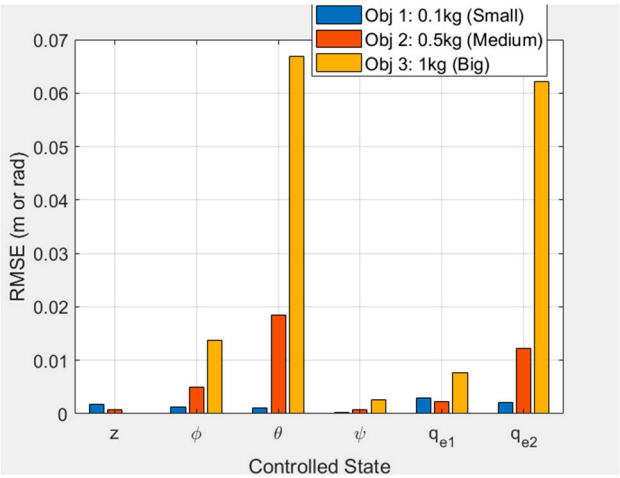

Figure 10. RMSE by State.

5. Conclusions

This paper successfully addressed the complex challenge of robust trajectory tracking for an aerial manipulator during fast maneuvers with varying payloads. A novel gain-scheduling framework was introduced that shifts the computational burden to pre-flight phase, overcoming the real-time slow computations and poor adaptability of traditional model-based controllers.

The core contribution is a method that first linearizes the system using approximate inverse dynamics with fixed nominal models, isolating the complex, time-varying dynamics as structured uncertainties. The entire configuration space, including payload variations, was then partitioned into 300 dynamically similar clusters using K-Means. For each local region, a dedicated robust PD controller was synthesized using a multi-objective GA, which optimized tracking performance while mathematically guaranteeing robust stability via the Small Gain Theorem. Simulation results validated this approach, confirming that all 300 local controllers were robustly stable within their operating regions. The system demonstrated high-precision trajectory tracking for all states while performing an aggressive sinusoidal motion and payload switching from 0 to 1 kg. Furthermore, a smooth gain interpolation technique based on k-nearest neighbors algorithm successfully mitigated abrupt control motions, ensuring continuous stability and consistently low tracking error across all

 doi:10.20944/preprints202605.0750.v1

operational scenarios. This work confirms that an offline-tuned, gain-scheduled control system can achieve the high performance, robustness, and adaptability required for fast aerial manipulation tasks. Future work will focus on the experimental validation of the proposed hybrid gain-scheduling framework on a physical UAV-manipulator platform to evaluate performance under real-world constraints such as sensor noise and actuator latency.

References

1. J. Meng, et al., "On aerial robots with grasping and perching capabilities: A Comprehensive Review", Frontiers in Robotics and AI, vol. 8, Mar. 2022, 1-24. DOI: 10.3389/frobt.2021. 739173.
2. J. Morais, D. Cardoso, G.V. Raffo, "Robust optimal nonlinear control strategies for an aerial manipulator", Congresso Brasileiro de Automática, vol. 2, no 1, 2020, DOI: 10.48011/asba.v2i1.1616.
3. N. Mimmo, et al., "Robust motion control of aerial manipulators", Annual Reviews in Control, vol. 49, 2020, 230-238, DOI: 10.1016/j.arcontrol.2020. 04.006.
4. R. Naldi, et al., "Robust Control of an Aerial Manipulator Interacting with the Environment", IFAC-PapersOnLine, vol. 51, Issue 13, 2018, 537-542, DOI: 10.1016/j.ifacol.2018.07 .335.
5. C. Carvajal, et al., "Multitask control of aerial manipulator robots with dynamic compensation based on numerical methods", Robotics and Autonomous Systems, vol. 173, 2024, DOI: 10.1016/j.robot.2023. 104614.
6. P. Kremer, J. Sanchez-Lopez, H. Voos, "A Hybrid Modelling Approach for Aerial Manipulators", Journal of Intelligent & Robotic Systems, vol. 105, no 74, 2022, DOI: doi.org/10.1007/s10846-022-01640-1.
7. Y. Chen et al., "Robust Control for Unmanned Aerial Manipulator Under Disturbances," IEEE Access, vol. 8, 2020, 129869–129877, DOI: 10.1109/ACCESS. 2020.3008971.
8. I. Al-Darraji, et al., "Adaptive Robust Controller Design-Based RBF Neural Network for Aerial Robot Arm Model", Electronics, vol. 10, 2021, 831, DOI: 10.3390/electronics10070831.
9. Hai Li, et al., "Adaptive Neural Network Backstepping Control Method for Aerial Manipulator Based on Variable Inertia Parameter Modeling", 2022, DOI: 10.48550/arXiv.2212.04250.
10. C. Park, A. Ramirez-Serrano, M. Bisheban, "Estimation of Time-Varying Inertia of Aerial Manipulators Performing Manipulation of Unknown Objects", Proceedings of the 10th International Conference of Control Systems, and Robotics, Niagara Falls, Canada, no. 209, June 2023, DOI: 10.11159/ cdsr23.209.
11. I. Imran, K. Wood, A. Montazeri, "Adaptive Control of Unmanned Aerial Vehicles with Varying Payload and Full Parametric Uncertainties", Electronics, vol. 13, 2024, 347, DOI: 10.3390/ electronics13020347.
12. Siciliano B., et al., Robotics: Modelling, Planning and Control, Springer Publishing Company, Incorporated, 2010.
13. Zhou K., Doyle J., Essentials of Robust Control, Prentice-Hall, Englewood Cliffs, NJ, 1998.
14. E. Umargono, J. Suseno, S.K. Gunawan, "K-Means Clustering Optimization Using the Elbow Method and Early Centroid Determination Based on Mean and Median Formula", Proceedings of the 2nd International Seminar on Science and Technology, vol. 474, 2020, 121-129, DOI: 10.2991/assehr.k.201010. 019.
15. Y. Ma, "Optimization of basic PID control algorithm based on genetic algorithm and Matlab", Proceedings of the 3rd International Conference on Computing Innovation and Applied Physics, vol. 30, 2024, 178-186, DOI: 10.54254/2753-8818/30/20241 103.
16. O. Gasparyan, et al, "Robustness Analysis of UAVs' Control Systems in Case of Motors' Partial Efficiency Degradation", Bulletin of High Technology, vol. 25, no 1, 2023, 67-80
17. Gasparyan O., Linear and Nonlinear Multivariable Feedback Control: A Classical Approach. John Wiley & Sons Ltd, UK, 2008, 374.
18. D. Li, W. Hongtao, "Dynamical Modelling and Robust Control for an Unmanned Aerial Robot Using Hexarotor with 2-DOF Manipulator", International Journal of Aerospace Engineering, 2019, 12, DOI: 10.1155/2019/5483073.

 doi:10.20944/preprints202605.0750.v1

19. S. Friedrich, M. Buss, "Parameterizing robust manipulator controllers under approximate inverse dynamics: A double-Youla approach", Int J Robust Nonlinear Control, vol. 29, June 2019, 5137–5163, DOI: 10.1002/rnc.4671.
20. Spong M., Hutchinson S., Vidyasagar M., Robot Modeling and Control, John Wiley & Sons, Inc., USA, 2006, 407.