AERO-LQG: Aerial-Enabled Robust Optimization for LQG-Based Quadrotor Flight Controller

Daniel Engelsman and Itzik Klein*
Hatter Department of Marine Technologies, University of Haifa, Israel.

http://www.imavs.org/

Abstract

Quadrotors are vital across civilian, industrial, and military domains, where balancing the high power demands of agility with energy efficiency remains a critical constraint. Meeting this trade-off requires mode-specific optimization frameworks. While optimal control policies offer robust performance, selecting their underlying weight matrices lacks analytical guidance and typically relies on exhaustive search. This interdependence can be framed as a bi-level optimization problem, where the outer loop determines weights a priori. This work introduces Aerial-Enabled Robust Optimization for LQG tuning (AERO-LQG), a framework employing evolutionary strategies to fine-tune Linear-Quadratic-Gaussian weighting parameters. Applied to the linearized hovering mode of quadrotor flight, AERO-LQG achieves performance gains of several tens of percent, underscoring its potential for high-performance, energy-efficient control. The project is available @ GitHub.

1 Introduction

Unmanned aerial vehicles (UAVs), particularly quadrotors, have rapidly evolved from research prototypes into practical tools across a broad range of applications. Their versatility and agility enable complex missions—such as surveillance [1], inspection [2], delivery [3], and search and rescue [4]—making them valuable assets in both civilian and military domains [5]. From a dynamical systems perspective, quadrotor flight modes span a wide stability spectrum—from near-equilibrium hovering to highly nonlinear, aggressive maneuvers. As a result, no single control strategy can achieve optimal performance across all regimes. Transitions between control modes are therefore common, especially as energy constraints impose competing demands: controllers must either deliver agility [6], or conversely, limit maneuverability to conserve energy and extend flight time [7, 8]. Monolithic control lacks such flexibility; adaptive scheduling with task-dependent adjustments is needed [9–11]. Frequent mode switching, however, complicates state estimation, which remains fragile to the quadrotor's nonlinear and underactuated dynamics—despite advances in hardware, inertial sensing, and AI-driven perception [12–18]. To mitigate these issues, linear quadratic Gaussian (LQG) control remains a staple in aerial robotics, valued for its theoretical rigor and optimality under specific assumptions [19]. Many stabilized flight modes can be approximated as linear systems, allowing precise LQG control, provided operating conditions stay near the linearization point [20]. Quadrotors are no exception—their hovering mode, in particular, can be effectively linearized to facilitate both control and state estimation. However, while the quadratic form of the cost function is convex for fixed weights, tuning these values constitutes an additional outer optimization task—highly non-convex due to the coupled estimator–controller dynamics [14]. This work investigates whether a bi-level optimization framework can systematically maximize the stable flight envelope of LQG-controlled quadrotors under dynamic state uncertainty.

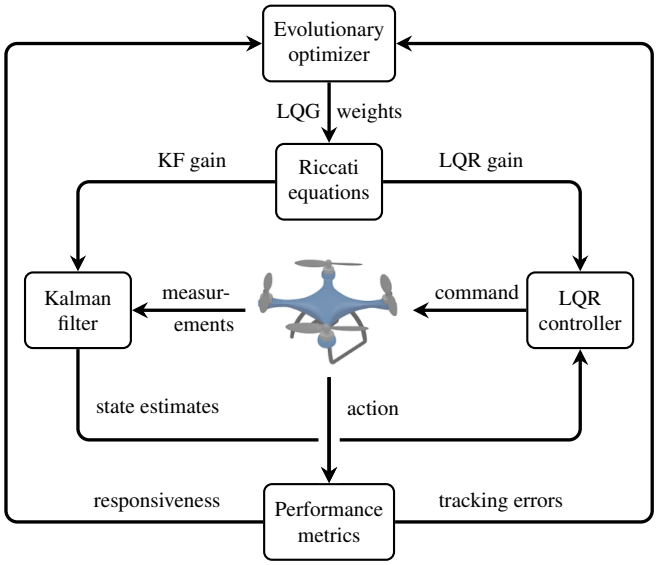

Figure 1: Flowchart of the AERO-LQG bi-level optimization.

To address this objective, we develop an aerial-enabled robust optimization for LQG tuning (AERO-LQG), tailored to modern quadrotors. As illustrated in Figure 1, an outer evolutionary loop refines the weighting matrices, while an inner loop computes the LQG gains, together forming a nested architecture. Evaluated across diverse metrics, AERO-LQG delivers unprecedented gains—reducing tracking error by over 55% and extending flight endurance by nearly 8%.

*Email addresses: {dengelsm@campus, kitzik@univ}.haifa.ac.il

http://www.imavs.org/

The paper unfolds as follows: Section 2 presents the problem setup and technical foundations; Section 3 outlines the proposed approach; Section 4 discusses the findings; and Section 5 concludes the study.

2 Problem Setup

Quadrotor behavior is modeled as a continuous-time dynamical system, defined by the following differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \,, \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the control input vector, and $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a smooth function that defines its dynamics. The solution to this system describes the trajectory of the state $\boldsymbol{x}(t)$, starting from an initial condition $\boldsymbol{x}(0)$ and evolving under the influence of the input $\boldsymbol{u}(t)$. Most aerial flight modes such as take-off, uncoordinated turns, maneuvers, and landing, require the system states to change dramatically, i.e. $\dot{\boldsymbol{x}}(t) \neq \boldsymbol{0}$, thereby causing $\boldsymbol{f}$ to evolve nonlinearly in time. That said, certain operating points, if controlled properly, enable maintaining the system at an equilibrium, denoted by subscript e, with their states demonstrating negligible change [21], such that

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}_e, \boldsymbol{u}_e) = \boldsymbol{0} \,. \tag{2}$$

The equilibrium behavior is assessed using local linearity, approximating the nonlinear dynamics $\boldsymbol{f}$ with a linear model near the equilibrium point $(\boldsymbol{x}_e, \boldsymbol{u}_e)$. General fixed-wing configurations permit linear approximation of several trimmed (stable) flight modes—such as straight-and-level flight, coordinated turns, and steady climbs or descents—thanks to the gliding capability of their large wing area. Rotorcraft, by contrast, depend entirely on rapid, coordinated motor corrections, yielding predominantly nonlinear dynamics [22, 23]. Consequently, the only flight mode that consistently maintains equilibrium under small perturbations is hovering (center), as depicted in the finite state machine in Figure 2.

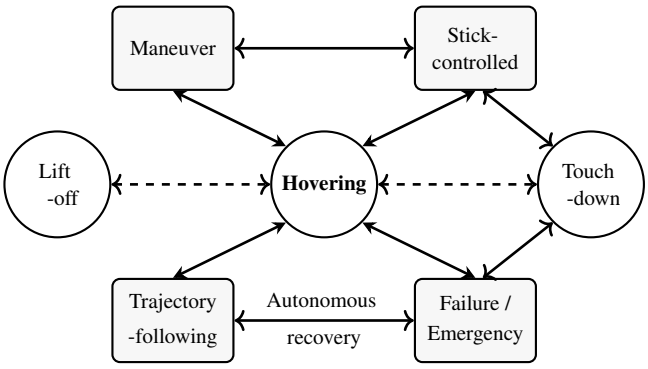

Figure 2: State transition diagram with hovering as the central hub connecting diverse nonlinear flight modes (boxes).

2.1 Equilibrium Point

Based on linear systems theory, steady-state dynamics allow the nonlinear system (1) to be linearized around the equilibrium point $(\boldsymbol{x}_e, \boldsymbol{u}_e)$ through a first-order Taylor expansion [24]. Since by definition, quadrotor hovering balances the net forces and net moments into zero, the following approximation holds around the hovering equilibrium point

$$\dot{\boldsymbol{x}}(t) \approx \mathbf{A}(\boldsymbol{x}_e, \boldsymbol{u}_e)(\boldsymbol{x}(t) - \boldsymbol{x}_e) + \mathbf{B}(\boldsymbol{x}_e, \boldsymbol{u}_e)(\boldsymbol{u}(t) - \boldsymbol{u}_e), \tag{3}$$

where $\boldsymbol{x}(t)$ is the state vector, $\boldsymbol{u}(t)$ the control inputs, and the deviations from equilibrium are expressed as

$$\delta\boldsymbol{x}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_e \;\text{ and }\; \delta\boldsymbol{u}(t) = \boldsymbol{u}(t) - \boldsymbol{u}_e \,. \tag{4}$$

Assuming linear time-invariant (LTI) dynamics in the vicinity of the equilibrium point, the system Jacobians with respect to the state and input are defined as

$$\mathbf{A}(\boldsymbol{x}_e, \boldsymbol{u}_e) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}_e, \boldsymbol{u}_e)} \text{ and } \mathbf{B}(\boldsymbol{x}_e, \boldsymbol{u}_e) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{x}_e, \boldsymbol{u}_e)} , \tag{5}$$

reducing the error dynamics to the standard LTI form

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \mathbf{A}\,\boldsymbol{x}(t) + \mathbf{B}\,\boldsymbol{u}(t) \,, \\ \boldsymbol{y}(t) &= \mathbf{C}\,\boldsymbol{x}(t) \,, \end{aligned} \tag{6}$$

where matrix $\mathbf{C}$ extracts the measurable states from $\boldsymbol{x}$. Model discrepancies and unmodeled dynamics are represented by zero-mean white Gaussian noise processes, with process covariance $\mathbf{W}$ and measurement covariance $\mathbf{V}$

$$\boldsymbol{w}(t) \sim \mathcal{N}(\boldsymbol{0}, \mathbf{W}) \,, \quad \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^\top] = \mathbf{W} \succ 0 \,, \tag{7}$$

$$\boldsymbol{v}(t) \sim \mathcal{N}(\boldsymbol{0}, \mathbf{V}) \,, \quad \mathbb{E}[\boldsymbol{v}\boldsymbol{v}^\top] = \mathbf{V} \succeq 0 \,. \tag{8}$$

These propagate the open-loop uncertainty, whose state covariance $\mathbf{P}(t)$ evolves by the continuous-time Lyapunov equation

$$\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^\top + \mathbf{W}. \tag{9}$$

At any fixed time t, the joint distribution of the states and outputs can thus be approximated as Gaussian, namely

$$\begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{y}(t) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(t) \\ \mathbf{C}\boldsymbol{\mu}(t) \end{bmatrix}, \begin{bmatrix} \mathbf{P}(t) & \mathbf{P}(t)\mathbf{C}^\top \\ \mathbf{C}\mathbf{P}(t) & \mathbf{C}\mathbf{P}(t)\mathbf{C}^\top + \mathbf{V} \end{bmatrix} \right). \tag{10}$$

where $\boldsymbol{\mu}(t)$ is the nominal noise-free state trajectory, initialized as $\boldsymbol{\mu}(t_0) = \mathbb{E}[\boldsymbol{x}(t_0)]$, with time evolution given by

$$\dot{\boldsymbol{\mu}}(t) = \mathbf{A}\boldsymbol{\mu}(t) + \mathbf{B}\boldsymbol{u}(t) \,. \tag{11}$$

http://www.imavs.org/

2.2 Inner-Level Optimization

Having identified the hovering operating point as locally linear under small perturbations enables the application of linear control theory. In this framework, solving the Riccati equations yields optimal controller ($\mathbf{K}$) and estimator ($\mathbf{L}$) gains, whereby the Kalman filter provides state estimates to the linear quadratic regulator (LQR). These gains are obtained by minimizing a quadratic cost over a finite time horizon T,

$$\min_{\boldsymbol{u}(t)} \mathcal{J}_{\text{in}} \triangleq \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\int_0^T \left(\|\boldsymbol{x}(t)\|^2_{\mathbf{Q}} + \|\boldsymbol{u}(t)\|^2_{\mathbf{R}}\right) \mathrm{d}t\right], \tag{12}$$

subject to (11), with a-priori specified $\mathbf{Q} \succeq 0$ and $\mathbf{R} \succ 0$ weighting deviations in $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$, respectively [25, 26]. By defining the estimation and the control errors as

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)\,, \tag{13}$$
$$\hat{\boldsymbol{\varepsilon}}(t) = \hat{\boldsymbol{x}}(t) - \boldsymbol{x}_{\text{ref}}\,, \tag{14}$$

the plant-estimator dynamics admits the block-state form of

$$\begin{bmatrix}\dot{\boldsymbol{x}}\\ \dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}\mathbf{A}-\mathbf{BK} & \mathbf{BK}\\ \mathbf{0} & \mathbf{A}-\mathbf{LC}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{e}\end{bmatrix} + \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{I} & -\mathbf{L}\end{bmatrix}\begin{bmatrix}\boldsymbol{w}\\ \boldsymbol{v}\end{bmatrix}, \tag{15}$$

highlighting the separation principle, with an upper-triangular form that decouples the estimator dynamics [27, 28].

2.3 System Dynamics

We begin by outlining the assumptions and coordinate frames relevant to hovering, a near-equilibrium condition that justifies standard modeling practices [29, 30]: (i) the quadrotor is a rigid symmetric body; (ii) thrust and moments scale with the square of rotor speeds; (iii) aerodynamic drag is neglected due to low speed; (iv) the center of mass coincides with the body-frame origin. The orientation of the body frame $\mathcal{B}$ relative to the inertial frame $\mathcal{I}$ is given by the rotation matrix

$$\mathcal{R}^{\mathcal{I}}_{\mathcal{B}} := \left(\mathbf{e}^{\mathcal{B}}_x, \mathbf{e}^{\mathcal{B}}_y, \mathbf{e}^{\mathcal{B}}_z\right) \in SO(3)\,, \tag{16}$$

which is constructed through a sequence of three consecutive rotations: $\mathcal{R}_z(\psi)$ about the $\mathbf{e}^{\mathcal{I}}_z$ axis, $\mathcal{R}_y(\theta)$ about the $\mathbf{e}^{\mathcal{I}}_y$ axis, and $\mathcal{R}_x(\phi)$ about the $\mathbf{e}^{\mathcal{I}}_x$ axis, where $\{\phi,\theta\} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ and $\psi \in (-\pi, \pi)$ correspond to the roll, pitch, and yaw angles, respectively. Based on the Newton-Euler formalism [31], the translational-rotational coupled dynamics are expressed as

$$\begin{pmatrix}\boldsymbol{f}^{\mathcal{B}}\\ \boldsymbol{\tau}^{\mathcal{B}}\end{pmatrix} = \begin{pmatrix}m\boldsymbol{I}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \boldsymbol{J}\end{pmatrix}\begin{pmatrix}\dot{\boldsymbol{v}}^{\mathcal{B}}\\ \dot{\boldsymbol{\omega}}^{\mathcal{B}}\end{pmatrix} + \begin{pmatrix}\boldsymbol{\omega}^{\mathcal{B}} \times m\,\boldsymbol{v}^{\mathcal{B}}\\ \boldsymbol{\omega}^{\mathcal{B}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathcal{B}}\end{pmatrix}, \tag{17}$$

where, m is the body mass; $\boldsymbol{J}$, $\boldsymbol{I}_3$, and $\mathbf{0}_3$ are 3×3 inertia tensor, identity, and zero matrices. Forces $\boldsymbol{f}$ and moments $\boldsymbol{\tau}$, expressed in the body frame, govern the time derivatives of $\boldsymbol{v}^{\mathcal{B}}$ and $\boldsymbol{\omega}^{\mathcal{B}}$. In fixed-point hover, near-zero body velocities make drag negligible, so the net body-frame force includes only gravity $\mathbf{g}^{\mathcal{B}} = \mathrm{g}\,\mathbf{e}^{\mathcal{B}}_z$ and vertical rotor thrusts $\boldsymbol{f}^{\mathcal{B}}_{z,i}$, as in

$$\boldsymbol{f}^{\mathcal{B}} = \sum_{i=1}^{4} \boldsymbol{f}^{\mathcal{B}}_{z,i} - m\,\boldsymbol{g}^{\mathcal{B}}\,. \tag{18}$$

Each rotor spins along the body-frame z-axis, generating thrust by the i-th propeller modeled as

$$\boldsymbol{f}^{\mathcal{B}}_{z,i} = f_{z,i}\,\mathbf{e}^{\mathcal{B}}_z = \left(k_T\Omega_i^2\right)\mathbf{e}^{\mathcal{B}}_z\,, \tag{19}$$

where k_T is a lumped coefficient. Attitude stabilization is achieved through differential thrust between opposing rotors spaced by ℓ. The rolling moment arises from the thrust difference between motors #4 (left) and #2 (right), given by

$$\boldsymbol{\tau}^{\mathcal{B}}_{\phi} = \ell\left(\boldsymbol{f}^{\mathcal{B}}_{z,4} - \boldsymbol{f}^{\mathcal{B}}_{z,2}\right)\mathbf{e}^{\mathcal{B}}_x\,, \tag{20}$$

while the pitch moment is controlled by the longitudinal thrust difference between motors #1 (front) and #3 (rear), as

$$\boldsymbol{\tau}^{\mathcal{B}}_{\theta} = \ell\left(\boldsymbol{f}^{\mathcal{B}}_{z,3} - \boldsymbol{f}^{\mathcal{B}}_{z,1}\right)\mathbf{e}^{\mathcal{B}}_y\,. \tag{21}$$

Finally, yaw arises from asymmetry in rotor drag torques, i.e., when $\sum_i \tau_{M,i} \neq 0$. Defining k_M as the torque coefficient and positive yaw as counter-clockwise about $\mathbf{e}^{\mathcal{B}}_z$, yaw torque is

$$\boldsymbol{\tau}^{\mathcal{B}}_{\psi} = \sum_{i=1}^{4} \tau_{M,i} = (-1)^i\left(k_M\Omega_i^2\right)\mathbf{e}^{\mathcal{B}}_z\,. \tag{22}$$

The platform's trajectory is derived by time-integrating its dynamics and applying a coordinate transformation to an inertial reference frame, letting the position vector be [19]

$$\boldsymbol{\xi}^{\mathcal{I}} = (x, y, z)^{\top}\,, \tag{23}$$

with the corresponding body velocities related by

$$\dot{\boldsymbol{\xi}}^{\mathcal{I}} = \mathcal{R}^{\mathcal{I}}_{\mathcal{B}}\boldsymbol{v}^{\mathcal{B}}\,. \tag{24}$$

The Euler angles vector in the inertial frame is denoted as

$$\boldsymbol{\eta}^{\mathcal{I}} = (\phi, \theta, \psi)^{\top}\,, \tag{25}$$

with its rate of change related to the body angular rates (p, q, r) through the transformation matrix $\boldsymbol{W}_{\boldsymbol{\eta}}$, such that

$$\dot{\boldsymbol{\eta}}^{\mathcal{I}} = \boldsymbol{W}_{\boldsymbol{\eta}}\,\boldsymbol{\omega}^{\mathcal{B}} = \begin{pmatrix}1 & \mathrm{s}_\phi\mathrm{t}_\theta & \mathrm{c}_\phi\mathrm{t}_\theta\\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi\\ 0 & \mathrm{s}_\phi/\mathrm{c}_\theta & \mathrm{c}_\phi/\mathrm{c}_\theta\end{pmatrix}\begin{pmatrix}p\\ q\\ r\end{pmatrix}, \tag{26}$$

where 't' is the tangent function. For derivation, thrust and torques are idealized as direct control input functions, hence

$$\boldsymbol{u}(\Omega) = \begin{pmatrix}\boldsymbol{u}_z\\ \boldsymbol{u}_\phi\\ \boldsymbol{u}_\theta\\ \boldsymbol{u}_\psi\end{pmatrix} = \begin{pmatrix}f_z\,\mathbf{e}^{\mathcal{B}}_z\\ \tau_\phi\,\mathbf{e}^{\mathcal{B}}_x\\ \tau_\theta\,\mathbf{e}^{\mathcal{B}}_y\\ \tau_\psi\,\mathbf{e}^{\mathcal{B}}_z\end{pmatrix} \in \mathbb{R}^4\,. \tag{27}$$

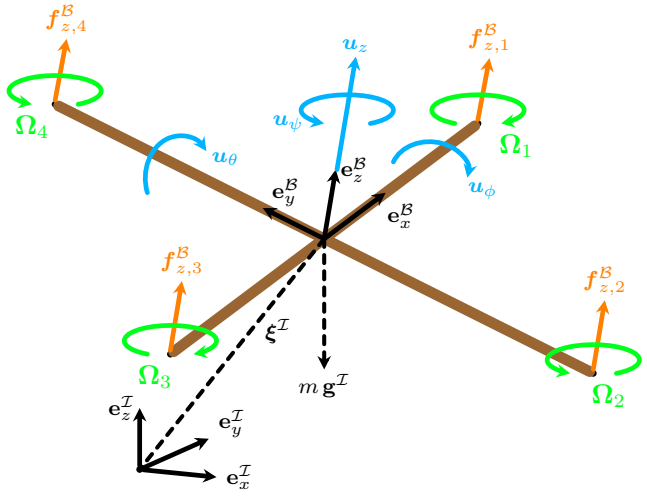

Figure 3: Free-body diagram of the quadrotor system model.

Finally, by incorporating the generalized coordinates into the system model, the augmented state vector is defined as

$$\boldsymbol{x} = \begin{pmatrix} (\boldsymbol{\xi}^{\mathcal{I}})^\top & (\boldsymbol{v}^{\mathcal{B}})^\top & (\boldsymbol{\eta}^{\mathcal{I}})^\top & (\boldsymbol{\omega}^{\mathcal{B}})^\top \end{pmatrix}^\top \in \mathbb{R}^{12}, \quad (28)$$

with the nonlinear state-space model in (1) evolves as

$$\begin{pmatrix} \dot{\boldsymbol{\xi}}^{\mathcal{I}} \\ \dot{\boldsymbol{v}}^{\mathcal{B}} \\ \dot{\boldsymbol{\eta}}^{\mathcal{I}} \\ \dot{\boldsymbol{\omega}}^{\mathcal{B}} \end{pmatrix} = \begin{pmatrix} (\mathcal{R}^{\mathcal{I}}_{\mathcal{B}})\,\boldsymbol{v}^{\mathcal{B}} \\ m^{-1}\boldsymbol{u}_z - \boldsymbol{\omega}^{\mathcal{B}} \times \boldsymbol{v}^{\mathcal{B}} - g\,\mathbf{e}^{\mathcal{B}}_z \\ (\boldsymbol{W}_{\boldsymbol{\eta}})\boldsymbol{\omega}^{\mathcal{B}} \\ \boldsymbol{J}^{-1}\left((\boldsymbol{u}_\phi + \boldsymbol{u}_\theta + \boldsymbol{u}_\psi) - \boldsymbol{\omega}^{\mathcal{B}} \times \boldsymbol{J}\boldsymbol{\omega}^{\mathcal{B}}\right) \end{pmatrix}. \quad (29)$$

Figure 3 depicts the quadrotor free-body diagram: airframe (brown), reference axes (black), rotor thrust vectors (green), control inputs (azure), and rotor spin actuation (orange); gravity and inertial displacement are shown as dashed guides.

$$\mathbf{B} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\Big|_{(\boldsymbol{x}_e, \boldsymbol{u}_e)} = \begin{pmatrix} \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 3} \\ m^{-1}\mathbf{e}^{\mathcal{B}}_z & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 1} & \boldsymbol{J}^{-1} \end{pmatrix} \in \mathbb{R}^{12\times 4}. \quad (30)$$

3 Proposed Approach

We begin by analyzing the linearized dynamics about the hover equilibrium, outlining the outer evolutionary LQG optimization process, and lastly presenting our proposed implementation amongst several state-of-the-art tuning methods.

3.1 Hovering Linearization

Since hovering occurs near equilibrium with small state variables, local behavior can be approximated using small-angle assumptions. This simplifies (26) to $\dot{\boldsymbol{\eta}}^{\mathcal{I}} \approx \boldsymbol{\omega}^{\mathcal{B}}$ and $\mathcal{R}^{\mathcal{I}}_{\mathcal{B}} \approx \boldsymbol{I}_3 + (\boldsymbol{\eta})_\times$, respectively, where $(\cdot)_\times$ denotes the skew-symmetric operator. With these approximations, the Jacobian of the nonlinear dynamics in (29) is computed as

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{pmatrix} \mathbf{0}_3 & \boldsymbol{I}_3 + (\boldsymbol{\eta})_\times & -(\boldsymbol{v}^{\mathcal{B}})_\times & \mathbf{0}_3 \\ \mathbf{0}_3 & -(\boldsymbol{\omega}^{\mathcal{B}})_\times & -(\boldsymbol{g}^{\mathcal{I}})_\times & (\boldsymbol{v}^{\mathcal{B}})_\times \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \nabla_{\boldsymbol{\omega}}(\dot{\boldsymbol{\omega}}^{\mathcal{B}}) \end{pmatrix}, \quad (31)$$

with the corresponding sub-Jacobian of $\dot{\boldsymbol{\omega}}^{\mathcal{B}}$ given by

$$\nabla_{\boldsymbol{\omega}}(\dot{\boldsymbol{\omega}}^{\mathcal{B}}) = \boldsymbol{J}^{-1}\left((\boldsymbol{J}\boldsymbol{\omega}^{\mathcal{B}})_\times - \boldsymbol{\omega}^{\mathcal{B}} \times \boldsymbol{J}\right). \quad (32)$$

Assuming a hovering equilibrium defined by a state vector at a fixed point and a nominal thrust control input

$$\boldsymbol{x}_e = \begin{pmatrix} \boldsymbol{\xi}^{\mathcal{I}}_e & \mathbf{0}_3 & \boldsymbol{\eta}^{\mathcal{I}}_e & \mathbf{0}_3 \end{pmatrix}^\top \in \mathbb{R}^{12}, \quad (33)$$

$$\boldsymbol{u}_e = \begin{pmatrix} mg & 0 & 0 & 0 \end{pmatrix}^\top \in \mathbb{R}^4, \quad (34)$$

substituting this operating point $(\boldsymbol{x}_e, \boldsymbol{u}_e)$ into the state Jacobian yields the 12×12 linearized system dynamics model

$$\mathbf{A} = \begin{pmatrix} \mathbf{0}_3 & \boldsymbol{I}_3 + (\boldsymbol{\eta}_e)_\times & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -(\boldsymbol{g}^{\mathcal{I}})_\times & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{pmatrix}, \quad (35)$$

where higher-order centrifugal terms and their Coriolis couplings vanish. Likewise, linearizing the control input around the thrust required to balance gravity gives Within the state estimation process, these linearized dynamics hold locally around the hover equilibrium $(\boldsymbol{x}_e, \boldsymbol{u}_e)$, enabling state propagation via the state-space model (6).

Table 1: Comparative analysis of weight optimization methods, ordered top-to-bottom by their novelty and precision.

Symbol	Optimization method	Approach	Pros	Cons	Complexity
MT	Manual tuning	Trial-and-error	Trivial implementation	Inefficient (exhaustive)	$O(n_p f_e)$
BR	Bryson's rule [32]	Heuristic	Intuitive; Fast initial guess	Inherently suboptimal	$O(f_e)$
PS	Particle swarm [33]	Swarm-based	Robust against local minima	Premature convergence	$O(k_i n_p f_e)$
GA	Genetic algorithm [34]	Evolutionary	Global search; gradient-free	Computation-intensive	$O(k_i n_p f_e)$
BY	Bayesian [35]	Surrogate-based	Sample-efficient	Scalability issues	$O(k_i f_e + k_i^3)$
CMA	Matrix adaptation [36]	Simulation-based	Robust in rugged search spaces	Slower convergence	$O(k_i n_p f_e)$

http://www.imavs.org/

http://www.imavs.org/

Table 1 lists the six optimization techniques used as benchmarks in this study, distinguished by their key characteristics and ranked in ascending order of complexity. To assess the effectiveness of our covariance matrix adaptation (CMA) implementation—highlighted in green at the bottom—complexity is expressed in terms of number of parameters (n_p), iterations (k_i), and cost per function evaluation (f_e).

3.2 *Outer-Level Optimization*

Gradient-based methods, while effective in high-dimensional settings, are unsuitable here since analytical gradients are unavailable: the cost depends implicitly on closed-loop dynamics. The LQG cost (12) is set a priori by $\mathbf{Q}$ and $\mathbf{R}$, whose iterative tuning forms an inner feedback loop within the outer optimizer. As hovering prioritizes position holding, followed by attitude and yaw stabilization, the outer cost is defined as

$$\mathcal{J}_{\text{out}} = \int_0^T \Big(\|\boldsymbol{\varepsilon}_{\boldsymbol{\xi}}(t)\|_2 + \lambda \|\boldsymbol{\varepsilon}_{\boldsymbol{\eta}}(t)\|_2 \Big) \, \mathrm{d}t \,, \tag{36}$$

where $\lambda \ll 1$ penalizes the translational–rotational error trade-off. For a state-space model (6) with n states and m control inputs, the search spans $n^2 + m^2$ weight parameters. This places the bi-level tuning firmly in the domain of black-box optimization, where weights are iteratively refined without explicit gradients or access to internal system dynamics. Algorithm 1 formalizes the proposed framework in a high-level, optimizer-agnostic form, with the inner LQG control loop nested within the outer weight-optimization process.

Algorithm 1: AERO-LQG bi-level optimization structure

Input: Optimizer $\mathcal{O}$, weight λ, tolerance ϵ
Initialize $\mathcal{O}$ and weights $(\mathbf{Q}, \mathbf{R})$; `// outer-loop`
while *not converged* **do**
  Propose candidate sets $\{(\mathbf{Q}_i, \mathbf{R}_i)\}$ using $\mathcal{O}$;
  foreach *candidate* $(\mathbf{Q}_i, \mathbf{R}_i)$ **do**
    $(\boldsymbol{\xi}(t), \boldsymbol{\eta}(t)) \leftarrow \arg\min \mathcal{J}_{\text{in}}$; `// inner-loop`
  Feed optimal errors $(\boldsymbol{\varepsilon}^*_{\boldsymbol{\xi}}, \boldsymbol{\varepsilon}^*_{\boldsymbol{\eta}})$ into optimizer $\mathcal{O}$;
  if $\mathcal{J}_{out} < \epsilon$ **then**
    break; `// convergence`
return $(\mathbf{Q}^*, \mathbf{R}^*)$;

4 Analysis and Results

To explore the intricacies of LQG tuning in quadrotor hovering, AERO-LQG results are organized into three subsections.

4.1 *Cost Landscape*

When applied to a 12-D state space rich in cross-coupled dynamics, the LQG quadratic cost (12) yields a highly non-convex landscape over the gain parameters. In such settings, gradient-based optimizers are prone to becoming trapped in inferior local minima, with little prospect of escape.

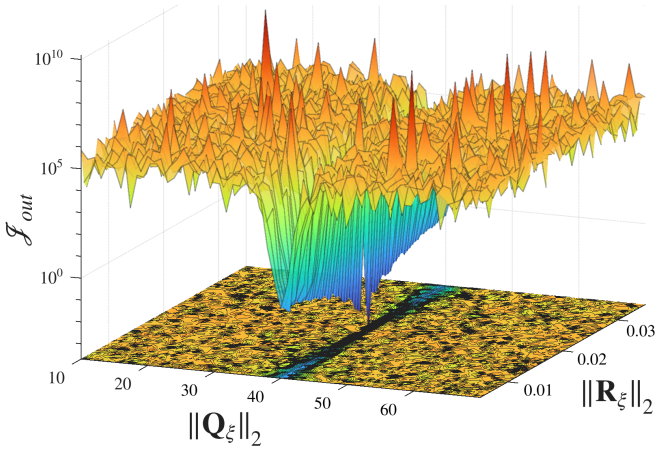

Figure 4: LQG's marginal stability spawns a rugged, minima-rich cost surface, calling for gradient-free optimization.

To demonstrate, we compute the cost of a 10-second standardized gust trajectory, assuming isotropic 3D errors. For clarity, $\mathbf{Q}_\xi$ and $\mathbf{R}_\xi$ utilize standard L_2 norms, while the estimator noise covariance matrices $\mathbf{W}$ and $\mathbf{V}$ remain fixed a priori. Figure 4 illustrates the scalar surface riddled with local minima, exposing the shortcomings of gradient-based techniques. This accentuates the challenge of proper LQG tuning: large regions of the cost surface correspond to poor controller performance, with log-scale costs on the order of $\mathcal{J}_{\text{out}} \gtrsim 10^5$. Yet, precise calibration of $\mathbf{Q}_\xi$ and $\mathbf{R}_\xi$ norms reduces the cost by nearly seven orders of magnitude, achieving near-global optimality and substantially improved performance.

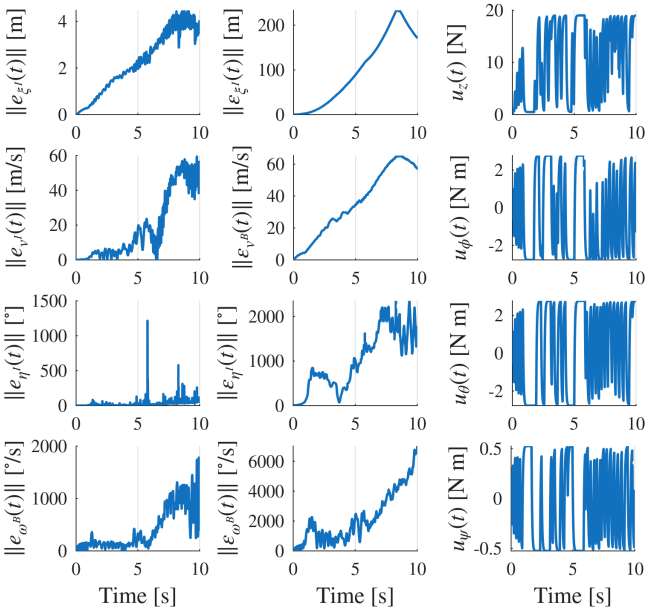

Figure 5: State trajectories under nominal LQG tuning.

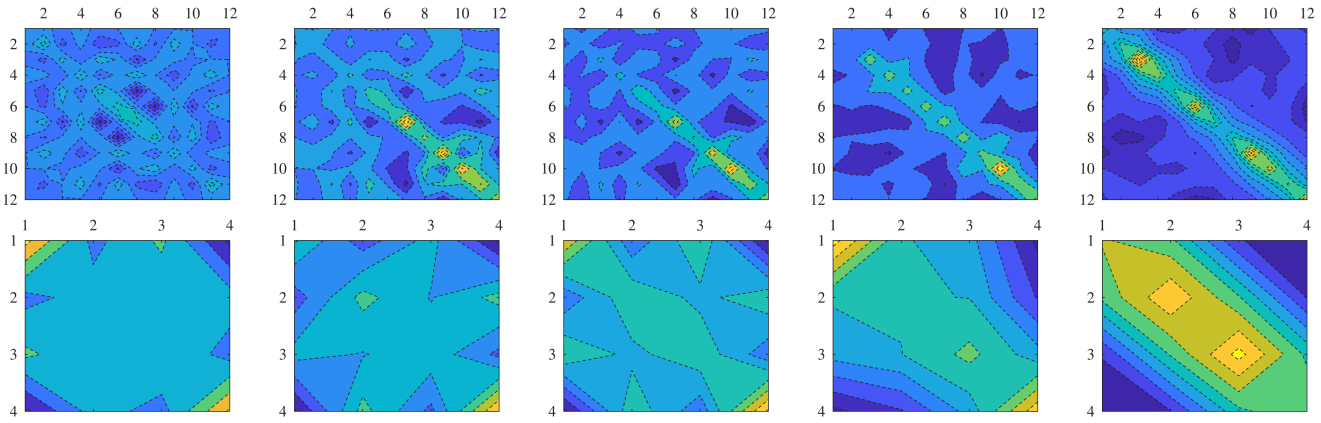

Figure 6: Evolutionary optimization guides $\mathbf{Q}$ (top) and $\mathbf{R}$ (bottom) from random start (left) toward diagonal dominance (right).

4.2 *Robustness Limits under Disturbances*

Figure 5 illustrates the quadrotor states over a 10-second gust impulse [7], with 3D error norms (13) and (14) on the left, and control inputs (27) on the right. Even with careful manual tuning, the system diverges. In delicate hovering, LQG's narrow stability margins make it highly sensitive to disturbances; once error assumptions are violated, divergence grows unchecked. The actuators respond with rapid but ineffective corrections, oscillating between their mechanical limits. This instability stems from the tight coupling between the estimator and controller: small deviations degrade the Kalman filter, initiating a feedback loop that amplifies errors. Unlike LQR, which acts directly on measured states, LQG's reliance on these imperfect state estimates compounds its sensitivity to inaccuracies.

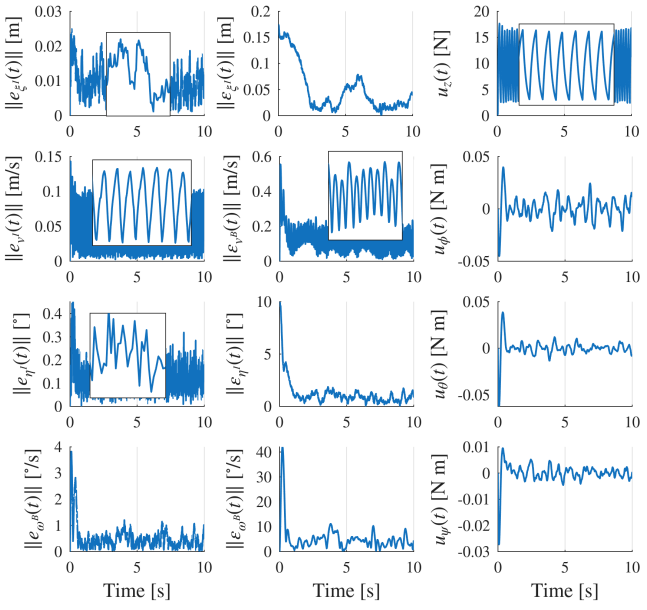

Figure 7: State trajectories under AERO-LQG tuning.

4.3 *Performance Evaluation*

Building on the need for robust, global optimization methods that withstand large perturbations, this section: (i) presents our CMA implementation, (ii) analyzes its performance, and (iii) benchmarks it against several state-of-the-art approaches. Figure 6 illustrates CMA's iterative refinement across five stages, guiding the search distribution from random initialization (left) to convergence (right). Normalized for clarity, both $\mathbf{Q}$ and $\mathbf{R}$ progressively concentrate weight along the diagonal as off-diagonal entries diminish, indicating negligible cross-correlation.

Figure 7 follows the same top-to-bottom error analysis under these finalized parameters. In contrast to the saturated controllers in Figure 5, the optimized trajectories here recover immediately from the same unit disturbance. A residual drift persists, visible as sawtooth patterns in the insets; however, the actuators remain within limits, achieving stabilization with significantly lower control effort (rightmost column). Table 2 provides supporting numbers: MT at the top exhibits complete divergence (∞), illustrating why manual tuning falls short next to automated algorithms.

Table 2: Comparison of optimizers by final-time errors.

Error / Model	Position [m]		Orientation [deg]		Effort [N·s]
	$\|\boldsymbol{e}_{\boldsymbol{\xi}^{\mathcal{I}}}\|$	$\|\boldsymbol{\varepsilon}_{\boldsymbol{\xi}^{\mathcal{I}}}\|$	$\|\boldsymbol{e}_{\boldsymbol{\eta}^{\mathcal{I}}}\|$	$\|\boldsymbol{\varepsilon}_{\boldsymbol{\eta}^{\mathcal{I}}}\|$	$\int_0^T \boldsymbol{u}(t)\,\mathrm{d}t$
MT	$\to\infty$	$\to\infty$	$\to\infty$	$\to\infty$	$\to\infty$
BR	0.48	1.12	0.57	12.3	$>10^3$
PS	0.22	0.77	0.18	6.13	293.8
GA	0.03	0.09	0.22	2.05	119.1
BY	0.31	0.27	0.47	3.21	143.1
CMA	**0.01**	**0.05**	**0.17**	**1.74**	**98.6**

http://www.imavs.org/

In contrast, the optimizers below perform far better, with CMA surpassing the leading GA by reducing estimation and control errors by at least 33% and 55%, respectively. Figure 8 shows these differences across eight flight metrics from [7], benchmarking all candidate optimizers (excluding MT) with color-coded distinction.

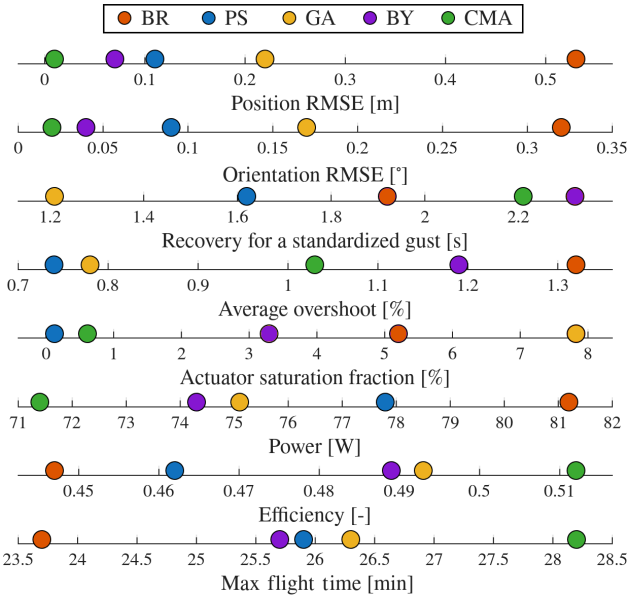

Figure 8: Comparative analysis of the candidate optimizers.

As shown, CMA (green) attains optimality in five of eight criteria, excelling in precision and energy efficiency but lagging in agility and responsiveness. Overall, evolutionary strategies (PS, GA, CMA) consistently outperform heuristic methods (MT, BR) and Bayesian optimization by avoiding premature convergence in dense local minima. These advantages provide tangible benefits, particularly in resource-constrained applications where extended flight endurance is critical.

5 Discussion and Conclusion

This work introduced AERO-LQG, a systematic weight-tuning framework for quadrotor hovering that addresses the narrow stability margins and highly non-convex cost landscapes characteristic of such systems. By decoupling the inner control loop from the outer stochastic optimizer, weight initialization is delegated to a powerful evolutionary strategy that remains robust to complex cost landscapes. Benchmarking across eight flight metrics showed the consistent superiority of our CMA implementation in key hovering capabilities—energy efficiency, flight endurance, and disturbance rejection—a result particularly significant for resource-constrained platforms. While demonstrated on an aerial system, the approach generalizes to other robots, offering resource-aware, high-performance control for real-world demands.

References

[1] Kevin Leahy, Dingjiang Zhou, Cristian-Ioan Vasile, Konstantinos Oikonomopoulos, Mac Schwager, and Calin Belta. Persistent surveillance for unmanned aerial vehicles subject to charging and temporal logic constraints. *Autonomous Robots*, 40(8):1363–1378, 2016.

[2] Han Liang, Seong-Cheol Lee, Woosung Bae, Jeongyun Kim, and Suyoung Seo. Towards UAVs in construction: advancements, challenges, and future directions for monitoring and inspection. *Drones*, 7(3):202, 2023.

[3] Xueping Li, Jose Tupayachi, Aliza Sharmin, and Madelaine Martinez Ferguson. Drone-aided delivery methods, challenge, and the future: A methodological review. *Drones*, 7(3):191, 2023.

[4] Mingyang Lyu, Yibo Zhao, Chao Huang, and Hailong Huang. Unmanned aerial vehicles for search and rescue: A survey. *Remote Sensing*, 15(13):3266, 2023.

[5] Michael W Lewis. Drones and the boundaries of the battlefield. *Tex. Int'l LJ*, 47:293, 2011.

[6] Drew Hanover, Antonio Loquercio, Leonard Bauersfeld, Angel Romero, Robert Penicka, Yunlong Song, Giovanni Cioffi, Elia Kaufmann, and Davide Scaramuzza. Autonomous drone racing: A survey. *IEEE Transactions on Robotics*, 40:3044–3067, 2024.

[7] Daniel Engelsman and Itzik Klein. C-ZUPT: Stationarity-aided aerial hovering. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–15, 2026.

[8] Leonard Bauersfeld and Davide Scaramuzza. Range, endurance, and optimal speed estimates for multicopters. *IEEE Robotics and Automation Letters*, 7(2):2953–2960, 2022.

[9] Chun Fui Liew, Danielle DeLatte, Naoya Takeishi, and Takehisa Yairi. Recent developments in aerial robotics: A survey and prototypes overview. *arXiv preprint arXiv:1711.10085*, 2017.

[10] Nourhan Elmeseiry, Nancy Alshaer, and Tawfik Ismail. A detailed survey and future directions of unmanned aerial vehicles (UAVs) with potential applications. *Aerospace*, 8(12):363, 2021.

[11] Khaled Telli, Okba Kraa, Yassine Himeur, Abdelmalik Ouamane, Mohamed Boumehraz, Shadi Atalla, and Wathiq Mansoor. A comprehensive review of recent research trends on unmanned aerial vehicles (UAVs). *Systems*, 11(8):400, 2023.

[12] Naser El-Sheimy and You Li. Indoor navigation: State of the art and future trends. *Satellite Navigation*, 2(1):7, 2021.

http://www.imavs.org/

[13] Daniel Engelsman and Itzik Klein. Information-aided inertial navigation: A review. *IEEE Transactions on Instrumentation and Measurement*, 72:1–18, 2023.

[14] Alessandro Saviolo and Giuseppe Loianno. Learning quadrotor dynamics for precise, safe, and agile flight control. *Annual Reviews in Control*, 55:45–60, 2023.

[15] Nadav Cohen and Itzik Klein. Inertial navigation meets deep learning: A survey of current trends and future directions. *Results in Engineering*, page 103565, 2024.

[16] Daniel Engelsman and Itzik Klein. Towards learning-based gyrocompassing. *Engineering Applications of Artificial Intelligence*, 163:112842, 2026.

[17] Barak Diker and Itzik Klein. Neural aided adaptive innovation-based invariant kalman filter. *arXiv preprint arXiv:2603.26709*, 2026.

[18] Guy Damari, Zeev Yampolsky, Nadav Cohen, Arup Kumar Sahoo, Jeryes Danial, Felipe O Silva, and Itzik Klein. AI-aided advancements in autonomous underwater vehicle navigation. *arXiv preprint arXiv:2605.04672*, 2026.

[19] Kimon P Valavanis and George J Vachtsevanos. *Handbook of unmanned aerial vehicles*. Springer Publishing Company, Incorporated, 2014.

[20] Labane Chrif and Zemalache Meguenni Kadda. Aircraft control system using LQG and LQR controller with optimal estimation-Kalman filter design. *Procedia Engineering*, 80:245–257, 2014.

[21] David K Arrowsmith and Colin M Place. *An introduction to dynamical systems*. Cambridge university press, 1990.

[22] Amit Levy and Itzik Klein. Adaptive neural unscented Kalman filter. *IEEE Transactions on Intelligent Vehicles*, 11(3):428–437, 2026.

[23] Janis Peksa and Dmytro Mamchur. A review on the state of the art in copter drones and flight control systems. *Sensors*, 24(11):3349, 2024.

[24] Michael V Cook. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Butterworth-Heinemann, 2012.

[25] Karl J Åström. *Introduction to stochastic control theory*. Courier Corporation, 2012.

[26] Daniel Engelsman and Itzik Klein. Inertial-based LQG control: A new look at inverted-pendulum stabilization. *International Journal of Mechanical System Dynamics*, 2025.

[27] NI Mahmudov and A Denker. On controllability of linear stochastic systems. *International Journal of Control*, 73(2):144–151, 2000.

[28] Daniel Engelsman and Itzik Klein. Revisiting certainty equivalence: The structural coupling between estimation and control in underactuated nonlinear systems. *arXiv preprint arXiv:2607.07276*, 2026.

[29] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *AIAA guidance, navigation and control conference and exhibit*, page 6461, 2007.

[30] Sara Sadr, S Ali A Moosavian, and Payam Zarafshan. Dynamics modeling and control of a quadrotor with swing load. *Journal of Robotics*, 2014(1):265897, 2014.

[31] Pedro Castillo, Rogelio Lozano, and Alejandro Enrique Dzul. *Modelling and control of mini-flying machines*. Springer Science & Business Media, 2005.

[32] J.P. Hespanha. *Linear Systems Theory: Second Edition*. Princeton University Press, 2018.

[33] Rabii Fessi and Soufiene Bouallègue. LQG controller design for a quadrotor uav based on particle swarm optimisation. *International Journal of Automation and Control*, 13(5):569–594, 2019.

[34] Abdulla I Abdulla and Ibrahim K Mohammed. Aircraft pitch control design using LQG controller based on genetic algorithm. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 21(2):409–417, 2023.

[35] Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic LQR tuning based on gaussian process global optimization. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 270–277. IEEE, 2016.

[36] Nikolaus Hansen. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

http://www.imavs.org/