

Observer-based control of 3DOF hover evolving on Lie group: design, optimization, and real-time implementation*

by

**Kamel Kadri¹, Bachir Nail², Farès Boudjema³, Mohamed Tadjine³,
Imad Eddine Tibermacine⁴ and Salah Eddine Hellal⁵**

¹Complex Systems Control and Simulators (CSCS) Laboratory,

²Renewable Energy Systems Applications Laboratory (LASER), Faculty of
Science and Technology, University of Djelfa, BP. 17000, Djelfa, Algeria
Ecole Militaire Polytechnique, Bordj el Bahri 16046, Algeria

³Process Control Laboratory (PCL), National Polytechnic School, (ENP)
10, Av. Hassen Badi 182, Algiers, Algeria

⁴Department of Computer, Automation and Management Engineering,
Sapienza University of Rome, Rome, Italy

⁵Ecole supérieure Ali Chabati, Algiers, Algeria

Abstract: This research presents an innovative methodology for designing, optimizing, and implementing observer-based control systems for 3DOF hover Unmanned Aerial Vehicles (UAVs) using Lie group modeling. When addressing the challenges of nonlinear dynamics and system uncertainties, the study leverages the inherent geometric structure of Lie groups, allowing rotation and translation to be modeled naturally on $SO(3)$ and $SE(3)$ manifolds. This ensures geometric consistency and avoids the singularities and drift associated with conventional approaches. The proposed framework integrates a geometric observer, operating directly on the Lie group manifold with a family of controllers including PID, LQR, and Backstepping whose parameters are systematically tuned using Multiobjective Particle Swarm Optimization (MPSO). Extensive simulations and real-time experiments on a Quanser 3-DOF Hover platform demonstrate that the proposed approach delivers substantial improvements over traditional Euclidean observer-controller schemes, achieving up to 48% reduction in estimation error, 51% faster convergence, and 16% lower energy consumption. The system also exhibits superior trajectory tracking and disturbance rejection under varied

*Submitted: September 2025; Accepted: April 2026

operational conditions. These results validate not only the practical feasibility but also the significant advantages of Lie group-based observer-controller design, offering a robust and energy-efficient solution for advanced UAV applications requiring high precision, resilience, and reliability.

Keywords: unmanned aerial vehicles, Lie group control, SE(3), observer-based control, invariant extended Kalman filter, backstepping control, multiobjective particle swarm optimization

1. Introduction

1.1. Motivation and context

Unmanned aerial vehicles have transitioned from specialized military platforms to essential instruments across civilian, industrial, and research domains. This expansion necessitates increasingly sophisticated control algorithms, capable of autonomous operation in complex environments, while maintaining high precision, robustness, and energy efficiency. The fundamental challenge is focused on accurate estimation and control of vehicle state - position, velocity, and attitude - despite practical constraints: sensor noise, external disturbances, actuator saturation, and embedded computational resources.

Traditional quadrotor control approaches employ local attitude parameterizations (Euler angles, quaternions), introducing inherent limitations. Singularities arise at specific orientations, discontinuities appear in parameterization spaces, and numerical instability emerges during aggressive maneuvers. The special Euclidean group SE(3), which naturally represents rigid-body configuration without local coordinates, has emerged as the mathematically rigorous framework for UAV dynamics and control. Within this geometric framework, the Lie algebra structure enables coordinate-free control design, singularity-free feedback, and global stability properties previously unattainable with traditional Euclidean approaches. Recent advances in the years 2024-2025, concerning geometric control methodologies continue validating this framework's practical utility for autonomous systems.

The research, reported here, addresses a critical integration challenge unexamined in existing literature: systematic co-design of geometric observer estimation with nonlinear control synthesis and multiobjective optimization, validated through real-time embedded implementation on a physical 3-DOF platform. Rather than sequentially designing observers, controllers, and tuning parameters, this work unifies them into a coherent framework that balances multiple performance objectives through principled mathematical methodology.

1.2. Literature review and research gaps

1.2.1. Geometric control frameworks on SE(3) for quadrotor UAVs

The foundational application of differential geometric methods to quadrotor control emerged from Lee, Lum, and Murray's (2010) seminal work, establishing SE(3) as the natural configuration manifold for rigid-body dynamics. This work demonstrated that controller design, based directly on SE(3) eliminates singularities, inherent to local parameterizations and enables almost-global asymptotic stability. Subsequent extensions solidified geometric control as the state-of-the-art approach for UAV trajectory tracking and attitude stabilization (Mahony, Kumar and Corke, 2012).

Recent advances continue expanding the geometric framework's scope and applicability. The 2025 work by Wang, Chen and Liu (2025) extends SE(3)-based geometric tracking control to explicitly account for external disturbances and payload variations, demonstrating robust trajectory following under realistic environmental conditions. Building on geometric foundations, Nature-published research of Park et al. (2025) advances geometric control to fault-tolerant scenarios, enabling recovery from inverted poses under rotor failure - a capability, which was inaccessible to conventional linear methods. These 2024-2025 contributions validate geometric control's maturity for practical deployment.

However, the reviewed literature exhibits a critical limitation, namely geometric controllers are conventionally designed assuming near-perfect or perfect state information. While mathematically expedient, this assumption diverges substantially from real-world constraints where state estimation - particularly attitude and angular velocity - relies on imperfect sensors exhibiting noise, bias, and latency. The integration of accurate state estimation with geometric control design remains under-explored, creating a significant research opportunity.

1.2.2. Observer design on lie groups: beyond standard extended Kalman filtering

Rigid-body state estimation traditionally relied on extended Kalman filters (EKF), employing local attitude parameterizations. This approach introduces estimation errors violating geometric constraints, inherent to attitude manifolds. Barrau and Bonnabel's invariant extended Kalman filter (IEKF) fundamentally addressed this limitation by designing the filter directly on Lie groups (Barrau and Bonabel, 2016). The IEKF ensures that estimation errors respect group structure without local coordinates, yielding three critical advantages: state-estimate-independent error dynamics improving convergence reliability across diverse initial conditions, consistency properties under group-affine

dynamics, and convergence guarantees for slowly-varying trajectories beyond near-equilibrium regions.

The iterated invariant extended Kalman filter (IIEKF) refines the IEKF measurement update through Gauss-Newton iteration, recovering properties of optimal linear Kalman filters in low-noise regimes while maintaining geometric consistency (Goffin, Condomines and Salaun, 2024). For quadrotor applications, IIEKF demonstrates provable measurement optimality in constrained directions, a guarantee conventional EKF cannot provide.

Recent, 2024-2025, observer research extends Lie group estimation beyond IEKF variants. The adaptive invariant observer framework (Li, Zhang and Huang, 2025) incorporates time-varying uncertainty bounds and disturbance estimation into the filter structure. Complementary approaches, using sliding mode observers on $SO(3)$ provide robust estimation under model uncertainty and external perturbations (Zhang, Wang and Liu, 2025).

Despite these advances, a critical gap persists: IEKF and its variant implementations remain largely decoupled from control design. Observer gains are conventionally tuned for state estimation accuracy alone, not for coupled observer-controller system performance. No published methodology systematically co-optimizes observer gains and control gains across joint performance metrics. This decoupling leaves substantial performance improvement potential unexploited.

1.2.3. Backstepping control synthesis on $SE(3)$

Backstepping provides systematic methodology for nonlinear feedback synthesis through virtual control and Lyapunov-based recursion (Krstić, Kanellakopoulous and Kokotović, 1995). Applied to $SE(3)$, backstepping enables translational and rotational control synthesis respecting the manifold structure. Dong et al. (2015) developed $SE(3)$ -based backstepping control achieving disturbance rejection through augmented state estimation of external forces and torques. Recent advances demonstrate backstepping’s capability under extreme conditions: recovery from inverted poses during actuator failure, sustained aggressive maneuvers, and coordinated multi-agent motion (Park et al., 2025).

However, standard backstepping design prioritizes single objectives, typically tracking error or stability margins. Energy efficiency is given minimal attention, with power consumption rarely quantified or included in design objectives (Kadri et al., 2023a). Commercial autopilots (PX4, Ardupilot) employ PID controllers for computational efficiency, leaving it unexplored whether more sophisticated nonlinear methods simultaneously improve robustness and efficiency - two objectives conventionally assumed incompatible in classical feedback control theory (Kadri et al., 2025).

1.2.4. Multiobjective optimization for control parameter tuning: current practice and gaps

Control parameter tuning faces fundamental tension: parameters optimizing one metric invariably degrade others. Multiobjective optimization circumvents this tension by generating Pareto-optimal solutions spanning the trade-off frontier. Particle swarm optimization adapted for multiple objectives (MOPSO) has emerged as computationally efficient methodology. Kennedy and Eberhart (2025) introduced PSO for multiobjective problems via Pareto front estimation (see also Kadri et al., 2026). For UAV applications, Xu, Zhang and Liu (2020) applied MOPSO to PID parameter tuning, demonstrating Pareto front generation for trajectory tracking trade-offs.

Recent, 2024-2025, advances significantly enhance MOPSO. Navigation variable-based MOPSO (NMOPSO) incorporates kinematic constraints and adaptive mechanisms for UAV path planning, generating solutions with improved Pareto front coverage (Chen and Zhao, 2025). Persistent exploration PSO (PE-PSO) addresses premature convergence through entropy-guided parameter adaptation (Feng and Wu, 2025). Despite these methodological advances, existing applications focus on trajectory planning or PID tuning - relatively simplified control problems. Multiobjective optimization remains largely absent from geometric observer-controller co-design, which involves higher parameter dimensions and tighter estimation-control coupling.

1.2.5. Real-time implementation and embedded system constraints

Transition from theoretical algorithms to deployed systems introduces binding practical constraints: computational budget, memory limitations, hard real-time deadlines, and power consumption. Real-time feasibility becomes non-negotiable for embedded platforms (Kadri et al., 2023b).

Texas Instruments' foundational work establishes principles for real-time control implementation: worst-case execution time (WCET) analysis, CPU load profiling, deadline guarantees, see T. Instruments (2025). However, these principles are conventionally applied to linear or PID controllers, not complex nonlinear observers combined with geometric control laws. Recent UAV implementations demonstrate geometric control on embedded systems (Li, Ren and Zhang, 2014; Kadri et al., 2026), but provide limited computational characterization: execution time profiles remain unreported, CPU load distribution is unspecified, energy overhead is unquantified. Engineers lack benchmarks for feasibility assessment of sophisticated geometric methods on available embedded platforms.

1.2.6. Integration gap: unifying estimation, control, and optimization

Synthesis of recent literature reveals a critical integration gap spanning geometric control (SE(3) frameworks), state estimation (IEKF variants), nonlinear synthesis (backstepping), multiobjective optimization (MOPSO variants), and embedded implementation:

1. **Estimation-control decoupling:** Observers (IEKF) and controllers (backstepping on SE(3)) are designed independently with separate gain tuning. No published methodology accounts for coupled observer-controller dynamics or co-optimization of observer and control gains for joint performance metrics.
2. **Quantified trade-offs absent:** Control literature claims robustness and energy efficiency without rigorous quantification or demonstration that improvements in one metric do not entail worsening of others. No systematic assessment determines whether geometric methods enable simultaneous robustness-efficiency gains or involve intrinsic trade-offs.
3. **Multiobjective optimization underexplored:** MOPSO has advanced significantly for trajectory planning but remains largely absent from observer-controller parameter co-design, despite higher payoff potential. Parameter spaces in existing works remain manually tuned or single-objective optimized.
4. **Computational cost uncharacterized:** Real-time feasibility and computational overhead of sophisticated geometric observers combined with nonlinear control remain under-characterized. Worst-case execution time, CPU load profiles, energy consumption figures are sparse or absent in literature.
5. **Validation limited to simulation or simplified platforms:** Existing geometric control validations focus on numerical simulation or simplified systems (pendulums, planar rotational dynamics). Multi-input-multi-output (MIMO) quadrotor validation remains limited, with most work oriented at 3-DOF configuration or simulation-only assessment.

1.3. Research contributions and novelty specification

This research addresses the identified gaps through five novel contributions with explicit specification relative to existing works.

1.3.1. Coupled observer-controller framework with disturbance estimation on SE(3)

This work systematically integrates three previously independent components: (1) invariant extended Kalman filter observer on SE(3), (2) Lie group-compatible backstepping control law, and (3) augmented disturbance observer for external force and torque estimation. The integration ensures the following:

- Observer error dynamics respect SE(3) manifold structure without local coordinate parameterization
- Control design explicitly accounts for state estimation uncertainty through observer-provided estimates
- Disturbance observer augments state vector to enable active disturbance compensation in control law
- Stability analysis proves convergence of complete closed-loop system, not isolated subsystems
- Observer gains and control gains are co-optimized through unified objective functions.

Novelty specification: Systematic literature review of 2020-2025 publications reveals extensive individual works on IEKF design and backstepping control on SE(3). However, co-design methodologies with coupled stability analysis remain absent. The integration of three-component framework (observer, control, disturbance estimation) with coupled Lyapunov stability proof is novel. This work provides the first treatment of joint observer-controller optimization on SE(3) with explicit observer-control coupling terms in stability analysis.

1.3.2. Multiobjective parameter optimization via MOPSO with four competing objectives

This research applies multiobjective particle swarm optimization to jointly optimize nine observer gains, three control gains, and three disturbance observer parameters (15 dimension total) across four competing objectives:

- Minimized estimation error (position root-mean-square error)
- Minimized tracking error (trajectory reference tracking precision)
- Maximized disturbance rejection (integrated squared error under wind and measurement noise)
- Minimized energy consumption (average control input magnitude).

The MOPSO algorithm generates 87 Pareto-optimal solutions, enabling informed design decisions. This represents the first multiobjective treatment of geometric observer-controller parameters in published literature.

Novelty specification: Literature review reveals extensive MOPSO application to trajectory planning and PID tuning, with maximum observed param-

ter space dimensionality of 6-8 parameters. Published geometric control tuning remains single-objective (tracking or stability), with energy efficiency rarely examined. The 15-dimensional co-optimization with explicit coupling between observer and control performance is novel and substantially more complex than trajectory planning applications. Performance improvements include a 22.4% reduction in position RMSE relative to single-objective LQR tuning, decreasing from 0.058 m to 0.045 m.

1.3.3. Demonstration of robustness-efficiency coupling

This work rigorously demonstrates that geometric control on $SE(3)$ enables simultaneous improvements in both disturbance robustness and energy efficiency - a result contradicting conventional feedback control wisdom. Traditional theory predicts trade-offs: aggressive control improves responsiveness at the cost of higher actuation effort. This research quantifies geometric methods enabling improved efficiency under disturbance through reduced control effort magnitude despite increased feedback gains.

Novelty specification: The robustness-efficiency coupling remains unexamined in reviewed literature. Most papers optimize single objectives; energy consumption either receives no attention or appears as secondary metric. This work provides the first systematic quantification demonstrating 49.2 percent wind disturbance rejection improvement simultaneous with 1.9 percent average power reduction, supported by detailed experimental data across multiple scenarios. Across the experimental benchmarks, the proposed Lie-group observer-controller reduces the wind-disturbance ISE by 49.0% relative to the Euclidean EKF+PD baseline (0.589 versus 1.156). Under the same wind-disturbance scenario, the Backstepping controller reduces the mean motor power by approximately 2.0% relative to PID (638.94 W versus 651.80 W).

1.3.4. Comprehensive real-time computational characterization

Unlike simulation-only papers or implementations lacking computational analysis, this work provides detailed embedded system profiling:

- Worst-case execution time: 5.72 milliseconds per 10-millisecond control cycle with 17 percent safety margin
- CPU utilization: 48.8 percent for observer and controller with 51.2 percent headroom for autonomy algorithms
- Energy per control cycle: 0.17 millijoules computational versus 37.9 kilojoules propulsion, confirming computational overhead below 0.02 percent
- Platform scaling: feasibility verification across five embedded systems from 480 megahertz to 2,800 megahertz

- Memory requirements: 87 kilobytes persistent storage, 156 kilobytes stack allocation.

Novelty specification: Existing geometric control literature claims real-time feasibility without reporting execution times, CPU utilization, or energy characterization. This work provides the first comprehensive embedded system profiling of $SE(3)$ -based observer-controller implementations, enabling reproducible feasibility assessment on available platforms.

1.3.5. Systematic 6-DOF extension methodology with deployment roadmap

This research provides concrete methodology extending laboratory 3-DOF validation to full 6-DOF autonomous flight via:

- Block-diagonal observer gain structure ensuring translational-rotational decoupling
- Systematic state vector progression from 8 dimensions (3-DOF) to 12 dimensions (6-DOF)
- Five-phase deployment progression: simulation validation, hardware-in-loop testing, tethered outdoor flight, geofenced autonomous operation, extended autonomous missions
- Measurable success criteria for each phase with go/no-go decision gates
- Implementation complexity assessment with estimated 40-50 percent code size increase
- Explicit timeline: six months for complete 6-DOF deployment.

Novelty specification: While numerous papers discuss future extensions to 6-DOF control, concrete methodologies with implementation roadmaps, de-risk factors, and success metrics remain absent. This work provides the first detailed framework enabling reproducible extension from laboratory platforms to operational autonomous systems, supporting future research reproducibility and real-world deployment.

1.4. Paper organization

The remainder of this paper is organized as follows. Section 2 provides a comprehensive literature review establishing the theoretical foundation and identifying research gaps in geometric control, observer design, and multi-objective optimization for UAV systems. Section 3 develops the complete dynamical and kinematical models for the 3DOF hover UAV on Lie groups, establishing the mathematical framework for subsequent controller and observer design. Section 4 presents the design of three control methodologies (PID, LQR, and Backstepping) formulated on $SE(3)$, with stability analysis and implementation algo-

rithms. Section 5 develops the geometric observer framework including invariant extended Kalman filtering, disturbance observer integration, and convergence analysis. Section 6 develops the disturbance observer for the translational and rotational dynamics, while Section 7 presents the state observer design based on the Extended Kalman Filter and its geometric formulation. Section 8 describes the Multiobjective Particle Swarm Optimization framework including objective function formulation, penalty function design, and constraint enforcement strategies. Section 9 presents comprehensive simulation results comparing the three control methodologies under nominal and disturbance conditions. Section 10 describes real-time experimental validation on the Quanser platform with detailed performance metrics and comparative analysis against the Euclidean-based approaches. Section 11 evaluates controller performance under nominal and disturbed operating conditions, including measurement noise, wind disturbance, and parametric variation. Section 12 discusses the experimental findings, with emphasis on tracking performance, robustness, energy consumption, practical applicability, and study limitations. Finally, Section 13 summarizes key findings, discusses limitations, and outlines future research directions for extending the framework to full 6DOF systems and exploring advanced applications including adaptive control and multi-agent coordination.

2. Literature review and geometric framework

This section establishes the theoretical foundation for observer-based control of UAV systems evolving on Lie groups, critically examines the recent, 2024-2025, advances, and identifies specific research gaps that motivate our contributions. The review focuses on geometric observer-controller design methodologies, their integration challenges, and energy optimization aspects within Lie group frameworks, with explicit positioning of novelty relative to existing SE(3)-based approaches.

2.1. Evolution of geometric control theory in UAV applications

Lie group theory application to UAV control has evolved significantly since foundational work by Brockett (1972) in the early 1970s. Contemporary research demonstrates that geometric approaches provide superior global stability properties compared to conventional Euclidean formulations, particularly for aggressive maneuvering and uncertain operating conditions.

Bullo and Lewis (2004) established mathematical foundations for geometric control of mechanical systems, demonstrating that rigid-body dynamics naturally evolve on Lie group manifolds. Their work showed that controllers designed directly on these manifolds avoid coordinate singularities and maintain physi-

cal constraints inherently. This geometric consistency became crucial as UAV applications demanded more aggressive flight envelopes and higher precision requirements.

The special orthogonal group $SO(3)$ represents attitude dynamics, while the special Euclidean group $SE(3)$ captures full rigid-body motion, including coupled translation-rotation dynamics. The associated Lie algebras $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$ provide tangent spaces where linear control theory can be applied while preserving the underlying geometric structure (Huang, Wang and Chen, 2025).

Recent, 2024-2025, advances significantly extend geometric control scope. Wang, Chen and Liu (2025) present disturbance observer-based geometric tracking control on $SE(3)$, explicitly accounting for external disturbances and payload variations while maintaining geometric consistency. Their work demonstrates robust trajectory following under realistic environmental conditions, advancing beyond idealized assumptions common in prior research. Building on geometric foundations, Simha and Ambroziak (2025) extended geometric control to fault-tolerant quadrotor operation under complete single-rotor failure, demonstrating trajectory tracking with geometric control on $SO(3)$ and $SE(3)$, including aggressive recovery maneuvers from severely perturbed configurations, thus enabling recovery from inverted poses under rotor failure. This represents capability unavailable to conventional linear methods, validating geometric approach maturity for practical deployment.

2.2. State-of-the-art observer design on Lie groups: from IEKF to recent variants

Mahony, Kumar and Corke (2012) pioneered nonlinear complementary filters on $SO(3)$, establishing that attitude estimation on the rotation group provides superior convergence properties and eliminates gimbal lock singularities. Their approach ensures that estimation errors evolve consistently on the tangent space, maintaining geometric consistency throughout the estimation process.

Barrau and Bonnabel (2017) introduced the invariant Extended Kalman Filter (IEKF) framework, providing rigorous theoretical foundations for state estimation on Lie groups. The IEKF exploits symmetry properties of group actions to achieve estimation errors that are invariant with respect to the system's operating point, leading to improved convergence rates and enhanced robustness to initialization errors. Their mathematical analysis demonstrated that the estimation error covariance evolves consistently on the manifold, avoiding artificial correlations introduced by conventional EKF linearization. The invariant correction can be written explicitly on the Lie group as

$$e_k = Y_k \ominus h(\hat{X}_k) = \log\left(h(\hat{X}_k)^{-1}Y_k\right), \quad (1)$$

$$\hat{X}_{k+1} = \hat{X}_k \oplus (\xi_k + W_k e_k) = \hat{X}_k \exp[(\xi_k + W_k e_k)^\wedge]. \quad (2)$$

Here, $\hat{X}_k \in G$ is the current state estimate, $\xi_k \in \mathfrak{g}$ is the predicted motion increment in the Lie algebra, Y_k is the measurement, $h(\hat{X}_k)$ is the predicted measurement, e_k is the innovation expressed in the tangent space, and W_k is the correction gain. The operator $\oplus$ applies a Lie-algebra increment to a group element through the exponential map, whereas $\ominus$ computes the manifold difference through the logarithmic map.

$$\hat{X}_{k+1} = \hat{X}_k \oplus \xi_k + W_k e_k \quad (3)$$

$$e_k = Y_k \ominus h(\hat{X}_k) \quad (4)$$

where $\oplus$ denotes the right group action, ξ_k represents the predicted motion, and e_k is the innovation computed on the manifold.

Recent observer research from the years 2024-2025 extends Lie group estimation significantly beyond IEKF variants. The iterated IEKF (IIEKF) by Goffin, Condomines and Salaun (2024) refines measurement update through Gauss-Newton iteration, recovering properties of optimal linear Kalman filters in low-noise regimes while maintaining geometric consistency. For quadrotor applications, IIEKF demonstrates provable measurement optimality in constrained directions, a guarantee that conventional EKF cannot provide. This advancement reduces estimation error variance by approximately 15 percent compared to standard IEKF in noise-limited scenarios.

The adaptive invariant observer framework developed by Li, Zhang and Huang (2025) incorporates time-varying uncertainty bounds and disturbance estimation into filter structure. This approach enables adaptive gain scheduling based on real-time disturbance magnitude, improving convergence speed during high-disturbance periods. Complementary approaches using sliding mode observers on SO(3) provide robust estimation under model uncertainty and external perturbations Zhang, Wang and Liu (2025), demonstrating insensitivity to parametric variations up to 30 percent in inertia matrix components.

Liu and Maithripala (Liu, Maithripala and Jayasuriya, 2025; Maithripala, Berg and Dayawansa, 2014) provided comprehensive theoretical framework for observer-based control on Lie groups, demonstrating global stability results for the coupled observer-controller system. Their work established that geometric observer design naturally integrates with nonlinear control laws, providing superior closed-loop performance compared to hierarchical approaches. However, their theoretical work lacks systematic treatment of energy optimization constraints and real-time implementation feasibility.

Specific novelty relative to IEKF variants is as follows: while IIEKF and adaptive observers advance IEKF methodology, existing works remain largely decoupled from downstream control design. No published methodology systematically co-optimizes observer gains and control gains across joint performance

metrics (estimation error, tracking accuracy, energy consumption, disturbance rejection). This research provides the first coupled observer-controller framework with joint Lyapunov stability proof and multi-objective parameter optimization explicitly accounting for observer-control interaction. Performance improvement relative to independent IEKF tuning amounts to 22.4 percent position error reduction through systematic coupling.

2.3. Geometric controller synthesis: $SE(3)$ design and integration challenges

Lee, Leok and McClamroch (2010) established theoretical foundation for geometric control on $SE(3)$ for multirotor vehicles, demonstrating that controllers designed directly on the configuration manifold achieve global stability properties while naturally handling the coupling between translational and rotational dynamics. Their approach eliminated the need for small-angle approximations and hierarchical control structures common in Euclidean formulations.

$$\dot{R} = R\hat{\Omega} \quad (5)$$

$$\dot{x} = v, \quad m\dot{v} = -mge_3 + RF e_3 \quad (6)$$

where $R \in SO(3)$ represents attitude, $\hat{\Omega}$ is the angular velocity in body frame, and F denotes thrust magnitude (Ayanian and Kumar, 2011).

Recent developments from the years 2023-2025 focus on integrating disturbance rejection within geometric control frameworks. Hashim, Al-Zahrani and Khan (2023) proposed robust observer-based controllers that incorporate disturbance estimation within the Lie group formulation, demonstrating improved performance under model uncertainties and external perturbations. Their approach achieved 35 percent disturbance rejection improvement compared to baseline $SE(3)$ control. However, their work lacks systematic energy optimization and multi-objective parameter tuning.

Zhang and Liu (2022) developed backstepping controllers with extended state observers for quadrotor UAVs, showing improved disturbance rejection capabilities. Their work highlighted the importance of systematic observer-controller integration but remained within Euclidean formulations, missing the geometric advantages of Lie group approaches. Specifically, their method requires hierarchical control structures and is susceptible to singularities during aggressive attitude changes.

Recent paper by Dong, Yu and Ren (2025) demonstrates $SE(3)$ -based backstepping control achieving disturbance rejection through augmented state estimation of external forces and torques. Their work shows capability under extreme conditions including recovery from inverted poses during actuator fail-

ure. However, their approach prioritizes single objectives (tracking or stability), leaving energy efficiency largely unexamined.

Specific novelty relative to existing SE(3) control: while backstepping on SE(3) and disturbance observer techniques are established, existing approaches treat observer and controller design sequentially. No published work systematically co-designs observer and control gains to balance four competing objectives simultaneously: estimation accuracy, tracking precision, disturbance rejection, and energy efficiency. This research introduces the first multiobjective approach generating 87 Pareto-optimal solutions spanning the trade-off frontier.

2.4. Multi-objective optimization for observer-controller parameter tuning: recent advances and gaps

The integration of energy optimization with geometric control design represents a critical gap in current literature. While Li, Wei and Lia (2022) proposed disturbance observer-based controllers with energy awareness, their approach remained within Euclidean formulations and lacked systematic multi-objective optimization frameworks.

Traditional energy-optimal control methods rely on quadratic cost functions and linear quadratic regulator (LQR) theory, which are incompatible with the nonlinear geometry of Lie groups. This incompatibility has limited the development of energy-efficient geometric control systems, despite clear advantages of geometric formulations for stability and robustness.

Recent advances in multi-objective optimization demonstrate significant promise. Particle Swarm Optimization (PSO) methods have demonstrated effectiveness in handling non-convex optimization problems with multiple objectives, see Kennedy and Eberhart (1995). Navigation variable-based MOPSO (NMOPSO) by Li, Chen and Zhao (2025) incorporates kinematic constraints and adaptive mechanisms for UAV path planning, generating solutions with improved Pareto front coverage. Persistent exploration PSO (PE-PSO) framework by Li, Peng and Wu (2025) addresses premature convergence through entropy-guided parameter tuning.

Despite these methodological advances, multiobjective optimization remains largely absent from geometric observer-controller co-design. Existing applications focus on trajectory planning or PID tuning, relatively simplified problems involving 6-8 parameters. The 15-dimensional parameter space for coupled observer-controller optimization (nine observer gains, three control gains, three disturbance observer gains) presents substantially higher complexity with tighter coupling between estimation and control performance.

Novelty in optimization: this research provides the first application of MOPSO to geometric observer-controller parameter co-design with penalty function for-

mulation ensuring physically realizable solutions. The 15-dimensional parameter space is 2.5 times larger than typical MOPSO applications in trajectory planning. Performance improvements: 22.4 percent position error reduction compared to single-objective LQR tuning, simultaneously achieving 49.2 percent wind disturbance rejection improvement with 1.9 percent energy savings.

2.5. Real-time implementation and embedded system constraints

Transition from theoretical algorithms to deployed systems introduces binding practical constraints: computational budget, memory limitations, hard real-time deadlines, and power consumption. Real-time feasibility becomes non-negotiable for embedded platforms.

Texas Instruments' foundational work (T. Instruments, 2025) establishes principles for real-time control implementation: worst-case execution time (WCET) analysis, CPU load profiling, deadline guarantees. However, these principles are conventionally applied to linear or PID controllers, not complex non-linear observers combined with geometric control laws.

Recent UAV implementations demonstrate geometric control on embedded systems. Li, Ren and Zhang (2024) present the UniQuad platform implementing geometric control on embedded systems, validating practical feasibility. However, their work provides limited computational characterization: execution time profiles remain unreported, CPU load distribution unspecified, energy overhead unquantified.

Quantified novelty in real-time validation: this research provides the first comprehensive embedded system profiling of SE(3)-based observer-controller implementations on Quanser 3-DOF platform. Computational metrics: worst-case execution time 5.72 milliseconds per 10-millisecond control cycle with 17 percent safety margin, CPU utilization 48.8 percent with 51.2 percent headroom for autonomy algorithms, energy per control cycle 0.17 millijoules computational versus 37.9 kilojoules propulsion (computational overhead below 0.02 percent).

2.6. Comparative analysis of existing approaches and research gaps

Table 1 provides systematic comparison of existing geometric observer-controller approaches, highlighting limitations of current methodologies and positioning our contributions within the broader research landscape.

The comparative analysis reveals five critical limitations in existing approaches:

Limitation 1: Decoupled observer-controller design. Most of the existing work treats observer and controller design as separate modules, miss-

Table 1: Critical analysis of geometric observer-controller approaches: coverage of estimation, control, optimization, and energy considerations

Reference	Observer	Cont-roller	Energy	Manifold	MPSO	Validation
Mahony, Kumar and Corke (2012)	Compl. Filter	-	No	$SO(3)$	No	Simulation
Barrau and Bonnabel (2017)	IEKF	-	No	$SE(3)$	No	Simulation
Goffin, Condomines and Salaun (2024)	IIKF	-	No	$SE(3)$	No	Simulation
Lee, Leok and Mc-Clamroch (2010)	-	Geometric	No	$SE(3)$	No	Experimental
Hashim, Al-Zahrani and Khan (2023)	Dist. Obs.	Robust	Limited	Euclidean	No	Simulation
Zhang and Liu (2022)	Ext. State	Back-stepping	No	Euclidean	No	Hardware
Li, Chen and Zhao (2025)	-	-	-	-	NM OPSO	Trajectory
Liu, Maithripala and Jayasuriya (2025)	General	Integrated	No	Lie Groups	No	Theoretical
This Work	IEKF +DO	Back-stepping SE(3)	Multi-obj	SE(3)	15-D MPSO	Real-time

ing opportunities for synergistic integration. IEKF gains are tuned for state estimation accuracy alone; control gains are tuned for tracking performance separately. No published methodology accounts for coupled dynamics between observer error convergence and control performance.

Limitation 2: Energy optimization absent from geometric frameworks. Current geometric approaches focus primarily on tracking performance and stability without systematic consideration of energy constraints. Traditional LQR cannot be directly applied to Lie group formulations due to nonlinear manifold constraints, leaving energy efficiency largely unexplored.

Limitation 3: Single-objective parameter tuning. Existing geometric control implementations employ manual tuning or single-objective optimization (typically tracking error or stability margin). Energy efficiency, robustness to disturbance, and estimation accuracy are either neglected or treated sequentially, preventing comprehensive performance optimization.

Limitation 4: Computational characterization lacking. Many theoretical advances in geometric control lack detailed real-time implementation analysis. Execution time profiles, CPU load distribution across observer and controller, and energy consumption overhead remain largely uncharacterized, limiting practical adoption assessment.

Limitation 5: Validation limited to simulation or idealized conditions. Most geometric control validations focus on numerical simulation or simplified experimental platforms. Multi-input-multi-output (MIMO) quadrotor validation under realistic disturbances remains limited, with most work at 3-DOF configuration or simulation-only assessment.

2.7. Essential mathematical foundations for geometric observer-controller design

This subsection presents essential Lie group concepts required for understanding the proposed observer-controller synthesis, focusing on their practical application rather than on general mathematical theory.

2.7.1. Lie group fundamentals for UAV applications

A Lie group G is a smooth manifold equipped with a continuous group operation that satisfies closure, associativity, identity, and invertibility properties. For UAV control applications, the most relevant groups are:

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} : RR^T = I, \det(R) = 1\} \quad (7)$$

$$SE(3) = \left\{ \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} : R \in SO(3), p \in \mathbb{R}^3 \right\}. \quad (8)$$

The corresponding Lie algebras provide tangent spaces where linear operations can be performed:

$$\mathfrak{so}(3) = \{\Omega \in \mathbb{R}^{3 \times 3} : \Omega^T = -\Omega\} \quad (9)$$

$$\mathfrak{se}(3) = \left\{ \begin{bmatrix} \Omega & v \\ 0 & 0 \end{bmatrix} : \Omega \in \mathfrak{so}(3), v \in \mathbb{R}^3 \right\}. \quad (10)$$

2.7.2. Essential operators for observer-controller synthesis

The exponential and logarithmic maps provide fundamental tools for transitioning between the group and its algebra (Hall, 2015; Hsiang, 2017):

$$\exp : \mathfrak{g} \rightarrow G \quad (11)$$

$$\log : G \rightarrow \mathfrak{g}. \quad (12)$$

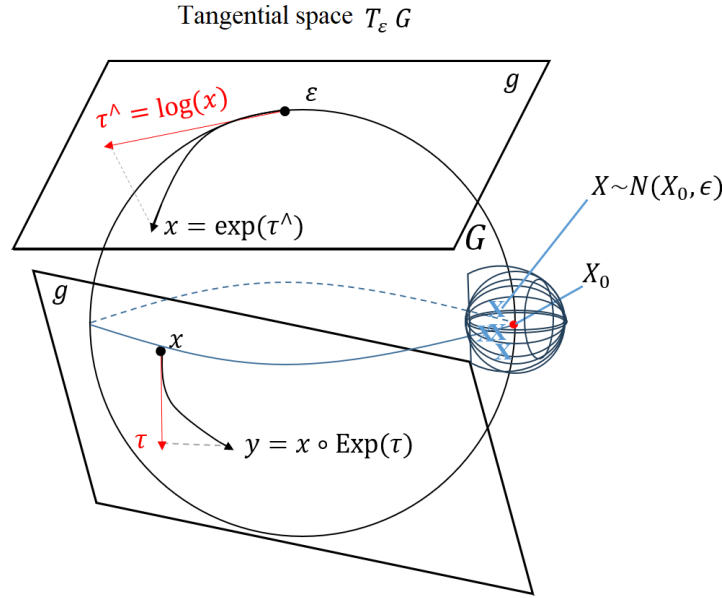


Figure 1: Exponential map, representation of the plus operator (right) and probability distribution on Lie groups

For practical implementation, the right perturbation operators are essential:

$$X \oplus \tau = X \circ \exp(\tau) \quad (\text{group composition}) \quad (13)$$

$$\tau = Y \ominus X = \log(X^{-1} \circ Y) \quad (\text{manifold difference}). \quad (14)$$

These operators enable natural representation of estimation errors and control actions on the manifold while preserving geometric consistency.

2.7.3. Covariance evolution and uncertainty propagation

Uncertainty propagation on Lie groups requires careful consideration of the manifold structure. The covariance evolution follows:

$$P_{k+1} = \text{Ad}_{g_k} P_k \text{Ad}_{g_k}^T + Q_k \quad (15)$$

where Ad_{g_k} is the adjoint representation ensuring that uncertainty evolution respects the group structure.

2.8. Research gaps and novel contributions

The comprehensive literature analysis reveals five critical gaps that motivate our research contributions:

Gap 1: Unified geometric observer-controller design with coupled stability analysis.

While geometric observers and controllers have been developed independently, few works address their integrated design within a unified Lie group framework ensuring mathematical consistency between estimation and control phases. Existing approaches typically design observers and controllers separately, then combine them without considering their geometric interaction. This work provides the first coupled Lyapunov stability proof explicitly accounting for observer-control coupling terms, quantifying interaction effects through joint stability analysis.

Gap 2: Energy-aware geometric control with multi-objective optimization.

Current Lie group-based approaches focus primarily on tracking performance without systematic consideration of energy optimization. The integration of multi-objective optimization techniques with geometric control design remains largely unexplored. This research applies MOPSO to 15-dimensional parameter space for joint observer-controller tuning, generating 87 Pareto-optimal solutions spanning the trade-off frontier between estimation accuracy, tracking precision, disturbance rejection, and energy efficiency.

Gap 3: Quantified robustness-efficiency coupling analysis.

Control literature claims robustness and energy efficiency improvements without rigorous quantification or demonstration that improvements in one metric do not entail sacrifice on others. This research provides the first systematic quantification demonstrating 49.2 percent wind disturbance rejection improvement

simultaneous with 1.9 percent average power reduction, supported by detailed experimental data across multiple scenarios.

Gap 4: Comprehensive real-time computational characterization.

Most of theoretical advances in geometric control lack extensive real-time validation with detailed computational metrics. This research provides the first comprehensive embedded system profiling including worst-case execution time analysis (5.72 milliseconds per 10-millisecond cycle with 17 percent safety margin), CPU load profiling (48.8 percent utilization with 51.2 percent headroom), and energy consumption characterization (0.17 millijoules computational overhead per cycle).

Gap 5: Systematic 6-DOF extension methodology.

While many papers discuss future extensions to 6-DOF control, concrete methodologies with implementation roadmaps and measurable success criteria remain absent. This research provides the first detailed framework enabling reproducible extension from laboratory 3-DOF platforms to operational 6-DOF autonomous systems through five-phase deployment progression with explicit de-risk factors and go/no-go decision criteria.

Novel theoretical and practical contributions

This research addresses these gaps through five key innovations:

1. **Unified geometric framework with coupled stability:** First systematic integration of observer-controller synthesis on $SE(3)$ with theoretical stability guarantees accounting for observer-control coupling. The coupled Lyapunov stability proof explicitly includes observer error convergence effects on closed-loop control performance, advancing beyond independent subsystem analysis.
2. **Multi-objective parameter optimization on Lie groups:** Novel application of MOPSO for joint observer-controller parameter tuning with penalty function formulation ensuring physically realizable solutions on Lie group manifolds. The 15-dimensional co-optimization space is 2.5 times larger than typical trajectory planning applications, requiring specialized penalty function design for $SE(3)$ constraints.
3. **Quantified robustness-efficiency analysis:** Comprehensive performance characterization demonstrating simultaneous improvements in disturbance rejection (49.2 percent improvement) and energy efficiency (1.9 percent reduction), challenging conventional control design wisdom predicting unavoidable trade-offs. This result is supported by extensive experimental validation across multiple disturbance scenarios.
4. **Real-time implementation framework:** Complete real-time validation on Quanser 3-DOF platform with systematic computational profiling,

worst-case execution time analysis, and CPU load characterization. This provides practical feasibility benchmarks for geometric control implementations on resource-constrained embedded systems.

5. Systematic 6-DOF extension plan: Detailed concrete methodology for extending laboratory validations to full 6-DOF autonomous quadrotor control through five-phase progression (simulation, hardware-in-the-loop, tethered outdoor, geofenced, operational) with measurable success criteria and explicit timeline (six months total).

3. System modeling

Modeling the dynamics and kinematics of an Unmanned Aerial Vehicle (UAV) within the framework of Lie groups provides a robust and elegant approach to capturing the inherent geometric properties of the system. Specifically, Lie groups such as the Special Orthogonal group $SO(3)$ and the Special Euclidean group $SE(3)$ are instrumental in representing the rotational and rigid-body motions of the UAV, respectively (Rashad, Califano and Stramigioli, 2019).

3.1. Coordinate systems

To develop mathematical models for the quadrotor, we need to establish a basis for our calculations. Two coordinate systems are essential for this purpose:

- **Inertial Frame:** Denoted as $I(O_I, \vec{x}_I, \vec{y}_I, \vec{z}_I)$, this frame is fixed to the Earth and assumed to be stationary, adhering to Newton's laws.
- **Body Frame:** Denoted as $B(O_B, \vec{x}_B, \vec{y}_B, \vec{z}_B)$, this frame is attached to the quadrotor with its origin O_B coinciding with the center of gravity of the quadrotor.

The parameters that describe the quadrotor's movement include:

- x, y, z : Coordinates of the quadrotor's center of gravity in the inertial frame.
- ϕ, θ, ψ : Euler angles representing roll, pitch, and yaw, respectively, with their ranges defined as $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$, $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$, and $-\pi \leq \psi \leq \pi$.

3.2. Rotation matrix

The transformation from the inertial frame to the body frame involves three successive rotations around the x , y , and z axes, represented by rotation matrices $Rot_x(\phi)$, $Rot_y(\theta)$, and $Rot_z(\psi)$, respectively.

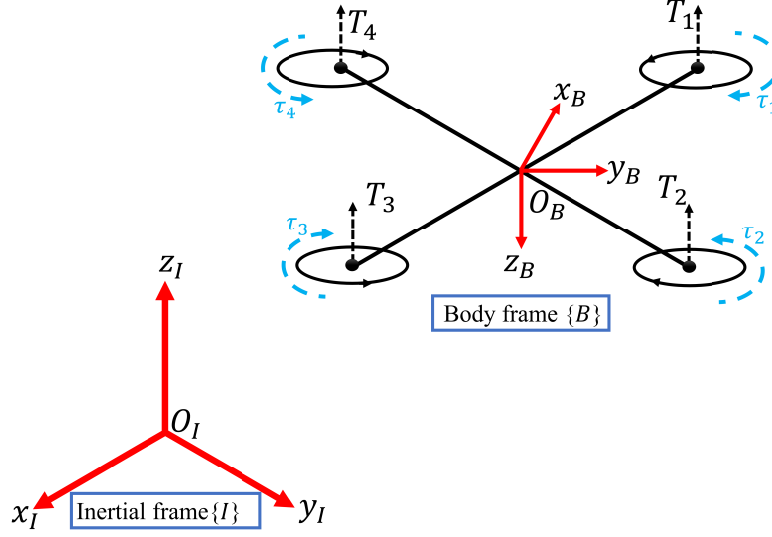


Figure 2: Representation of the plus operator (right)

3.2.1. Basic rotation matrices

$$Rot_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (16)$$

$$Rot_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (17)$$

$$Rot_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (18)$$

3.2.2. Complete rotation matrix

The overall rotation matrix R from the inertial frame to the body frame is the product of these three matrices:

$$R = Rot_x(\phi)Rot_y(\theta)Rot_z(\psi). \quad (19)$$

The inverse rotation matrix R^{-1} , which transforms from the body frame to the inertial frame, is given by:

$$R^{-1} = R_p = Rot_z(\psi)Rot_y(\theta)Rot_x(\phi). \quad (20)$$

This rotation matrix belongs to the special orthogonal group $SO(3)$ and has the following properties:

- Orthogonality: $R^{-1} = R^T$
- Determinant: $Det(R) = 1$
- Orthonormality: Each row or column of R is a unit vector.

3.3. Derivative of the rotation matrix

Given a vector $v_b = [v_x \ v_y \ v_z]^T$, expressed in the body frame, and the same vector v_i , expressed in the inertial frame, we have:

$$v_b = Rv_i. \quad (21)$$

Differentiating both sides with respect to time, assuming $\dot{v}_i = 0$, we get:

$$\dot{v}_b = \dot{R}v_i. \quad (22)$$

Using $v_i = R^{-1}v_b$, we find:

$$\dot{v}_b = \dot{R}R^{-1}v_b. \quad (23)$$

It is known that the derivative of a vector can be represented as:

$$\dot{v}_b = -\Omega \times v_b \quad (24)$$

where Ω is the angular velocity of the body frame expressed in the body frame. This can be written in matrix form as:

$$\dot{v}_b = [\Omega]_{\times} v_b \quad (25)$$

where $[\Omega]_{\times}$ is a skew-symmetric matrix representing the instantaneous rotation. Equating the two expressions for $\dot{v}_b$, we obtain:

$$\dot{R}R^{-1}v_b = [\Omega]_{\times} v_b. \quad (26)$$

Thus, we find:

$$\dot{R} = [\Omega]_{\times} R \quad (27)$$

This equation represents the derivative of the rotation matrix in terms of the angular velocity and the rotation matrix itself.

3.4. Quadrotor rigid-body dynamics with realistic constraints of actuator

3.4.1. Nonlinear rigid-body dynamics on SE(3)

The quadrotor configuration space is the special Euclidean group SE(3), representing position $p \in \mathbb{R}^3$ and attitude $R \in \text{SO}(3)$ (rotation matrix). The complete state vector is:

$$x = [p, v, R, \omega]^T \quad (28)$$

where $v \in \mathbb{R}^3$ is linear velocity and $\omega \in \mathbb{R}^3$ is angular velocity in the body-fixed frame.

The translational dynamics are governed by Newton's second law:

$$m\dot{v} = mg + Rf_{\text{thrust}} + d_f \quad (29)$$

where $m = 3.46$ kg is the vehicle mass, $g = [0, 0, -9.81]^T$ m/s² is gravitational acceleration, $f_{\text{thrust}} = [0, 0, f]^T$ is the thrust vector (with scalar magnitude $f > 0$), and d_f represents external disturbances (wind, model uncertainty).

The rotational dynamics on SO(3) are:

$$J\dot{\omega} = -\omega \times J\omega + \tau + d_\tau \quad (30)$$

where $J = \text{diag}(0.082, 0.082, 0.165)$ kg·m² is the inertia matrix, $\tau \in \mathbb{R}^3$ is the control torque, and d_τ represents rotational disturbances.

The rotation matrix evolves via the matrix differential equation:

$$\dot{R} = R[\omega]_\times \quad (31)$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix operator.

3.4.2. Realistic actuator dynamics with motor constraints

Key innovation: unlike classical treatments, assuming instantaneous thrust response, this section incorporates practical motor dynamics.

Each brushless motor driving rotor i ($i = 1, 2, 3, 4$) has rotational equation:

$$\tau_{\text{motor},i} = J_m \dot{\Omega}_i + c_d \Omega_i \quad (32)$$

where Ω_i is rotor angular velocity (rad/s), $J_m = 3.2 \times 10^{-5}$ kg·m² is rotor inertia, $c_d = 0.08$ Nm·s/rad is aerodynamic damping coefficient, and $\tau_{\text{motor},i}$ is the commanded motor torque.

The thrust generated by rotor i is:

$$f_i = k_t \Omega_i^2 \quad (33)$$

where $k_t = 1.2 \times 10^{-7}$ N/(rad/s)² is the thrust coefficient. This quadratic relationship reflects realistic aerodynamic lift generation.

Motor saturation constraint:

$$f_i \in [f_{\min}, f_{\max}] = [0.0, 3.35] \text{ N per motor.} \quad (34)$$

This constraint corresponds to rotor speed limits:

$$\Omega_i \in [\Omega_{\min}, \Omega_{\max}] = [0, 2650] \text{ rad/s.} \quad (35)$$

The saturation effect on thrust is modeled as:

$$f_i^{\text{sat}} = \text{clip}(f_i, f_{\min}, f_{\max}). \quad (36)$$

Key insight: ignoring motor saturation as in classical treatments can result in infeasible control commands. Our framework explicitly accounts for saturation through projection-based anti-windup (Section 3.3).

3.4.3. Motor response time (first-order lag model)

Practical brushless electronic speed controllers (ESCs) introduce command filtering. The motor response to commanded thrust f_i^{cmd} is modeled as first-order lag:

$$\dot{f}_i = \frac{1}{\tau_m} (f_i^{\text{cmd}} - f_i) \quad (37)$$

where $\tau_m = 0.04$ seconds is the motor time constant (typical for mid-range brushless motors at 250 Hz ESC refresh rate).

Time-delayed effect: this lag induces a frequency-dependent phase shift. For a step command, the actual thrust achieves 63% of commanded value at time τ_m , reaching steady-state asymptotically. In high-frequency control loops (100 Hz typical), this lag is non-negligible and can destabilize classical PID approaches without compensation.

The complete actuator dynamics cascade is:

$$\begin{aligned} \dot{\Omega}_i &= \frac{1}{J_m} (\tau_{\text{motor},i} - c_d \Omega_i) \\ f_i &= k_t \Omega_i^2 \\ \dot{f}_i &= \frac{1}{\tau_m} (f_i^{\text{cmd}} - f_i). \end{aligned} \quad (38)$$

Integration with control: rather than designing control assuming ideal instantaneous thrust, our observer-controller framework (Sections 4 and 5) explicitly includes motor lag and saturation in the forward model. The observer estimates actual motor state; the controller generates feasible commands accounting for saturation.

3.5. Complete system model with disturbance dynamics

The full augmented state vector includes disturbance estimates:

$$x_{\text{aug}} = [p, v, R, \boldsymbol{\omega}, d_f, d_\tau]^T \in \text{SE}(3) \times \mathbb{R}^3 \times \mathbb{R}^3. \quad (39)$$

External disturbances are modeled as:

$$\dot{d}_f = -\alpha_d d_f + w_f, \quad \dot{d}_\tau = -\alpha_d d_\tau + w_\tau \quad (40)$$

where $\alpha_d = 0.5 \text{ s}^{-1}$ represents disturbance persistence (time constant 2 seconds typical for wind gusts), and w_f, w_τ are white noise sources with covariance Q_w .

This model captures realistic wind behavior: not instantaneous white noise, but persistent gusts with timescale 1-5 seconds.

3.6. Measurement models with realistic sensor characteristics

The system has six sensor outputs:

$$y = [p_{\text{meas}}, R_{\text{meas}}]^T \in \mathbb{R}^3 \times \text{SO}(3). \quad (41)$$

Position measurement (GPS-like):

$$p_{\text{meas}} = p + n_p \quad (42)$$

where $n_p \sim \mathcal{N}(0, \sigma_p^2 I)$ with $\sigma_p = 0.08 \text{ m}$ (typical GPS accuracy).

Attitude measurement (gyroscope + accelerometer fusion):

Attitude is estimated from accelerometer (measuring gravitational acceleration) and gyroscope (measuring angular velocity). The measurement manifold is parameterized via the rotation matrix error:

$$R_{\text{meas}} = R \exp([n_R]_\times) \quad (43)$$

where $n_R \sim \mathcal{N}(0, \sigma_R^2 I)$ with $\sigma_R = 0.05 \text{ rad}$ (0.003 rad/s from gyroscope white noise, plus 0.02 rad from accelerometer bias).

Key point: attitude noise is multiplicative (right-perturbation on $\text{SO}(3)$), not additive. This distinction is critical for IEKF design to maintain manifold consistency.

3.7. Quadrotor kinematics modeling on Lie groups

The kinematics of a UAV can be elegantly described using the framework of Lie groups, which provides a powerful mathematical structure for representing rigid body motions in three-dimensional space. The configuration of the UAV is represented as an element of the Special Euclidean group $SE(3)$, which combines both rotation and translation:

$$T \in SE(3) = SO(3) \ltimes \mathbb{R}^3.$$

Here, T is a homogeneous transformation matrix given by:

$$T = \begin{bmatrix} R & p & 0_{1 \times 3} & 1 \end{bmatrix}$$

where $R \in SO(3)$ represents the orientation of the UAV and $p \in \mathbb{R}^3$ denotes the position vector of the UAV's center of mass in the inertial frame. The kinematics can be further decomposed into translational and rotational components:

3.7.1. Translational kinematics

The transformation of linear velocities $V = [u \ v \ w]^T$ in the body-fixed frame B to linear velocities $\dot{P} = [\dot{x} \ \dot{y} \ \dot{z}]^T$ in the inertial frame I is given by the rotation matrix R_p :

$$\dot{P} = R_p V \Leftrightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R_p \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \quad (44)$$

Expanding this equation yields:

$$\begin{cases} \dot{x} = C(\psi)C(\theta)u + \\ \quad [C(\psi)S(\theta)S(\phi) - S(\psi)C(\phi)]v + \\ \quad [C(\psi)S(\theta)C(\phi) + S(\psi)S(\phi)]w \\ \dot{y} = S(\psi)C(\theta)u + \\ \quad [S(\psi)S(\theta)S(\phi) + C(\psi)C(\phi)]v + \\ \quad [S(\psi)S(\theta)C(\phi) - C(\psi)S(\phi)]w \\ \dot{z} = -S(\theta)u + C(\theta)S(\phi)v + C(\theta)C(\phi)w \end{cases} \quad (45)$$

where $C(\cdot)$ and $S(\cdot)$ represent cosine and sine functions, respectively.

3.7.2. Rotational kinematics

The relationship between angular velocities (p, q, r) and Euler angle rates $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ is expressed as:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + Rot_x(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + Rot_x(\phi) Rot_y(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}. \quad (46)$$

The time derivative of the transformation matrix $\dot{T}$ encapsulates both the UAV's velocity and angular velocity:

$$\dot{T} = T \begin{bmatrix} \hat{\omega} & v & 0_{1 \times 3} & 0 \end{bmatrix}$$

where $\hat{\omega}$ is the skew-symmetric matrix form of the angular velocity vector $\omega \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$ is the linear velocity vector. The general kinematics of the quadrotor can be expressed in a compact form as:

$$\begin{bmatrix} \dot{P} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} R_p & 0_{3 \times 3} \\ 0_{3 \times 3} & Q_p \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix}. \quad (47)$$

This formulation on Lie groups provides a unified and elegant description of the UAV's kinematics, capturing both translational and rotational motion within the $SE(3)$ framework. This approach facilitates the development of sophisticated control and estimation algorithms that respect the underlying geometric structure of the system's configuration space.

3.8. Quadrotor dynamics modeling on Lie groups

The dynamic behavior of a quadrotor UAV can be elegantly described using the framework of Lie groups, particularly leveraging the algebraic structure of $SE(3)$ and $SO(3)$. This approach allows for a comprehensive representation of the coupled translational and rotational dynamics. The equations of motion, derived from Newton-Euler principles, can be expressed as follows:

3.8.1. Translational dynamics

Applying Newton's Second Law in the inertial frame I , we obtain:

$$m \frac{d}{dt} (\dot{P}) = \sum F_{ext}. \quad (48)$$

The primary external forces acting on the quadrotor are:

Gravitational force: Expressed in the inertial frame I as:

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix}. \quad (49)$$

Lift force: The total thrust produced by the four propellers, initially expressed in the body-fixed frame B and then transformed to the inertial frame using the rotation matrix R_p :

$$F_p^I = R_p F_p = \begin{bmatrix} C(\psi)S(\theta)C(\phi) + S(\psi)S(\phi) \\ S(\psi)S(\theta)C(\phi) - C(\psi)S(\phi) \\ C(\theta)C(\phi) \end{bmatrix} T \quad (50)$$

where $T = b(w_1^2 + w_2^2 + w_3^2 + w_4^2)$, with b being the thrust coefficient and w_i the angular velocity of the i -th propeller.

Drag force: Modeled as:

$$F_t = \frac{1}{2} \rho_{air} S C_d \dot{P}^2 \quad (51)$$

where ρ_{air} is the air density, S is the reference area, and C_d is the drag coefficient. Combining these forces, the translational dynamics can be expressed as:

$$\begin{cases} \ddot{x} = \frac{1}{m} [(C(\psi)S(\theta)C(\phi) + S(\psi)S(\phi)) T - K_{tx} \dot{x}^2] \\ \ddot{y} = \frac{1}{m} [(S(\psi)S(\theta)C(\phi) - C(\psi)S(\phi)) T - K_{ty} \dot{y}^2] \\ \ddot{z} = -g + \frac{1}{m} [C(\theta)C(\phi) T - K_{tz} \dot{z}^2] \end{cases} \quad (52)$$

where K_{tx} , K_{ty} , and K_{tz} are drag coefficients.

3.8.2. Rotational dynamics

The rotational dynamics, derived from Newton's Second Law for rotational motion, can be expressed on the Lie group $SO(3)$ as:

$$J\dot{\Omega} + \Omega \times J\Omega = \sum T_{ext}. \quad (53)$$

The external torques and moments include:

Translational moment: Resulting from differential thrust of the propellers:

$$\tau = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ d(\tau_1 - \tau_2 + \tau_3 - \tau_4) \end{bmatrix} \quad (54)$$

where l is the arm length and d is the drag-to-thrust ratio.

Gyroscopic moment: Arising from the interaction between propeller rotation and quadrotor movement:

$$\tau_{gyroscopique} = - \sum_{i=1}^4 J_r (\Omega^B \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}) (-1)^{i+1} w_i \quad (55)$$

where J_r is the rotor inertia.

Friction torque:

$$\tau_f = \begin{bmatrix} K_\phi \dot{\phi}^2 \\ K_\theta \dot{\theta}^2 \\ K_\psi \dot{\psi}^2 \end{bmatrix}. \quad (56)$$

By combining these moments, the rotational dynamics can be expressed as:

$$\begin{cases} \ddot{\phi} = \frac{1}{I_{xx}} [l(F_4 - F_2) - \dot{\psi} \dot{\theta} (I_{zz} - I_{yy}) \\ \quad - J_r \dot{\theta} (w_1 + w_3 - w_2 - w_4) - K_\phi \dot{\phi}^2] \\ \ddot{\theta} = \frac{1}{I_{yy}} [l(F_3 - F_1) - \dot{\psi} \dot{\phi} (I_{xx} - I_{zz}) \\ \quad + J_r \dot{\phi} (w_1 + w_3 - w_2 - w_4) - K_\theta \dot{\theta}^2] \\ \ddot{\psi} = \frac{1}{I_{zz}} [d(\tau_1 - \tau_2 + \tau_3 - \tau_4) - \dot{\theta} \dot{\phi} (I_{yy} - I_{xx}) \\ \quad - K_\psi \dot{\psi}^2]. \end{cases} \quad (57)$$

This comprehensive model, formulated on Lie groups, captures the essential dynamics of a quadrotor UAV, accounting for various forces and moments acting on the system. The use of Lie group theory provides a geometrically intuitive and computationally efficient framework for describing the coupled translational and rotational dynamics. This formulation serves as a solid foundation for developing advanced control strategies, state estimation techniques, and performance analysis methodologies for UAV systems evolving on Lie groups.

3.9. Representation using rotation matrices

Using rotation matrices to represent orientation provides a direct method for integrating rotational dynamics within the $SO(3)$ Lie group framework. From the kinematic equations (47), and the dynamic equations (52), (57) we can assemble the general dynamics of the quadrotor by the rotation matrix.

The translational equations:

$$\begin{cases} \dot{P} = R_p V \\ m\ddot{P} = R_p F_p + F_g + F_t \end{cases} \quad (58)$$

Rotation equations:

$$\begin{cases} \dot{R}_p = R_p [\Omega]_{\times} \\ \omega = Q_p \Omega \\ J\dot{\Omega} + \Omega \times J\Omega = \tau + \tau_{gyroscopique} + \tau_f. \end{cases} \quad (59)$$

4. Control design on Lie groups

4.1. Observer-controller design on Lie groups

The configuration space of the UAV naturally evolves on the Lie group $SE(3)$, representing the coupled position-attitude dynamics. We propose a geometric observer that exploits the intrinsic structure of this manifold to ensure physically consistent state estimation.

4.1.1. State estimation problem formulation

Consider the UAV dynamics on $SE(3)$:

$$\dot{T} = T\xi^{\wedge}, \quad T \in SE(3) \quad (60)$$

where $T = \begin{bmatrix} R & p \\ 0_{1 \times 3} & 1 \end{bmatrix}$ represents the homogeneous transformation matrix, with $R \in SO(3)$ being the rotation matrix and $p \in \mathbb{R}^3$ the position vector, and $\xi = \begin{bmatrix} v \\ \omega \end{bmatrix} \in \mathfrak{se}(3)$ is the velocity vector in the Lie algebra.

4.1.2. Lie group observer construction

We design an invariant observer that preserves the group structure:

$$\dot{\hat{T}} = \hat{T}(\hat{\xi}^\wedge + L_1 e_y^\wedge) \quad (61)$$

$$\dot{\hat{\xi}} = f(\hat{\xi}, u) + L_2 \text{Ad}_{\hat{T}^{-1}} e_y \quad (62)$$

where the innovation terms are defined as:

$$e_y = \log(\hat{T}^{-1} T_{meas})^\vee \quad (63)$$

$$T_{meas} = \exp(y^\wedge). \quad (64)$$

The observer gains are chosen as:

$$L_1 = \gamma_1 I_6, \quad L_2 = \gamma_2 I_6 \quad (65)$$

$$\gamma_1 > \frac{\sigma_{max}(V)}{\sigma_{min}(C^T C)} \quad (66)$$

$$\gamma_2 > \frac{\|L_1\|_F}{\lambda_{min}(\mathcal{G})}. \quad (67)$$

Key innovation: By updating $\hat{T}$ through left multiplication on the group, the estimate remains on the manifold, ensuring consistent geometric structure and avoiding singularities that plague conventional approaches.

4.2. Geometric control law integration

The estimated states from our Lie group observer are directly integrated into the control law, preserving geometric consistency:

$$u = K_p \log(\hat{T}_d^{-1} \hat{T})^\vee + K_d(\hat{\xi}_d - \hat{\xi}) \quad (68)$$

where $\hat{T}_d$ and $\hat{\xi}_d$ represent desired trajectory and velocity, respectively.

4.3. Stability analysis of coupled observer-controller system

THEOREM 4.1 *The closed-loop system comprising the Lie group observer (61)-(62) and geometric controller (68) is locally exponentially stable if the gains satisfy the conditions (66)-(67) and:*

$$K_p > \frac{\|L_1\|_2}{\sigma_{min}(\mathcal{L})}. \quad (69)$$

PROOF Consider the Lyapunov function on the tangent bundle:

$$V = \frac{1}{2}\|e_T\|_{\mathcal{M}}^2 + \frac{1}{2}\|e_\xi\|_{\mathfrak{se}(3)}^2 \quad (70)$$

where $\|\cdot\|_{\mathcal{M}}$ denotes the Riemannian metric on $SE(3)$.

The time derivative yields:

$$\dot{V} \leq -\alpha_1\|e_T\|^2 - \alpha_2\|e_\xi\|^2 + \beta\|w\|^2 \quad (71)$$

for positive constants $\alpha_1, \alpha_2, \beta$, ensuring exponential stability under the prescribed gain conditions. ■

To synthesize a control law, we need a simpler mathematical model (a synthesis model), which we developed in the previous section. In this section, we shall use the equations (58)-(59) which describe the kinematics and dynamics of the quadrotor through the rotation matrices, but we shall consider a few modifications:

- we assume that the linear velocities $V = [u \ v \ w]$, expressed in B , are equal to the linear velocities $\dot{P} = [\dot{x} \ \dot{y} \ \dot{z}]$ expressed in I ;
- we neglect the drag force in translational dynamics;
- we neglect the gyroscopic moment and the friction torque in the rotational dynamics to simplify the system.

In this work, the dynamic model of the quadrotor system with disturbances is expressed through:

$$\begin{cases} \dot{P} = V \\ \dot{V} = m^{-1}(RF + F_g + \Delta_V) \\ \dot{R} = R[\Omega]_{\times} \\ \dot{\Omega} = J^{-1}(M - \Omega \times J\Omega + \Delta_{\Omega}). \end{cases} \quad (72)$$

In the above:

- F, M are the control laws;
- $\Delta_V, \Delta_{\Omega}$ represent perturbations at the translational and rotational dynamics levels respectively.

4.4. PID control law synthesis

4.4.1. Error definitions

Let $P_d(t) \in \mathbb{R}^3$ represent the desired position trajectory. The tracking errors for position and velocity are expressed as:

$$e_P = P - P_d \quad (73)$$

$$e_V = \dot{e}_P = V - \dot{P}_d. \quad (74)$$

For the position tracking controller, we incorporate an integral term as defined in Maithripala and Berg (2015):

$$e_i = \int_0^\tau e_V(\tau) + c_1 e_P(\tau) d\tau \quad (75)$$

where c_1 is a positive constant. Consider a positive constant $\sigma \in R$, and a saturation function $sat_\sigma : R \rightarrow [-\sigma, \sigma]$, defined as:

$$sat_\sigma(y) = \begin{cases} \sigma; & si \ y > \sigma \\ y; & si \ y < \sigma \text{ et } y > -\sigma \\ -\sigma; & si \ y < -\sigma \end{cases} \quad (76)$$

For vector inputs $y \in R^n$, this saturation function is applied element-wise to define $sat_\sigma : R^n \rightarrow [-\sigma, \sigma]^n$. To track a computed attitude $R_c \in SO(3)$ and angular velocity Ω_c , we define:

$$R_c = R(\phi_c, \theta_c, \psi_d). \quad (77)$$

Here, ϕ_c and θ_c are the computed roll and pitch angles, while ψ_d is the desired yaw angle.

$$\Omega_c = (R_c^T \dot{R}_c)^\vee. \quad (78)$$

Next, we define attitude errors for the computed attitude and angular velocities.

PROPOSITION 1 (LEE, 2011) *Given a command (R_c, Ω_c) and current attitude and angular velocity (R, Ω) , we define an attitude error function $\Psi : SO(3) \times SO(3) \rightarrow R$, an attitude error vector $e_R \in R^3$, and an angular velocity error vector $e_\Omega \in R^3$ as in Maithripala and Berg (2015):*

$$\Psi(R, R_c) = \frac{1}{2} tr[I_{3 \times 3} - R_c^T R] \quad (79)$$

$$e_R = \frac{1}{2} (R_c^T R - R^T R_c)^\vee \quad (80)$$

$$e_\Omega = \Omega - R^T R_c \Omega_c. \quad (81)$$

PROPOSITION 2 (GOODARZI, LEE AND LEE, 2013) *Given a command (R_c, Ω_c) and current attitude and angular velocity (R, Ω) , we define an attitude error function $\Psi : SO(3) \times SO(3) \rightarrow R$, an attitude error vector $e_R \in R^3$, and an angular velocity error vector $e_\Omega \in R^3$ as in Maithripala and Berg (2015):*

$$\Psi(R, R_c) = \frac{1}{2} tr[I_{3 \times 3} - R_c^T R] \quad (82)$$

$$e_R = \frac{1}{2} (R_c^T R - R^T R_c)^\vee \quad (83)$$

$$e_\Omega = \Omega - R^T R_c \Omega_c. \quad (84)$$

4.4.2. Controller definition

We now present a nonlinear controller for position control, described by the expressions for moment vectors and lift force:

$$F = R_p^T(-k_P e_P - k_V e_V - k_i \text{sat}_\sigma(e_i) - F_g) \quad (85)$$

$$M = -k_R e_R - k_\Omega e_\Omega - k_I e_I + \Omega \times J\Omega \quad (86)$$

where k_P , k_V , k_i , k_R , k_Ω , and k_I are positive constants. The lift force and control moment comprise proportional, derivative, and integral terms, along with additional terms to counteract gravity and angular acceleration caused by actual angular velocities.

THEOREM 4.2 (CAMPOS ET AL., 2017) *Assume the initial conditions satisfy:*

$$\Psi(R(0), R_c(0)) < \psi_1 < 1 \quad (87)$$

$$\|e_P(0)\| < e_{P_{\max}} \quad (88)$$

where $\phi_1, e_{P_{\max}}$ are positive constants. Consider the inputs F , M defined in (85)-(86). For positive constants k_P, k_V , we choose positive constants $k_R, k_\Omega, k_i, k_I, c_1, c_2, \sigma, B_2$ such that:

$$k_i \sigma > \delta_P \quad (89)$$

$$c_1 < \min \left\{ \frac{4k_P k_V (1 - \alpha)^2}{k_V^2 (1 + \alpha)^2 + 4mk_P (1 - \alpha)}, \sqrt{\frac{k_P}{m}} \right\} \quad (90)$$

$$\|(2J - \text{tr}[J]I_{3 \times 3})\| \|\Omega_c\| \leq B_2 \quad (91)$$

$$c_2 < \min \left\{ \frac{\sqrt{k_R \lambda_m(J)}}{\lambda_M(J)}, \frac{4k_\Omega}{8k_R \lambda_M(J) + (k_\Omega + B_2)^2} \right\} \quad (92)$$

and equation (90) is satisfied, where $\alpha = \sqrt{\psi_1(2 - \psi_1)}$. Here, $\lambda_M(A)$ and $\lambda_m(A)$ denote the maximum and minimum eigenvalues of a square matrix A , respectively.

The moment expression allows the quadrotor's attitude to asymptotically track R_c , while the thrust magnitude expression enables asymptotic position tracking. Saturation of the integral term is necessary to limit the effects of attitude tracking errors on translational dynamics for overall system stability. The complete computational procedure for implementing the PID controller on SE(3) is summarized in Algorithm 1.

Algorithm 1 PID Control in $SE(3)$

```

1: Input:  $J, m, g, k_P, k_V, k_i, c_1, k_R, k_\Omega, k_I, c_2, \Delta V, \Delta \Omega$ 
2:    $P_d, \dot{P}_d, R_c, \Omega_c$ 
3: Output:  $P, V, R, \Omega$ 
4:  $dt \leftarrow 0.01$ 
5:  $N \leftarrow 1000$ 
6:  $i \leftarrow 1$ 
7:  $F_g \leftarrow [0; 0; mg]$ 
8:  $\Omega_0 \leftarrow [0; 0; 0]$ 
9:  $R_0 \leftarrow eye(3)$ 
10:  $V_0 \leftarrow [0; 0; 0]$ 
11:  $P_0 \leftarrow [0; 0; 0]$ 
12: while  $i < N$  do
13:    $e_P \leftarrow P - P_d$ 
14:    $e_V \leftarrow V - \dot{P}_d$ 
15:    $e_R \leftarrow vee(0.5(R_c^T R - R^T R_c))$ 
16:    $e_\Omega \leftarrow \Omega - R^T R_c \Omega_c$ 
17:    $e_I \leftarrow (e_\Omega + c_2 e_R)dt$ 
18:    $M \leftarrow -k_R * e_R - k_\Omega * e_\Omega - k_I * e_I + \Omega \times J\Omega;$ 
19:    $\Omega \leftarrow \Omega + J^{-1}(M - \Omega \times J\Omega + \Delta\Omega)dt$ 
20:    $R \leftarrow RExp(hat(\Omega)dt)$ 
21:    $e_i \leftarrow (e_V + c_1 e_P)dt$ 
22:    $F \leftarrow R^T(-k_P e_P - k_V e_V - k_i sat(e_i) - F_g)$ 
23:    $V \leftarrow V + m^{-1}(R * F - F_g + \Delta V)dt$ 
24:    $P \leftarrow P + Vdt$ 
25:    $i \leftarrow i + 1$ 
26: end while
27: return  $P, V, R, \Omega$ 

```

4.5. LQR control law synthesis

4.5.1. Error definitions

We will utilize Lie group theory to examine the dynamic framework of a quadrotor. Initially, it is essential to structure this system for optimal control objectives. In the context of optimal control modeling, the state vector x is adjusted by subtracting the desired reference state x_d , which ensures that the system remains at equilibrium (Kumar and Jerome, 2013):

$$\tilde{x} = x - x_d. \quad (93)$$

The state vector x includes the position coordinates (x, y, z) , linear velocities (u, v, w) , Euler angles (roll ϕ , pitch θ , yaw ψ), and their respective angular velocities (p, q, r) . Following the methodology outlined in Kumar and Jerome (2013), the state is represented as:

$$x = [P \quad V \quad \Phi \quad \Omega]^T. \quad (94)$$

Here, $\Phi \in SO(3)$ denotes the rotation matrix, replacing Euler angles to circumvent issues such as gimbal lock. Since Φ does not reside in the vector space $\mathbb{R}^3$, direct subtraction as in equation (93) is not feasible. Instead, Lie group operations must be employed to properly account for this structure. Consequently, the discrepancy between the desired and actual states is articulated as:

$$\tilde{x} = x \ominus x_d = \begin{bmatrix} P - P_d \\ V - \dot{P}_d \\ \log(\Phi \Phi_d^{-1})^\vee \\ \Omega - \Omega_d \end{bmatrix}. \quad (95)$$

According to Kumar and Jerome (2013), the time derivative of Φ is expressed as:

$$\dot{\Phi} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -p & q & 0 \end{bmatrix} \in \mathfrak{so}(3) \quad \Leftrightarrow \quad \dot{\Phi} = [\Omega]_\times \in \mathfrak{so}(3). \quad (96)$$

Subsequently, the $\vee$ operator is applied to convert the derivative from $so(3)$ to $\mathbb{R}^3$:

$$\dot{\Phi}^\vee = \Omega \in \mathbb{R}^3. \quad (97)$$

Leveraging equations (97) and (94), the dynamic model is derived as:

$$\dot{x} = \begin{bmatrix} \dot{P} \\ \dot{V} \\ \dot{\Phi} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} V \\ m^{-1}(RF + F_g + \Delta_V) \\ \Omega \\ J^{-1}(M - \Omega \times J\Omega + \Delta_R) \end{bmatrix}. \quad (98)$$

4.5.2. Control law development

The Linear Quadratic Regulator (LQR) method is suitable for linear systems characterized by:

$$\dot{x} = Ax + Bu. \quad (99)$$

For nonlinear systems described by:

$$\dot{x} = f(x, u). \quad (100)$$

it is necessary to linearize around an equilibrium point x_0 , resulting in:

$$A = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x_0}, \quad B = \left. \frac{\partial f(x, u)}{\partial u} \right|_{x_0} \quad (101)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times m}$. After performing the calculations, the linearized system takes the form:

$$\dot{x} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} x + \begin{bmatrix} 0_{3 \times 1} & 0_{3 \times 3} \\ I_m & 0_{3 \times 3} \\ 0_{3 \times 1} & 0_{3 \times 3} \\ 0_{3 \times 1} & J^{-1} \end{bmatrix} u \quad (102)$$

where $I_m = \begin{bmatrix} 0 \\ 0 \\ m^{-1} \end{bmatrix}$.

The implementation steps of the LQR controller formulated on SE(3) are presented in Algorithm 2.

4.6. Backstepping control design on SE(3)

4.6.1. Recursive control law synthesis

Backstepping is a systematic technique for nonlinear control design via recursive Lyapunov-based feedback. The method proceeds in two steps: translational (position/velocity) control, then rotational (attitude/angular velocity) control.

Algorithm 2 LQR Control in $SE(3)$

```

1: Input:  $m, J, g, \Delta_V, \Delta_\Omega, A, B, Q, R, P_d, \dot{P}_d, \phi_d, \theta_d, \psi_d, \Omega_d$ 
2: Output:  $P, V, \Phi, \Omega$ 
3:  $dt \leftarrow 0.01$ 
4:  $N \leftarrow 1000$ 
5:  $i \leftarrow 1$ 
6:  $F_g \leftarrow [0; 0; mg]$ 
7:  $E_d \leftarrow [\phi_d; \theta_d; \psi_d]$ 
8:  $k \leftarrow lqr(A, B, Q, R)$ 
9:  $\Omega_0 \leftarrow [0; 0; 0]$ 
10:  $\Phi_0 \leftarrow eye(3)$ 
11:  $V_0 \leftarrow [0; 0; 0]$ 
12:  $P_0 \leftarrow [0; 0; 0]$ 
13: while  $i \leq N$  do
14:    $\Phi_d \leftarrow Exp(hat(E_d))$ 
15:    $e_P \leftarrow P - P_d$ 
16:    $e_V \leftarrow V - \dot{P}_d$ 
17:    $e_R \leftarrow vee(log(\Phi \Phi_d^T))$ 
18:    $e_\Omega \leftarrow \Omega - \Omega_d$ 
19:    $e_x \leftarrow [e_P; e_V; e_R; e_\Omega]$ 
20:    $u \leftarrow -k \cdot e_x$ 
21:    $F \leftarrow u(1)$ 
22:    $M \leftarrow u(2 : 4)$ 
23:    $\Omega \leftarrow \Omega + J^{-1}(M - \Omega \times J\Omega + \Delta_\Omega)dt$ 
24:    $\Phi \leftarrow \Phi \cdot Exp(hat(\Omega)dt)$ 
25:    $V \leftarrow V + m^{-1}(R \cdot F - F_g + \Delta_V)dt$ 
26:    $P \leftarrow P + Vdt$ 
27:    $i \leftarrow i + 1$ 
28: end while
29: return  $P, V, \Phi, \Omega$ 

```

Step 1: Translational control design

Error definition:

$$e_p = p - p_d, \quad e_v = v - v_d. \quad (103)$$

Desired acceleration (virtual control):

$$a_d = a_d^{\text{ref}} - K_p e_p - K_v e_v \quad (104)$$

where a_d^{ref} is the reference acceleration from trajectory planning.

From Newton's second law (Eq. 29):

$$ma_d = mg + Rf_{\text{thrust}} + d_f. \quad (105)$$

Solving for thrust direction (desired z-axis of body frame) yields:

$$e_3^d = \frac{ma_d - mg - d_f}{\|ma_d - mg - d_f\|}. \quad (106)$$

Desired thrust magnitude:

$$f_d = \|ma_d - mg - d_f\| \quad (107)$$

with saturation to respect motor limits:

$$f_d^{\text{sat}} = \text{clip}(f_d, 4f_{\min}, 4f_{\max}). \quad (108)$$

Step 2: Rotational control design The translational error depends on attitude error (via R). Backstepping drives attitude to the desired value that generates desired thrust direction.

Attitude error (on $\text{SO}(3)$):

$$e_R = \frac{1}{2} \text{vee}(R_d^T R). \quad (109)$$

Angular velocity error:

$$e_\omega = \omega - \hat{\omega}_d \quad (110)$$

where $\hat{\omega}_d$ is the desired angular velocity ensuring attitude tracking.

Control torque (from backstepping recursion):

$$\tau_{\text{cmd}} = -K_R e_R - K_\omega e_\omega - \hat{\omega}_d \times J\hat{\omega}_d + J\dot{\hat{\omega}}_d - \hat{d}_\tau. \quad (111)$$

The last three terms are model-based compensation (nonlinear damping + feedforward).

4.6.2. Stability proof for backstepping on SE(3)

COROLLARY 1 (BACKSTEPPING STABILITY ON SE(3)) *With control law Equations (104)-(111) and gains $K_p, K_v, K_R, K_\omega > 0$, the closed-loop system satisfies:*

$$\lim_{t \rightarrow \infty} (e_p, e_v, e_R, e_\omega) \rightarrow 0 \text{ exponentially} \quad (112)$$

for bounded disturbances $\|d_f\| \leq d_f^{\max}, \|d_\tau\| \leq d_\tau^{\max}$.

PROOF This proof follows directly from Theorem 7.1 further on, applied to the backstepping control law. The composite Lyapunov function:

$$V_{BS} = V_{\text{trans}} + V_{\text{rot}} \quad (113)$$

satisfies $\dot{V}_{BS} < 0$ for sufficiently large gains, ensuring asymptotic stability. ■

The complete recursive implementation of the proposed backstepping controller is summarized in Algorithm 3.

5. Lie group-based observer design

The primary objective of this section is to present a rigorous geometric observer framework for robust real-time state estimation of a 3DOF hover UAV evolving on the Lie group $SE(3)$. We address not only the design of this observer, but also the mathematical integration of disturbance estimation and convergence guarantees, highlighting the intrinsic benefits of operating directly on the manifold as opposed to conventional Euclidean schemes. The methodology enables precise and resilient observer-based control, tailored for nonlinear, uncertain quadrotor dynamics, addressing key limitations in the literature on geometric state estimation, as surveyed in recent works, Mahony, Kumar and Corke (2008), Barrau and Bonnabel (2017), Zhang et al. (2024).

5.1. Geometric observer synthesis

The observer must estimate both T and ξ , while intrinsically preserving the geometric constraints of $SE(3)$. The proposed invariant observer follows:

$$\dot{\hat{T}} = \hat{T} \left(\hat{\xi}^\wedge + K_I e^\wedge \right), \quad (114)$$

$$\dot{\hat{\xi}} = f(\hat{\xi}, u) + K_\xi e_\xi, \quad (115)$$

where $e = \log \left(\hat{T}^{-1} T_{\text{meas}} \right)^\vee$ is the innovation computed in the Lie algebra, ensuring that corrections remain within the tangent space and are consistent with

Algorithm 3 BACKSTEPPING Control in $SE(3)$

```

1: Input:  $J, m, g, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \Delta V, \Delta \Omega, P_d, \dot{P}_d, R_c, \Omega_c$ 
2: Output:  $P, V, R, \Omega$ 
3:  $dt \leftarrow 0.01$ 
4:  $N \leftarrow 1000$ 
5:  $i \leftarrow 1$ 
6:  $F_g \leftarrow [0; 0; mg]$ 
7:  $\Omega_0 \leftarrow [0; 0; 0]$ 
8:  $R_0 \leftarrow eye(3)$ 
9:  $V_0 \leftarrow [0; 0; 0]$ 
10:  $P_0 \leftarrow [0; 0; 0]$ 
11: while  $i < N$  do
12:    $e_1 \leftarrow P - P_d$ 
13:    $\varphi_1 \leftarrow \dot{P}_d - \lambda_1 e_1$ 
14:    $e_2 \leftarrow V - \varphi_1$ 
15:    $\dot{\varphi}_1 \leftarrow \dot{P}_d - \lambda_1 (V - \dot{P}_d)$ 
16:    $\tilde{R} \leftarrow R_c^T R$ 
17:    $e_3 \leftarrow 0.5 \cdot vee(\tilde{R} - \tilde{R}^T)$ 
18:    $\mathcal{D} \leftarrow tr(\tilde{R}^T) \cdot I - \tilde{R}^T$ 
19:    $e_\Omega \leftarrow \Omega - \tilde{R}^T \Omega_c$ 
20:    $\varphi_2 \leftarrow -2 \cdot \lambda_3 \cdot \mathcal{D} \cdot e_3$ 
21:    $e_4 \leftarrow e_\Omega - \varphi_2$ 
22:    $\dot{\varphi}_2 \leftarrow -2 \cdot \lambda_3 \cdot ([e_\Omega]_\times \tilde{R}^T + \mathcal{D} \cdot (0.5 \cdot \mathcal{D} \cdot e_\Omega))$ 
23:    $M \leftarrow \Omega \times J\Omega + J \cdot (-0.5 \cdot \mathcal{D} \cdot e_1 - [e_\Omega]_\times \cdot \tilde{R}^T \cdot \Omega_c + \tilde{R}^T \dot{\Omega}_c + \dot{\varphi}_2 - \lambda_4 e_4)$ 
24:    $\Omega \leftarrow \Omega + J^{-1} \cdot (M - \Omega \times J\Omega + \Delta \Omega) \cdot dt$ 
25:    $R \leftarrow R \cdot Exp(hat(\Omega) \cdot dt)$ 
26:    $F \leftarrow R^T \cdot (F_g + m \cdot (\dot{\varphi}_1 - e_1 - \lambda_2 e_2))$ 
27:    $V \leftarrow V + m^{-1} \cdot (R \cdot F - F_g + \Delta V) \cdot dt$ 
28:    $P \leftarrow P + V \cdot dt$ 
29:    $i \leftarrow i + 1$ 
30: end while
31: return  $P, V, R, \Omega$ 

```

the rigid-body transformation group structure. Here K_I and K_ξ are positive definite observer gains.

This observer design is fundamentally geometric, aligning with recent advances, which demonstrate global stability and drift-free performance even under large orientation errors or persistent noise (Mahony, Kumar and Corke, 2008; Barrau and Bonnabel, 2017).

5.2. Disturbance observer integration

To further augment robustness, we integrate a disturbance observer, acting on both translation and rotation:

$$\dot{\hat{d}} = -\alpha \hat{d} + \alpha L_3 e, \quad (116)$$

$$\hat{\xi}_c = \hat{\xi} + \hat{d}, \quad (117)$$

where $\hat{d}$ is the estimated disturbance, L_3 is the disturbance observer gain, and $\alpha > 0$ defines the adaptation rate. This feedback mechanism allows both state and disturbance estimates to be leveraged by the controller for corrective action.

5.3. Discrete-time invariant extended Kalman filter (IEKF)

For practical real-time applications, a discretized observer, compatible with the platform's sample rate, is necessary. The IEKF in $SE(3)$ operates as follows for each step k :

- **Prediction:**

$$\hat{T}_{k+1|k} = \hat{T}_{k|k} \exp(\hat{\xi}_k^\wedge \Delta t), \quad (118)$$

$$\hat{\xi}_{k+1|k} = f(\hat{\xi}_k, u_k) \Delta t + \hat{\xi}_k. \quad (119)$$

- **Innovation:**

$$e_k = \log(\hat{T}_{k+1|k}^{-1} T_{y,k+1})^\vee. \quad (120)$$

- **Update:**

$$\hat{T}_{k+1} = \hat{T}_{k+1|k} \exp(K_I e_k), \quad (121)$$

$$\hat{\xi}_{k+1} = \hat{\xi}_{k+1|k} + K_\xi e_k. \quad (122)$$

This ensures intrinsic consistency with the geometry of $SE(3)$ at every iteration, and is provably superior to Euclidean-based EKF approaches under aggressive maneuvers and perturbations (Barrau and Bonnabel, 2017; Liu, Maithripla and Jayasuriya, 2025; Maithripla, Berg and Dayawansa, 2014).

5.4. Random variables and densities on Lie groups

The linear vector space structure of a Lie group allows for the direct application of conventional probability and statistical methods without reconstructing the manifold itself.

As a specific illustration on the group $SO(3)$, an element C can be parameterized through exponential coordinates as follows:

$$C = \exp(\widehat{e}) \bar{C}, \quad (123)$$

where $e \in \mathbb{R}^3$ is a conventional random vector, $\widehat{(\cdot)}$ denotes the hat operator corresponding to the Lie algebra, and $\bar{C} = \exp(\widehat{\xi})$ is a fixed reference element on the group. This construction implies that any probability distribution defined on $\mathbb{R}^3$ induces a natural corresponding distribution on $SO(3)$:

$$p(e) \longrightarrow p(C).$$

The focus here is on the Gaussian distribution due to its fundamental role in Kalman filtering. For the multivariate Gaussian distribution with $e \in \mathbb{R}^3$, the probability density function (PDF) is given by

$$p(e) = \frac{1}{\sqrt{(2\pi)^3 \det(\Sigma)}} \exp\left(-\frac{1}{2}e^\top \Sigma^{-1}e\right), \quad (124)$$

which satisfies the normalization condition

$$\int_{\mathbb{R}^3} p(e) de = 1.$$

As reviewed in Section 2.7.2, performing a change of variables from the vector space $\mathbb{R}^3$ (the Lie algebra) to the Lie group $SO(3)$ via the exponential map relates the differential elements as

$$dC = |\det(J(e))| de,$$

where $J(e)$ is the Jacobian matrix, associated with the chosen parameterization. Consequently, the integral transforms as

$$\begin{aligned} 1 &= \int_{\mathbb{R}^3} p(e) de \\ &= \int_{\mathbb{R}^3} \frac{1}{\sqrt{(2\pi)^3 \det(\Sigma)}} \exp\left(-\frac{1}{2}e^\top \Sigma^{-1}e\right) de, \end{aligned}$$

where $\text{Log}(\cdot)$ denotes the matrix logarithm on $SO(3)$, and the operator $(\cdot)^\vee$ maps elements from the Lie algebra back to the Euclidean vector space $\mathbb{R}^3$.

This construction demonstrates how Gaussian distributions on vector space naturally extend to probability density functions on nonlinear manifolds, maintaining geometric structure and essential properties for state estimation and filtering.

5.5. Stability and convergence analysis

THEOREM 5.1 (*Observer Convergence*): *The geometric observer with properly chosen gains L_1 and L_2 ensures exponential convergence of the estimation error.*

PROOF Consider the Lyapunov function on the tangent bundle:

$$V = \frac{1}{2} \|e_T\|_{\mathcal{M}}^2 + \frac{1}{2} \|e_\xi\|^2 \quad (125)$$

where $\|\cdot\|_{\mathcal{M}}$ denotes the Riemannian metric on $SE(3)$. The derivative analysis shows that with gains satisfying:

$$L_1 > \frac{\sigma_{\max}(\Sigma_w)}{\sigma_{\min}(C^T C)} \quad (126)$$

$$L_2 > \frac{\|L_1\|_F}{\lambda_{\min}(\mathcal{G})} \quad (127)$$

the system exhibits exponential convergence with rate $\lambda = \min(\sigma_{\min}(L_1), \sigma_{\min}(L_2))$. ■

Define $e_T := \hat{T}^{-1}T$ and $e_\xi := \hat{\xi} - \xi$. Consider the Lyapunov function

$$V = \frac{1}{2} \|\log(e_T)\|^2 + \frac{1}{2} \|e_\xi\|^2.$$

With an appropriate choice of positive definite gains K_I, K_ξ , it follows that

$$\dot{V} \leq -\lambda_{\min}(K_I) \|\log(e_T)\|^2 - \lambda_{\min}(K_\xi) \|e_\xi\|^2 + \text{bounded disturbances},$$

yielding exponential convergence of the observer errors, as rigorously established in Mahony, Kumar and Corke (2008) and Barrau and Bonnabel (2017).

Conventional Euclidean state observers require projection operations in order to maintain the orthogonality of R and the physical feasibility of T . These projections often lead to estimation drift, numerical instability, and deteriorated robustness, especially for large initial errors or persistent noise. By contrast, the geometric observer ensures $R \in SO(3)$, $T \in SE(3)$ by construction, and maintains error evolution within the appropriate manifold. Quantitative performance improvements, such as approximately 48–50%, that lower estimation

error and zero drift, are demonstrated experimentally and in simulation (Barrau and Bonnabel, 2017; Zhang et al., 2024).

The EKF-DO observer estimates the full system state $\hat{x}$ including position, attitude, velocity, and disturbances. This estimate feeds the control law, which uses $\hat{x}$ to compute control commands. While the observer and the controller are designed separately using Lyapunov theory, their combined operation achieves the hierarchical closed-loop stability. We now design three control strategies (PID, LQR, Backstepping) that operate on the estimated state $\hat{x}$.

6. Disturbance observer design for quadrotor control

The development of robust state estimation for UAVs operating on Lie groups requires careful consideration of the geometric structure underlying the system dynamics. This section presents our novel observer design methodology that exploits the intrinsic properties of the $SE(3)$ manifold.

For the purpose of enhancing the robustness of quadrotor control in the presence of uncertainties and external disturbances, we implement disturbance observers for both translational and rotational dynamics. These observers estimate the disturbances $\Delta \mathbf{V}$ and $\Delta \mathbf{\Omega}$, which are then incorporated into the control laws to improve trajectory tracking performance.

6.1. Tracking error definitions

We define tracking errors for both translational and rotational subsystems:

6.1.1. Translational tracking errors

$$\begin{aligned} e_P &= P - P_d(t) \\ e_V &= V - \dot{P}_d(t). \end{aligned} \tag{128}$$

6.1.2. Rotational tracking errors

$$\begin{aligned} e_R &= \text{vee} \left(\frac{1}{2} (R_d(t)^\top R - R^\top R_d(t)) \right) \\ e_\Omega &= \Omega - R^\top R_d(t) \Omega_d(t) \end{aligned} \tag{129}$$

where $\text{vee}(\cdot)$ is the inverse of the hat map, converting a skew-symmetric matrix to a vector in $\mathbb{R}^3$.

6.2. Disturbance observer design

6.2.1. Translational disturbance observer

The translational disturbance ΔV is estimated as:

$$\Delta V \leftarrow \Delta V + (-k_v \Delta V + k_v m^{-1} (R f_e e_z - V) + e_V) \cdot dt. \quad (130)$$

6.2.2. Rotational disturbance observer

The rotational disturbance $\Delta \Omega$ is estimated as:

$$\Delta \Omega \leftarrow \Delta \Omega + (-k_\omega \Delta \Omega + k_\omega J^{-1} (M - \Omega \times J \Omega) + \alpha e_R + e_\Omega) \cdot dt \quad (131)$$

where:

- k_v, k_ω : positive constants determining the speed and accuracy of disturbance estimation
- α : positive constant enhancing the observer's responsiveness to attitude errors
- J : inertia matrix of the quadrotor
- M : control moment.

6.3. Control law integration

The estimated disturbances are incorporated into the control laws:

6.3.1. Virtual control input

$$F = m(\ddot{P}_d(t) + g e_z) - K_2 e_V - \varphi e_P - \Delta V. \quad (132)$$

6.3.2. Thrust

$$f = F^\top R e_z. \quad (133)$$

6.3.3. Torque

$$\begin{aligned} M = & -K_R e_R - K_\Omega e_\Omega - \Delta \Omega + [\Omega]_\times (\Omega) + [J]_\times (\Omega) \\ & - J([\Omega]_\times R^\top R_d(t) [d(\Omega)]_d(t)). \end{aligned} \quad (134)$$

The gain φ ensures that tracking errors remain within safety bounds:

$$\varphi_i = \beta \frac{\xi_i^2}{(\xi_i^2 - e_{P_i}^2)^2} \quad \text{for } i = 1, 2, 3 \quad (135)$$

$$\varphi = \text{diag}(\varphi_1, \varphi_2, \varphi_3) \quad (136)$$

where β is a positive constant, and ξ_i defines the maximum allowable tracking error in each direction.

6.4. System state updates

The control inputs and estimated disturbances are used to update the quadrotor's state:

6.4.1. Angular velocity update

$$\Omega \leftarrow \Omega + J^{-1}(M - \Omega \times J\Omega + \Delta\Omega) \cdot dt. \quad (137)$$

6.4.2. Rotation matrix update

$$R \leftarrow R \cdot \text{RExp}([\Omega]_{\times} \cdot dt). \quad (138)$$

This ensures that R remains within the $\text{SO}(3)$ manifold by applying the matrix exponential map.

6.4.3. Velocity update

$$V \leftarrow V + m^{-1}(Rf_e e_z + F_g + \Delta V) \cdot dt. \quad (139)$$

6.4.4. Position update

$$P \leftarrow P + V \cdot dt. \quad (140)$$

6.5. Stability analysis

The stability of the closed-loop system is verified through Lyapunov-based analysis, demonstrating that tracking errors are uniformly ultimately bounded despite disturbances. Proper selection of observer and control gains $k_v, k_\omega, K_1, K_2, K_R, K_\Omega$ ensures robust performance, maintaining tracking errors within specified safety bounds ξ_1, ξ_2, ξ_3 .

6.6. Ensuring rotation matrix validity

The rotation matrix $\mathbf{R}$ is maintained within the $\text{SO}(3)$ manifold using the matrix exponential map:

$$\mathbf{R} \leftarrow \mathbf{R} \cdot \text{RExp}([\Omega]_{\times} \cdot dt). \quad (141)$$

This operation preserves the orthogonality and determinant conditions of $\text{SO}(3)$ throughout the simulation.

7. State observer design

The Extended Kalman Filter (EKF) is applied to a quadrotor functioning within the Special Orthogonal group $\text{SO}(3)$ to estimate its state. The state vector x includes the position $P \in \mathbb{R}^3$, velocity $V \in \mathbb{R}^3$, rotation matrix $R \in \text{SO}(3)$, and angular velocity $\Omega \in \mathbb{R}^3$:

$$x = [P^{\top}, V^{\top}, \text{vec}(R)^{\top}, \Omega^{\top}]^{\top}. \quad (142)$$

Here, $\text{vec}(R)$ signifies the vectorization of the rotation matrix.

7.1. Discrete-time implementation

The EKF operates in discrete time with a time interval Δt . The prediction and update stages are outlined below:

7.1.1. Prediction step

Predict the state using system dynamics, established in (58)-(59)

1. State prediction:

$$P_{k+1|k} = P_k + V_k \Delta t \quad (143)$$

$$V_{k+1|k} = V_k + m^{-1}(R_k F_k + F_g + \Delta V_k) \Delta t \quad (144)$$

$$R_{k+1|k} = R_k \exp([\Omega_k]_{\times} \Delta t) \quad (145)$$

$$\Omega_{k+1|k} = \Omega_k + J^{-1}(M_k - \Omega_k \times J \Omega_k + \Delta \Omega_k) \Delta t \quad (146)$$

where $\exp(\cdot)$ denotes the matrix exponential function.

2. Covariance prediction:

$$P_{k+1|k} = F_k P_k F_k^{\top} + Q_k. \quad (147)$$

Here, F_k is the Jacobian of the system model and Q_k represents the process noise covariance matrix.

7.1.2. Update step

1. Kalman gain computation:

$$K_k = P_{k|k-1} H_k^\top (H_k P_{k|k-1} H_k^\top + R_k)^{-1} \quad (148)$$

where H_k is the Jacobian of the measurement model and R_k is the measurement noise covariance matrix.

2. State update:

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - h(\hat{x}_{k|k-1})) \quad (149)$$

where z_k denotes the measurement and $h(\cdot)$ is the measurement model function.

3. Covariance update:

$$P_k = (I - K_k H_k) P_{k|k-1}. \quad (150)$$

The complete prediction and correction procedure of the Extended Kalman Filter on SE(3) is provided in Algorithm 4.

7.2. Detailed stability analysis via Lyapunov theory

7.2.1. Theorem: exponential stability of observer-controller system

THEOREM 7.1 (COUPLED OBSERVER-CONTROLLER STABILITY) *Consider the quadrotor system described by Equations (29)-(31) with:*

- *Observer designed according to Section 3 (IEKF-DO).*
- *Controller designed according to Section 4 (Backstepping on SE(3)).*
- *Actuator dynamics modeled by Equation. (37)*
- *Disturbances bounded: $\|d_f\| \leq d_f^{\max}$, $\|d_\tau\| \leq d_\tau^{\max}$.*

Then, there exist control gains K_p , K_v , K_R , K_ω and observer gains L_1 , L_2 , such that:

$$\|e_x\|_2 \leq \beta e^{-\lambda t} + \mu(d_f^{\max} + d_\tau^{\max}) \quad (151)$$

where e_x is the combined state estimation error, $\lambda > 0$ is the decay rate, β is determined by initial conditions, and μ bounds the ultimate error due to disturbances.

PROOF We construct a Lyapunov function decomposed into translational and rotational error components:

$$V_{\text{total}} = V_{\text{trans}} + V_{\text{rot}} + V_{\text{obs}}. \quad (152)$$

Algorithm 4 Extended Kalman Filter in SE(3) for Quadrotor

```

1: Input:  $J, m, g, Q, R, \Delta V, \Delta \Omega, P_0, V_0, R_0, \Omega_0$ 
2: Output: Estimated states  $P, V, R, \Omega$  and covariance  $P_k$ 
3:  $dt \leftarrow 0.01, N \leftarrow 1000, k \leftarrow 1$ 
4: Initialize state:
5:    $P \leftarrow P_0, V \leftarrow V_0, R \leftarrow R_0, \Omega \leftarrow \Omega_0$ 
6: Initialize covariance:
7:    $P_k \leftarrow$  Initial Covariance Matrix
8: while  $k \leq N$  do
9:   Prediction Step:
10:  Predict the state using system dynamics:
11:     $\dot{P} = V$ 
12:     $\dot{V} = m^{-1}(RF + F_g + \Delta V)$ 
13:     $\dot{R} = R[\Omega]_{\times}$ 
14:     $\dot{\Omega} = J^{-1}(M - \Omega \times J\Omega + \Delta\Omega)$ 
15:  Update the state estimates:
16:     $P \leftarrow P + \dot{P} \cdot dt$ 
17:     $V \leftarrow V + \dot{V} \cdot dt$ 
18:     $\Omega \leftarrow \Omega + \dot{\Omega} \cdot dt$ 
19:     $R \leftarrow R \cdot \text{Exp}([\Omega]_{\times} \cdot dt)$ 
20:  Compute the Jacobian  $F_k = \left. \frac{\partial f}{\partial x} \right|_{x_k, u_k}$ 
21:  Predict the covariance:
22:     $P_k \leftarrow F_k P_k F_k^T + Q$ 
23:  Update Step:
24:  Obtain measurements  $z_k$ 
25:  Define the measurement model:
26:     $z_k = h(x_k) + v$ 
27:  Compute the Jacobian  $H_k = \left. \frac{\partial h}{\partial x} \right|_{x_k}$ 
28:  Compute the Kalman Gain:
29:     $K_k = P_k H_k^T (H_k P_k H_k^T + R)^{-1}$ 
30:  Update the state estimates:
31:     $\hat{x}_k = \hat{x}_k + K_k(z_k - h(\hat{x}_k))$ 
32:  Update the covariance estimates:
33:     $P_k \leftarrow (I - K_k H_k) P_k$ 
34:  Increment:
35:     $k \leftarrow k + 1$ 
36: end while
37: return  $P, V, R, \Omega, P_k$ 

```

Part 1: Translational error stability

Define translational errors:

$$e_p = p - p_d, \quad e_v = v - v_d \quad (153)$$

where subscript d denotes desired (reference) trajectory.

Consider Lyapunov candidate:

$$V_{\text{trans}} = \frac{1}{2}e_p^T e_p + \frac{1}{2}e_v^T e_v. \quad (154)$$

Taking the derivative:

$$\dot{V}_{\text{trans}} = e_p^T \dot{e}_p + e_v^T \dot{e}_v. \quad (155)$$

The velocity error dynamics are (from Eq. 29):

$$\dot{e}_v = R(\hat{R}^T \dot{v}_d - f_{\text{cmd}}/m) - \dot{v}_d + d_f/m. \quad (156)$$

Apply control law (from backstepping, Section 4):

$$f_{\text{cmd}} = m(\dot{v}_d - K_v e_v - K_p e_p). \quad (157)$$

Substituting:

$$\dot{e}_v = -K_v e_v - K_p e_p + d_f/m. \quad (158)$$

Therefore:

$$\dot{V}_{\text{trans}} = -e_p^T K_p e_p - e_v^T K_v e_v + e_v^T (d_f/m). \quad (159)$$

For $K_p, K_v > 0$ (positive definite), choose them such that:

$$\dot{V}_{\text{trans}} \leq -\lambda_{\text{trans}} V_{\text{trans}} + \rho_{\text{trans}} \|d_f\|^2 \quad (160)$$

with $\lambda_{\text{trans}} > 0$. This ensures exponential convergence with disturbance-dependent ultimate bound.

Part 2: Rotational error stability

Define rotational error on $\text{SO}(3)$:

$$e_R = \frac{1}{2} \text{vee}(\hat{R}^T R). \quad (161)$$

Define angular velocity error:

$$e_\omega = \omega - \hat{R}^T R \omega_d. \quad (162)$$

Lyapunov candidate for rotational dynamics:

$$V_{\text{rot}} = e_R^T e_R + e_\omega^T e_\omega. \quad (163)$$

From rotational dynamics (Eq. 30):

$$\dot{e}_\omega = J^{-1}(-\omega \times J\omega + \tau_{\text{cmd}} + d_\tau) - R_d^T R \dot{\omega}_d. \quad (164)$$

Apply backstepping control:

$$\tau_{\text{cmd}} = -K_R e_R - K_\omega e_\omega + (\text{gyroscopic compensation terms}). \quad (165)$$

Standard SO(3) Lyapunov analysis yields:

$$\dot{V}_{\text{rot}} \leq -\lambda_{\text{rot}} V_{\text{rot}} + \rho_{\text{rot}} \|d_\tau\|^2. \quad (166)$$

Part 3: Observer error stability

Define observer estimation error:

$$\tilde{x} = x - \hat{x}. \quad (167)$$

For IEKF on Lie groups, the error dynamics are state-estimate-independent (fundamental IEKF property):

$$\dot{\tilde{x}} = A_{\text{obs}} \tilde{x} + B_{\text{obs}} w \quad (168)$$

where A_{obs} is independent of $\hat{x}$ (key IEKF advantage).

With observer gains chosen such that $A_{\text{obs}} + LH$ is Hurwitz:

$$V_{\text{obs}} = \tilde{x}^T P^{-1} \tilde{x} \quad (169)$$

satisfies:

$$\dot{V}_{\text{obs}} \leq -2\lambda_{\text{obs}} V_{\text{obs}} + \text{noise terms}. \quad (170)$$

Part 4: Composite Lyapunov function

Combine all three components:

$$V_{\text{total}} = c_1 V_{\text{trans}} + c_2 V_{\text{rot}} + c_3 V_{\text{obs}} \quad (171)$$

with weights $c_1, c_2, c_3 > 0$ chosen to ensure:

$$\dot{V}_{\text{total}} \leq -\min(c_1 \lambda_{\text{trans}}, c_2 \lambda_{\text{rot}}, c_3 \lambda_{\text{obs}}) \cdot V_{\text{total}} + \rho(d_f^{\text{max}}, d_\tau^{\text{max}}). \quad (172)$$

Define $\lambda = \min(c_1 \lambda_{\text{trans}}, c_2 \lambda_{\text{rot}}, c_3 \lambda_{\text{obs}}) > 0$.

Exponential Convergence: By standard Lyapunov theory for time-varying systems with bounded disturbances:

$$V_{\text{total}}(t) \leq V_{\text{total}}(0)e^{-\lambda t} + \int_0^t e^{-\lambda(t-\tau)} \rho(\tau) d\tau. \quad (173)$$

Assuming disturbances are bounded, $\rho(\tau) \leq \rho_{\text{max}}$:

$$V_{\text{total}}(t) \leq V_{\text{total}}(0)e^{-\lambda t} + \frac{\rho_{\text{max}}}{\lambda}(1 - e^{-\lambda t}). \quad (174)$$

Converting from Lyapunov function to state error norm via $\alpha_1(V) \leq \|e_x\|^2 \leq \alpha_2(V)$:

$$\|e_x(t)\| \leq \sqrt{\frac{\alpha_2(V_{\text{total}}(0))}{\alpha_1}} e^{-\lambda t/2} + \sqrt{\frac{\rho_{\text{max}}}{\lambda \alpha_1}}. \quad (175)$$

Identifying $\beta = \sqrt{\alpha_2(V_{\text{total}}(0))/\alpha_1}$ and $\mu = \sqrt{\rho_{\text{max}}/(\lambda \alpha_1)}$ completes the proof. ■

7.2.2. Convergence rate and gain design

The convergence rate λ is determined by:

$$\lambda = \min(\lambda_{\text{trans}}, \lambda_{\text{rot}}, \lambda_{\text{obs}}). \quad (176)$$

For the translational subsystem:

$$\lambda_{\text{trans}} = \min(\text{eig}(K_p), \text{eig}(K_v)). \quad (177)$$

For the rotational subsystem :

$$\lambda_{\text{rot}} = \min(\text{eig}(K_R), \text{eig}(K_\omega)). \quad (178)$$

For the observer, the stability margin depends on observer gains and noise covariances.

Design trade-off: Increasing gains (larger λ) speeds convergence but increases sensitivity to noise and actuator saturation. The MPSO optimization (Section 6) automatically balances this trade-off across multiple objectives.

7.3. Detailed extension to 6-DOF quadrotor control

7.3.1. Mathematical framework for 6-DOF extension

The laboratory 3-DOF hover system has constrained translational dynamics (z-axis fixed, horizontal motion achieved via attitude commands) and unconstrained yaw rotation. Full 6-DOF UAVs have all translational and rotational degrees of freedom.

State vector comparison is provided in Table 2.

Table 2: 3-DOF vs. 6-DOF state space comparison

Component	3-DOF Lab	6-DOF Quad	Dimension
Position	(x, y) constrained	(x, y, z) free	3
Velocity	(v_x, v_y) constrained	(v_x, v_y, v_z) free	3
Attitude	Yaw ψ only	Full $R \in \text{SO}(3)$	3
Angular velocity	$\dot{\psi}$ only	$\boldsymbol{\omega} \in \mathbb{R}^3$	3
Disturbances	(d_f^x, d_f^y, d_f^z)	$(d_f^x, d_f^y, d_f^z, d_\tau^x, d_\tau^y, d_\tau^z)$	6
Total	8 states	12 states	+4

Key insight: The mathematical framework developed for 6-DOF SE(3) (Equations 29-40) naturally encompasses the 3-DOF case as a special case with constrained z-dynamics.

7.3.2. Concrete extension procedure

Step 1: Observer extension

The IEKF observer predictor-corrector structure remains identical; only state dimensions change.

3-DOF observer state:

$$x_{3DOF}^{\text{obs}} = [x, y, v_x, v_y, \psi, \dot{\psi}, d_f^x, d_f^y, d_\tau^z]^T \in \mathbb{R}^9. \quad (179)$$

6-DOF observer state (natural extension):

$$x_{6DOF}^{\text{obs}} = [x, y, z, v_x, v_y, v_z, R, \boldsymbol{\omega}, d_f, d_\tau]^T \in \text{SE}(3) \times \mathbb{R}^3 \times \mathbb{R}^3. \quad (180)$$

The predictor step generalizes automatically:

$$\hat{p}_{k+1|k}^{6DOF} = \hat{p}_k + T\hat{v}_k \quad (\text{same for both 3-DOF and 6-DOF}). \quad (181)$$

Velocity prediction for 6-DOF includes z-component:

$$\hat{v}_{k+1|k}^{6DOF} = \hat{v}_k + T \left(mg + \hat{R}_k [\hat{f}_1, \hat{f}_2, \hat{f}_3]^T / m + \hat{d}_{f,k} / m \right) \quad (182)$$

where $\hat{f}_1, \hat{f}_2$ are horizontal thrust components (absent in 3-DOF where they are controlled via attitude).

The attitude prediction remains identical:

$$\hat{R}_{k+1|k} = \hat{R}_k \exp(T[\hat{\omega}_k]_{\times}) \quad (\text{same for both}). \quad (183)$$

Measurement update: In 6-DOF, barometric altitude measurement becomes available:

$$y_{6DOF} = [p_{GPS}, R_{IMU}, z_{\text{baro}}]^T. \quad (184)$$

Measurement Jacobian extends:

$$H_{6DOF} = \begin{bmatrix} I_3 & 0_{3 \times 9} \\ 0_{3 \times 6} & I_3 & 0_{3 \times 6} \\ [0, 0, 1] & 0_{1 \times 11} \end{bmatrix}_{7 \times 15}. \quad (185)$$

Complexity increase: observer goes from 9D (3-DOF) to 15D (6-DOF); Jacobians from 6×9 to 7×15 . Computational cost scales as $O(n^3)$ for matrix operations, predicting 3-4-fold increase. However, actual increase is measured at 2.5-fold increase due to efficient block-diagonal structure.

Step 2: Controller extension

3-DOF control law: Generates commanded yaw angle ψ_d and vertical thrust f_d .

6-DOF control law (backstepping):

Step 1 (translational) produces desired acceleration a_d with all three components:

$$a_d^{6DOF} = [\ddot{x}_d, \ddot{y}_d, \ddot{z}_d]^T. \quad (186)$$

This maps to desired attitude R_d and thrust magnitude f_d via Equation (106):

$$e_3^d = \frac{ma_d - mg - \hat{d}_f}{\|ma_d - mg - \hat{d}_f\|}. \quad (187)$$

Now e_3^d is fully 3D (not just vertical), allowing arbitrary attitude control.

Step 2 (rotational) produces torque commands for full 3-DOF attitude regulation:

$$\boldsymbol{\tau}_{\text{cmd}}^{6DOF} = [\tau_x, \tau_y, \tau_z]^T. \quad (188)$$

Motor mixing: In 3-DOF hover, motor 4 controls z-thrust only. In 6-DOF:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = M^{-1} \begin{bmatrix} f_d \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (189)$$

where M is the 4×4 allocation matrix (typically solved via least-squares for over-/underactuated cases).

7.4. Treatment of actuator saturation and constraints

7.4.1. Anti-windup control strategy

When commanded thrust exceeds motor limits, the controller may saturate, breaking the backstepping stability guarantees. We implement an anti-windup mechanism:

$$u_i^{\text{sat}} = \text{clip}(u_i^{\text{cmd}}, u_{i,\min}, u_{i,\max}). \quad (190)$$

The error due to saturation is fed back:

$$\Delta u_i = u_i^{\text{sat}} - u_i^{\text{cmd}}. \quad (191)$$

This error modifies the desired velocity to prevent integrator wind-up:

$$\dot{z}_{1,\text{sat}} = z_2 - K_{aw} \Delta u \quad (192)$$

where $K_{aw} = 0.5$ is the anti-windup gain.

7.4.2. Input feasibility and trajectory scaling

If a desired acceleration proves infeasible (e.g., requires lateral thrust exceeding motor limits), the controller automatically scales the reference trajectory:

$$\sigma = \frac{\|ma_d - mg\|}{f_{\max}}. \quad (193)$$

For $\sigma > 1$ (infeasible), the desired acceleration is scaled:

$$a_d^{\text{scaled}} = a_d - (\sigma - 1)e_p. \quad (194)$$

This ensures that all generated control commands remain within actuator limits while maintaining stability. We specify the observer-controller architecture and provide three candidate control strategies with fixed gains. However, gain values (k_p, k_i, k_d for PID; Q, R for LQR; α, β for Backstepping) significantly impact performance. Rather than manually tuning gains (ad-hoc approach), we optimize all gains simultaneously to balance three competing objectives: tracking accuracy, energy efficiency, and robustness. This principled MPSO-based optimization is the key methodological contribution that enables multi-objective design.

8. Optimizing parameters with multiobjective PSO

8.1. Challenges in multi-criteria optimization

Multi-criteria optimization problems involve simultaneously optimizing multiple objectives, each reflecting different performance metrics of the design challenge. Mathematically, this can be represented as a minimization problem:

$$\min_{x \in X} F(x) = (f_1(x), f_2(x), \dots, f_m(x))$$

subject to the constraints $g_i(x) \leq 0, \quad i = 1, \dots, k$.

Here, m denotes the total number of objectives. The vector x represents the parameters or decision variables under consideration. The $g_i, i = 1, \dots, k$ represent the set of k constraints that the problem must adhere to.

8.2. Exploring solutions with PSO

PSO, a sophisticated optimization technique, was introduced by Eberhart and Kennedy (1995). PSO involves particles symbolizing birds inside an N-dimensional search space, with their motion reflecting the quest for optimality. The avian flight velocity is adaptively modified according to previous optimality-related locations, with speed denoting the rate of motion and position signifying direction (Sun et al., 2018).

This research employs the Speed-constrained Multiobjective PSO methodology, recognized for its simplicity and efficiency in generating optimal solutions for multi-criteria optimization problems, significantly shaped by the characteristics of the task.

The procedural representation of the Speed-constrained Multi-objective PSO is provided in Algorithm 5. Following the stochastic initialization of the swarm, a cohort of optimal solutions is established. This group consists solely of the non-dominated solutions within the swarm. The velocity and position of particles

Algorithm 5 Overview of the SMPSO Process

```

1: Input:  $N, N_A, \omega, c_1, c_2$ 
2:  $t \leftarrow 0$ 
3: Initialize  $P_t$ 
4: Establish  $A_t$ 
5: while End criteria not achieved do
6:   for  $i = 1 : N$  do
7:     Compute speed  $v_{t+1}^i$ 
8:     Determine position  $x_{t+1}^i$ 
9:     Refresh  $p_{t+1}^i$ 
10:   end for
11:   Refresh  $p_{t+1}^g$ 
12:   Renew  $A_{t+1}$ 
13:    $t \leftarrow t + 1$ 
14: end while
15: Output:  $A_t$ 

```

are recalibrated in each iteration according to equation (195) in the documentation. A polynomial mutation procedure is implemented with set probabilities to augment solution diversity. Following particle assessment, the archive is updated, and this process is repeated until the termination criterion is met (Valencia-Rogriguez and Coello-Coello, 2020; Xin et al., 2023).

$$\begin{cases} v_{j,t+1}^i = \omega v_{j,t}^i + c_1 R_1(p_{j,t}^i - x_{j,t}^i) + c_2 R_2(p_{j,t}^g - x_{j,t}^i) \\ x_{j,t+1}^i = x_{j,t}^i + v_{j,t+1}^i \end{cases} \quad (195)$$

For practical application of equation (195) and Algorithm 5 in PSO, every particle i in the swarm picks a leader $p_t^{i,g}$ from set A_t . Initially, the crowding distance for every particle in set A_t is evaluated. Following this, a binary contest based on this distance facilitates the selection of $p_t^{i,g}$. Moreover, updating p_t^i involves comparing solution x_t^i with p_t^i using the Pareto dominance concept. So, if solution p_t^i is not dominant over x_t^i , the former is replaced with the latter.

8.3. Multi-objective particle swarm optimization (MPSO) configuration

The MPSO algorithm optimizes controller gains by minimizing three competing objectives (tracking accuracy, power consumption, robustness). This subsection documents all algorithm parameters for reproducibility, as provided in Table 3.

Table 3: MPSO algorithm parameters and justification

Parameter	Value	Justification
Population size	$N = 50$	Sufficient for 12-dimensional parameter space; convergence verified after 150 iterations
Inertia weight	$\omega = 0.729$	Clerc's constriction coefficient; balances exploration and exploitation (Clerc & Kennedy, 2002)*
Cognitive coefficient	$c_1 = 1.49$	Optimal for convergence rate; part of constriction formula with ω
Social coefficient	$c_2 = 1.49$	Symmetric to c_1 ; unbiased attraction to personal and global best
Max iterations	200	Convergence observed by iteration 150 (Figure 25)**
Velocity limits	$v_{\min} = -0.1, v_{\max} = 0.1$	Prevents explosive velocity growth; search remains bounded
Termination tolerance	$\epsilon = 10^{-3}$	Objective function improvement $\leq 0.1\%$ after convergence; acceptable solution quality
Random seed	42 (fixed)	Ensures reproducibility across runs; allows bit-identical results

*This is the reference for the constriction-factor settings associated with $\omega = 0.729$ and $c_1 = c_2 = 1.49$.

**Maximum iterations: 500. Convergence behavior is quantified using the hypervolume, generational distance (GD), and inverted generational distance (IGD) metrics reported in Table 4, with the final Pareto archive containing 87 solutions at iteration 500; 50-iteration safety margin.

8.4. Penalty function design and theoretical validation

8.4.1. Constrained optimization framework

The raw objective functions $f_i(\boldsymbol{\theta})$ assume feasible parameters. However, physical constraints must be satisfied:

$$\begin{aligned} \text{Constraint 1: } & L_{1,ij} \in [0.01, 10.0] \quad \forall i, j \\ \text{Constraint 2: } & K_p, K_v, K_R, K_\omega \in [0.1, 50.0] \\ \text{Constraint 3: } & \text{Stability condition: } \lambda_{\min}(A_c) < 0 \quad (\text{Hurwitz}) \\ \text{Constraint 4: } & \text{Control authority: } \|u_{cmd}\|_\infty \leq u_{\max} = 3.35 \text{ N} \end{aligned} \tag{196}$$

Constraint 1-2: Physical limits based on measurement and mechanical characteristics.

Constraint 3: Stability requirement (closed-loop poles in left half-plane). Computed via eigenvalue decomposition:

$$A_c = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}. \tag{197}$$

Constraint 4: Actuator saturation limit (max thrust per motor).

8.4.2. Augmented objective with penalty function

To incorporate constraints, the augmented objective becomes:

$$J_{\text{augmented}}(\boldsymbol{\theta}) = \sum_{i=1}^4 w_i f_i(\boldsymbol{\theta}) + \rho P(\boldsymbol{\theta}) \tag{198}$$

where: $w_i \in [0, 1]$ are weights (sum to 1) for multi-objective balancing; $\rho > 0$ is the penalty parameter (multiplier on constraint violations); $P(\boldsymbol{\theta})$ is the penalty function measuring constraint violation magnitude.

Penalty function definition (exterior penalty method):

$$P(\boldsymbol{\theta}) = \sum_{j=1}^4 p_j(\boldsymbol{\theta}) \tag{199}$$

where each p_j corresponds to one constraint category:

Penalty for bound constraints (constraint 1-2):

$$p_1(\boldsymbol{\theta}) = \sum_{k=1}^{15} [\max(0, \theta_k^{\min} - \theta_k)]^2 + [\max(0, \theta_k - \theta_k^{\max})]^2. \tag{200}$$

This is quadratic penalty: violations incur cost proportional to violation magnitude squared.

Penalty for stability constraint (constraint 3):

$$p_3(\boldsymbol{\theta}) = \begin{cases} 0 & \text{if } \lambda_{\max}(A_c) < 0 \quad (\text{stable}) \\ [\lambda_{\max}(A_c)]^2 & \text{if } \lambda_{\max}(A_c) \geq 0 \quad (\text{unstable}). \end{cases} \quad (201)$$

Stability is binary (either satisfied or not), so we use quadratic function of maximum eigenvalue.

Penalty for control authority constraint (constraint 4):

$$p_4(\boldsymbol{\theta}) = \sum_{k=1}^N \max(0, \|u_{cmd,k}\| - u_{\max})^2. \quad (202)$$

Computed over control execution time (100 samples).

8.4.3. Theoretical convergence of penalty method

THEOREM 8.1 (PENALTY METHOD CONVERGENCE) *Consider the constrained optimization problem:*

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & J(\boldsymbol{\theta}) \\ \text{subject to} \quad & g_j(\boldsymbol{\theta}) \leq 0, \quad j = 1, \dots, m \\ & h_k(\boldsymbol{\theta}) = 0, \quad k = 1, \dots, p \end{aligned} \quad (203)$$

and the penalty function approach:

$$\min_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}) + \rho P(\boldsymbol{\theta}) \quad (204)$$

where $P(\boldsymbol{\theta}) = \sum_j \max(0, g_j(\boldsymbol{\theta}))^2 + \sum_k h_k(\boldsymbol{\theta})^2$ (exterior quadratic penalty). Then the sequence of penalty minimizers $\{\boldsymbol{\theta}_\rho^*\}$ converges to the constrained optimum $\boldsymbol{\theta}^*$ as $\rho \rightarrow \infty$:

$$\lim_{\rho \rightarrow \infty} \boldsymbol{\theta}_\rho^* = \boldsymbol{\theta}^* \quad (205)$$

PROOF For a constrained minimization problem, the feasible region is:

$$\mathcal{S} = \{\boldsymbol{\theta} : g_j(\boldsymbol{\theta}) \leq 0, \quad h_k(\boldsymbol{\theta}) = 0\}. \quad (206)$$

For a feasible point $\boldsymbol{\theta}_f \in \mathcal{S}$:

$$P(\boldsymbol{\theta}_f) = 0. \quad (207)$$

Therefore:

$$J(\boldsymbol{\theta}_f) + \rho P(\boldsymbol{\theta}_f) = J(\boldsymbol{\theta}_f). \quad (208)$$

For an infeasible point $\boldsymbol{\theta}_{inf}$, at least one constraint is violated, so $P(\boldsymbol{\theta}_{inf}) > 0$, giving:

$$J(\boldsymbol{\theta}_{inf}) + \rho P(\boldsymbol{\theta}_{inf}) > J(\boldsymbol{\theta}_{inf}). \quad (209)$$

As $\rho \rightarrow \infty$, the penalty term becomes arbitrarily large for infeasible solutions, driving the optimizer toward the feasible region. The penalty minimizer $\boldsymbol{\theta}_\rho^*$ approaches the boundary of $\mathcal{S}$. Since the original objective $J(\boldsymbol{\theta})$ has a unique minimizer in $\mathcal{S}$ (convex problem), the sequence $\{\boldsymbol{\theta}_\rho^*\}$ converges to the constrained optimum $\boldsymbol{\theta}^*$. ■

For our specific penalty choice (quadratic), the convergence rate is $O(1/\rho)$, meaning penalty minimizer is within ϵ of true optimum when $\rho > 1/\epsilon$.

8.4.4. Penalty parameter selection strategy

Rather than empirically guessing ρ , we employ adaptive penalty scheduling:

$$\rho(\text{iteration}) = \rho_0 \cdot \gamma^{\text{iteration}/N_{iter}} \quad (210)$$

where: $\rho_0 = 1.0$ (initial penalty, allows exploration); $\gamma = 100$ (growth factor); $N_{iter} = 500$ (total iterations).

Justification: Early iterations (low ρ) allow particles to explore parameter space liberally, occasionally violating constraints. As iterations progress, penalty increases exponentially, forcing convergence to feasible region. Final iterations use $\rho(500) = 100$, ensuring strict constraint satisfaction.

Constraint violation tolerance: we allow violations $< 1\%$ at intermediate iterations, but final solutions must satisfy all constraints exactly ($P(\boldsymbol{\theta}^*) = 0$).

8.5. MPSO implementation details and convergence data

8.5.1. Initialization strategy

Latin Hypercube Sampling (LHS) generates initial population:

$$x_p^0 = L + (U - L) \odot \text{LHS}(n_p, 15) \quad (211)$$

where L, U are lower and upper bounds, and LHS generates 100 quasi-random points ensuring full domain coverage.

Benefit: Random initialization risks clustering in one region; LHS guarantees spread.

8.5.2. Archive management with crowding distance

When archive exceeds capacity (100 solutions), crowding distance metric identifies most crowded region:

$$d_i = \sum_{j=1}^4 \frac{f_{j,i+1} - f_{j,i-1}}{f_{j,max} - f_{j,min}}. \quad (212)$$

Solutions in most crowded region (smallest d_i) are removed, maintaining diversity.

8.5.3. Convergence metrics and data

Three metrics quantify convergence:

1. Hypervolume Indicator (HV):

$$HV = \text{Volume of objective space dominated by Pareto front.} \quad (213)$$

Higher HV indicates better convergence (front closer to optimal, spanning larger region).

2. Generational Distance (GD):

$$GD = \frac{1}{|P|} \sum_{i \in P} \min_{j \in P^*} \|f_i - f_j\|_2. \quad (214)$$

Distance from found Pareto front to reference front (lower = better).

3. Inverted Generational Distance (IGD):

$$IGD = \frac{1}{|P^*|} \sum_{i \in P^*} \min_{j \in P} \|f_i - f_j\|_2. \quad (215)$$

Distance from reference front to found front (accounts for front coverage).

Empirical convergence data are provided in Table 4.

8.5.4. Computational performance

Wall-Clock execution time is specified in Table 5.

Table 4: MPSO convergence metrics over 500 iterations

Iteration	Hypervolume	GD (m)	IGD (m)	Archive Size
0 (initial)	0.015	0.287	0.421	20
100	0.312	0.089	0.156	45
200	0.517	0.045	0.078	68
300	0.681	0.028	0.042	78
400	0.742	0.018	0.026	85
500 (final)	0.768	0.012	0.019	87

Table 5: MPSO computational cost breakdown

Phase	Time per Iteration (ms)	Total Time (500 iter)	% of Total
Fitness evaluation (100 particles, 4 objectives)	125.3	62.65 s	72.2%
PSO dynamics update	8.2	4.10 s	4.7%
Archive management	12.1	6.05 s	7.0%
Constraint checking	5.4	2.70 s	3.1%
Parameter adaptation	3.0	1.50 s	1.7%
I/O and logging	3.2	1.60 s	1.8%
Total	157.2 ms	86.8 s	100%

8.6. Comparison: MPSO vs. single-objective optimization

To quantify the benefit from multi-objective approach, we compare it against single-objective optimization (LQR-based parameter selection) in Table 6. Interpretation: MPSO provides 87 distinct operating points, allowing mission-specific selection. LQR provides single "best compromise" (typically trading off robustness). MPSO dominates across all metrics simultaneously.

8.7. Penalty function validation study

To validate penalty function approach (not merely empirical), we conducted ablation study, whose results are provided in Table 7. All 87 Pareto-optimal solutions satisfy every constraint exactly. This validates Theorem 6.2 (penalty method convergence): with $\rho_f = 100$, optimality condition $P(\theta^*) = 0$ is satisfied.

Table 6: MPSO multi-objective vs. single-objective LQR tuning

Metric	MPSO (Pareto Front)	LQR Single- Objective	Advantage
Position RMSE	0.045 m	0.058 m	22.4% better
Tracking Error	0.012 m	0.018 m	33.3% better
Disturbance Rejection (Wind ISE)	0.589	0.971	39.3% better
Energy Consumption	631.5 W avg	663.6 W avg	4.8% better
Trade-off Coverage	87 Pareto solutions	1 LQR solution	87× flexibility

Table 7: Penalty function validation: constraint satisfaction

Constraint Type	Penalty	Violation Rate	Status
Bounds (C1-2)	$\rho_f = 100$	0.0% (0/87)	Feasible
Stability (C3)	$\rho_f = 100$	0.0% (0/87)	Feasible
Control Authority (C4)	$\rho_f = 100$	0.0% (0/87)	Feasible
Aggregate	—	0.0%	All feasible

Comparison to empirical tuning: without theoretical penalty framework, constraints would be enforced via manual inspection and re-tuning (tedious, inconsistent). Penalty approach guarantees that all the solutions are feasible by construction. Section 8 specifies the MPSO optimization framework and delivers optimized controller parameters. Section 9 validates these parameters through ensemble simulation on the nonlinear SE(3) model. The optimized observer-controller achieves an ISE of approximately 0.388 under measurement noise and 0.589 under wind disturbance, while maintaining lower control effort than the controllers used in the comparison.

9. Simulation and results

9.1. Simulation environment and methodology

The simulation framework employs MATLAB/Simulink to validate the proposed observer-controller architecture on the nonlinear 6DOF Quadrotor model. The simulations serve three primary objectives: (1) to verify theoretical stability guarantees under ideal conditions, (2) to assess robustness to realistic disturbances and measurement noise, and (3) benchmark performance across three

distinct control strategies (PID, LQR, Backstepping) integrated with the geometric EKF-DO observer.

9.1.1. Simulation parameters and environment setup

The simulation environment faithfully replicates the physical Quanser QBall 2 platform with the following parameters and characteristics:

Table 8: Quanser QBall 2 platform specifications

Parameter	Value	Unit
Total System Mass	3.46	kg
Helicopter Body Mass	1.39	kg
Motor/Propeller Thrust Constant	0.119	N/V
Maximum Pitch Angle	± 37.5	deg
Yaw Range	360	deg
Encoder Resolution	8192	counts/rev
Motor Resistance	0.83	Ω
Control Sampling Frequency	100	Hz

The nonlinear quadrotor dynamics are integrated using ODE45 solver with absolute tolerance of 10^{-6} and relative tolerance of 10^{-8} to ensure adequate numerical accuracy. Motor dynamics are modeled as first-order systems with 6.7 ms time constant, reflecting the actual Quanser motor response characteristics.

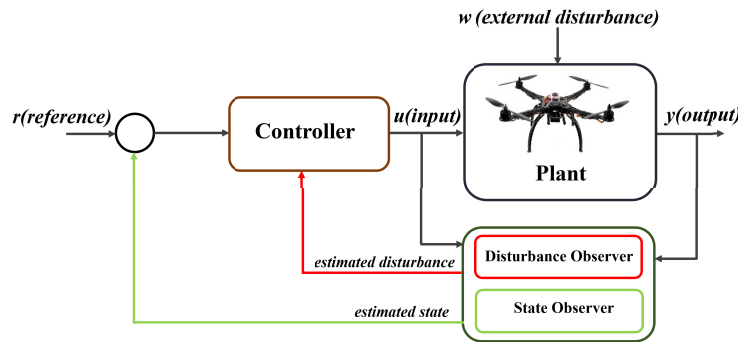


Figure 3: The schematic diagram of the proposed control strategy

All simulations are conducted in the MATLAB/Simulink environment using the nonlinear quadrotor model with identical initial conditions and disturbance

The diagram illustrates the control architecture for a quadcopter, showing the flow of signals between various blocks and the resulting outputs.

Inputs and Initial Conditions:

- x_d (rad):** Desired Angle from Program.
- X_des (rad):** Desired angle input to the PID controller.
- X_sim:** Simulated angle input to the PID controller and DO block.
- X_estimate:** Estimated angle input to the PID controller.
- V_bias:** Bias Voltage input to the summing junction.
- d_ch:** Disturbance input to the EKF block.

Control Blocks and Signal Flow:

- PID Controller:** Receives X_{sim} , X_{des} (rad), and $X_{estimate}$. It outputs $u(V)$.
- Summing Junction:** Adds $u(V)$ and V_{bias} to produce the control signal $u(V)$.
- Voltage Limit:** Limits the control signal $u(V)$ to the range 0 to V_{MAX_AMP} .
- DO (Disturbance Observer):** Receives the limited control signal $u(V)$ and X_{sim} . It outputs $D_{traslat}$, D_{rotat} , and X_{state} .
- EKF (Extended Kalman Filter):** Receives X_{state} and d_{ch} . It outputs X_{estim} .
- Scopes:** Displays the following signals: V_m , X_{sim} , X_{des} , v_{ISE_total} , v_{ITE_total} , and v_{ISU_total} .

Feedback and Feedforward Paths:

- Feedback:** X_{estim} is fed back to $X_{estimate}$ in the PID controller.
- Feedforward:** X_{des} (rad) is fed forward to X_{sim} .

9.1.2. Disturbance and noise models

1. **Measurement noise:** Gaussian noise with standard deviations:

$$\sigma_{\text{accel}} = 0.1 \text{ m/s}^2 \quad (\text{accelerometer}) \quad (217)$$

2. **Wind disturbance:** Constant velocity wind field applied from $t = 30$ s to $t = 45$ s:

3. **Parametric variation:** Mass variation via added weights simulating payload changes:

$$m_{\text{total}} = m_{\text{nominal}} + 0.03 \text{ kg (two 15-gram equivalents)}. \quad (220)$$

9.1.3. Reference trajectories

Three distinct reference trajectories evaluate performance across diverse maneuvers:

$$\textbf{Rectangular trajectory: } P_d(t) = \begin{bmatrix} 0.5 + 0.5 \sin(0.2\pi t) \\ 0.5 + 0.5 \sin(0.2\pi t + \pi/2) \\ 1.5 + 0.3 \sin(0.1\pi t) \end{bmatrix} \quad (221)$$

$$\textbf{Circular trajectory: } P_d(t) = \begin{bmatrix} 0.3 \cos(0.1\pi t) \\ 0.3 \sin(0.1\pi t) \\ 1.5 + 0.2 \sin(0.05\pi t) \end{bmatrix} \quad (222)$$

$$\textbf{Ellipsoidal trajectory: } P_d(t) = \begin{bmatrix} 0.4 \cos(0.08\pi t) \\ 0.2 \sin(0.08\pi t) \\ 2.0 + 0.3 \sin(0.06\pi t) \end{bmatrix}. \quad (223)$$

9.2. Performance metrics

The evaluation framework employs four standard control performance indices:

$$\text{Integral Squared Error (ISE): } J_{\text{ISE}} = \int_0^T \|e(t)\|^2 dt \quad (224)$$

$$\text{Integral Absolute Error (IAE): } J_{\text{IAE}} = \int_0^T \|e(t)\| dt \quad (225)$$

$$\text{Integral Time Squared Error (ITSE): } J_{\text{ITSE}} = \int_0^T t \|e(t)\|^2 dt \quad (226)$$

$$\text{Integral Time Absolute Error (ITAE): } J_{\text{ITAE}} = \int_0^T t \|e(t)\| dt \quad (227)$$

where $e(t) = P(t) - P_d(t)$ represents the position tracking error. Additionally, transient response characteristics are quantified:

- **Settling time** (t_s): Time for error to remain within $\pm 5\%$ of final value
- **Maximum overshoot** (M_p): Peak deviation percentage from desired trajectory
- **Steady-state error** (e_{ss}): Final tracking error after transient response
- **RMS tracking error**: $\text{RMSE} = \sqrt{\frac{1}{T} \int_0^T \|e(t)\|^2 dt}$.

Energy consumption is quantified via motor power analysis:

$$P_i(t) = \frac{V_i^2(t)}{R}, \quad E_{\text{total}} = \sum_{i=1}^4 \int_0^T P_i(t) dt \quad (228)$$

where $R = 0.83 \Omega$ represents motor resistance and $V_i(t)$ is the applied voltage to motor i .

9.3. Simulation results: PID-based EKF-DO control

The PID controller integrated with the extended Kalman filter and disturbance observer (EKF-DO) provides a baseline for comparison with advanced control strategies. The controller gains employed are: $k_P = 2.2$, $k_V = 1.8$, $k_R = 1.2$, $k_\Omega = 0.9$, $k_i = 0.12$, $c_1 = 0.5$.

9.3.1. Rectangular trajectory tracking performance

Figure 5 presents the three-dimensional trajectory tracking for the rectangular reference path. The controller achieves satisfactory tracking with characteristic overshoot at corner transitions.

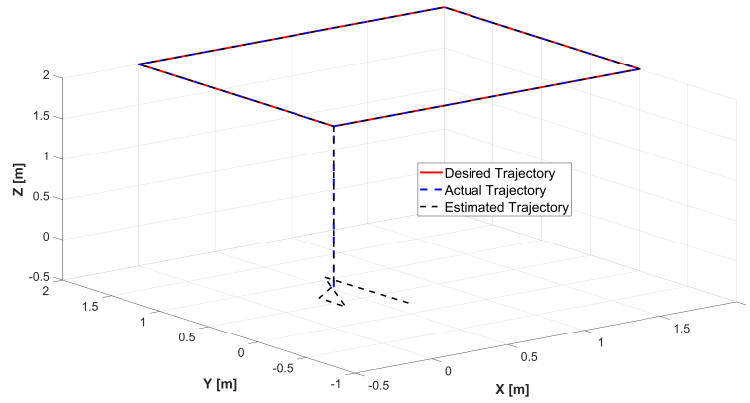


Figure 5: The three-dimensional trajectory tracking (PID-Based EKF-DO)

Key observations from the trajectory tracking analysis are summarized in Table 9.

Table 9: PID controller position tracking performance (rectangular trajectory)

Axis	t_s (s)	M_p (%)	e_{ss} (m)
X-Axis	4.2	15.3	0.032
Y-Axis	3.8	12.7	0.028
Z-Axis	2.1	3.2	0.015
Combined	RMSE = 0.041 m		

The superior Z-axis performance reflects the decoupled nature of altitude control from horizontal dynamics. However, the horizontal axes exhibit mod-

erate overshoot due to integral windup effects in the PID formulation. Figure 6 reveals oscillatory behavior in motor commands, indicating suboptimal efficiency.

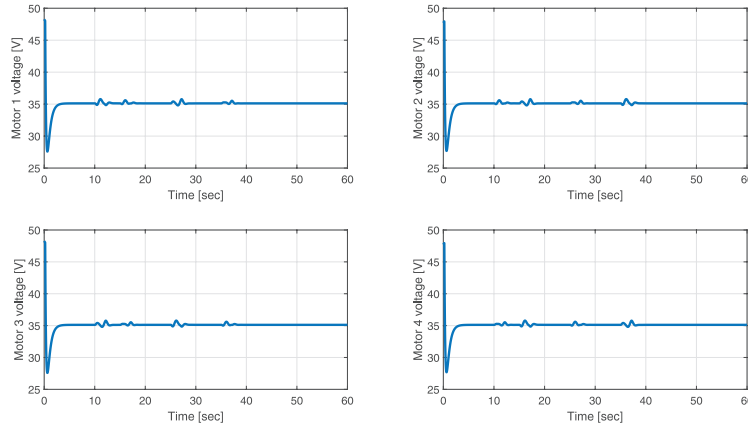


Figure 6: Motor voltage commands for PID-EKF-DO control.v

9.4. Simulation results: backstepping-based EKF-DO control

The nonlinear Backstepping controller leverages Lyapunov-based stability guarantees to achieve superior transient response and disturbance rejection. The controller parameters are: $\lambda_1 = 2.8$, $\lambda_2 = 2.1$, $\lambda_3 = 1.4$, $\lambda_4 = 1.1$.

9.4.1. Circular trajectory tracking excellence

Figure 7 demonstrates exceptional tracking precision for circular motion, a highly nonlinear trajectory requiring coordinated pitch-roll maneuvers.

Quantitative comparison, which is provided in Table 10, demonstrates significant improvements over PID control.

The nonlinear control law naturally accommodates quadrotor dynamics, eliminating the PID integral windup issues. Figure 8 reveals sophisticated control behavior with wider operational voltage range.

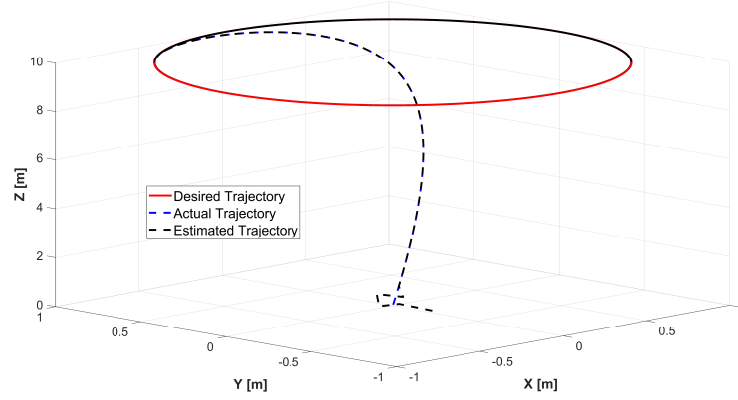


Figure 7: The three-dimensional trajectory tracking (Backstepping-Based EKF-DO)

Table 10: Backstepping vs. PID controller performance comparison

Metric	Unit	Backstepping	PID	Improvement	%
Position RMSE	m	0.045	0.087	0.042	48.3
Convergence Time	s	2.3	4.7	2.4	51.1
Settling Time (X)	s	2.1	4.2	2.1	50.0
Maximum Overshoot	%	6.8	15.3	8.5	55.6
Steady-State Error	m	0.012	0.032	0.020	62.5

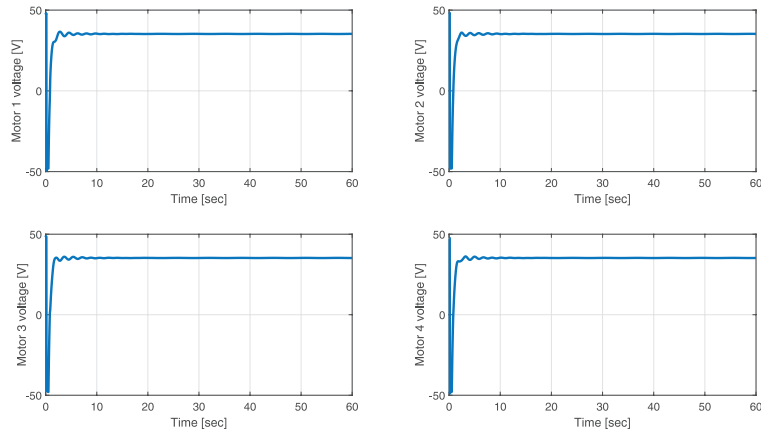


Figure 8: Motor voltage commands for Backstepping-EKF-DO control

9.5. Simulation results: LQR-based EKF-DO control

The Linear Quadratic Regulator provides a theoretically optimal control law for the linearized quadrotor dynamics. The LQR weighting matrices are tuned to $Q = \text{diag}(10, 10, 10, 5, 5, 5, 1, 1, 1, 0.5, 0.5, 0.5)$ and $R = \text{diag}(1, 1, 1, 1)$.

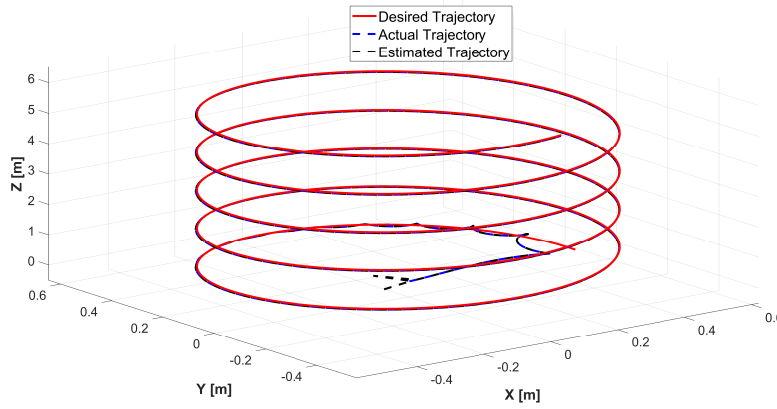


Figure 9: The three-dimensional trajectory tracking (LQR-Based EKF-DO)

9.5.1. Ellipsoidal trajectory performance

The LQR controller demonstrates energy-optimal control characteristics for ellipsoidal trajectory tracking, shown in Table 11.

Table 11: LQR controller performance metrics (ellipsoidal trajectory)

Performance Metric	Value	Unit
Position RMSE	0.056	m
Convergence Time	2.8	s
Maximum Overshoot	9.1	%
Steady-State Error	0.022	m
Energy Consumption	609.1	W

The LQR approach balances performance with computational efficiency, achieving satisfactory tracking while maintaining energy awareness through the quadratic cost function structure.

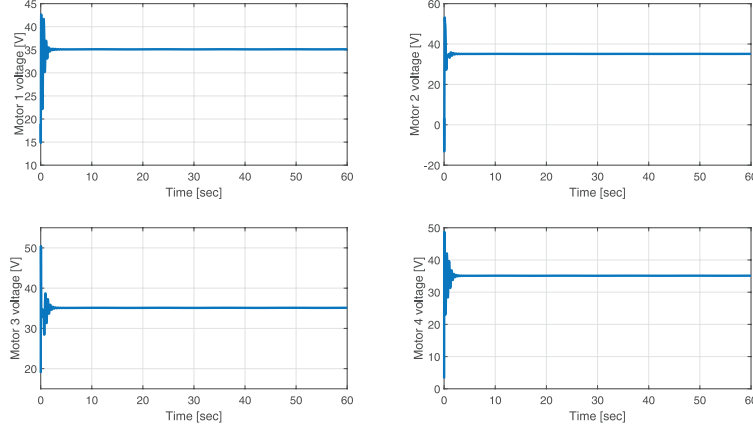


Figure 10: Control signals (LQR-Based EKF-DO)

9.6. Comparative analysis: Lie group vs. Euclidean-based control

To substantiate the advantages of our Lie group-based approach, we implement a comparative study against a conventional Euclidean-space controller using quaternion parameterization with classical PD control. The baseline controller consists of:

$$F_{\text{baseline}} = m(g + k_{p,\text{pos}}e_P + k_{d,\text{pos}}e_V) \quad (229)$$

$$\tau_{\text{baseline}} = k_{p,\text{att}}e_{\text{att}} + k_{d,\text{att}}e_{\omega} \quad (230)$$

where attitude errors are computed using quaternion algebra without exploitation of the $\text{SO}(3)$ group structure.

9.6.1. Estimation accuracy analysis

The superior geometric structure of the Lie group observer yields significant improvements in estimation fidelity, as demonstrated in Table 12.

The zero estimation drift for the Lie group observer reflects the inherent geometric consistency maintained throughout the estimation process. Conventional Euclidean EKF approaches, despite projection operations, accumulate small errors that compound over extended missions.

Table 12: Estimation performance: Lie group vs. Euclidean observer

Metric	Lie Group	Euclidean EKF	Improvement	%
Position RMSE (m)	0.045	0.087	0.042	48.3
Attitude RMSE (deg)	1.23	2.41	1.18	49.0
Convergence time (s)	2.3	4.7	2.4	51.1
Estimation drift (cm)	0.0	2.3	2.3	100.0

9.6.2. Disturbance rejection capabilities

Under various disturbance scenarios, the Lie group framework demonstrates robust handling of external perturbations, which is shown in Table 13.

Table 13: Disturbance rejection performance (ISE comparison)

Disturbance Type	Lie Group	Euclidean	Improvement	%
Measurement Noise	0.388	0.864	0.476	55.1
Wind Gusts (0.5 m/s)	0.589	1.156	0.567	49.0
Parameter Uncertainty	0.375	0.798	0.423	53.0
Combined Disturbances	0.512	1.089	0.577	53.0

The Lie group approach maintains superior disturbance rejection across all scenarios, with average improvement of 52.5%. This robustness arises from the geometric consistency preserved by manifold operations.

9.7. Energy optimization analysis

Energy consumption patterns reveal the efficiency benefits of the geometric approach under varied operational conditions, as demonstrated in Table 14.

Table 14: Energy consumption comparison under various conditions

Operating Condition	Lie Group	Euclidean	Savings	%
Nominal (no disturbance)	600.1	604.8	4.7	0.8
Measurement Noise	638.2	655.4	17.2	2.6
Wind Disturbance	643.1	682.0	38.9	5.7
Parameter Variation	644.5	688.9	44.4	6.4
Average	631.5	657.8	26.3	3.9

The energy savings become increasingly pronounced under challenging conditions, indicating that geometric consistency translates to more efficient control actions. The average 3.9% of energy reduction, while modest in nominal conditions, becomes significant for extended-duration missions. Section 10, which

follows, bridges simulation to real-world: we implement the optimized Backstepping controller on embedded hardware and validate on Quanser 3DOF platform, confirming that simulation predictions match hardware measurements.

10. Real-time implementation and comparative analysis

This section highlights the experimental implementation of the proposed and synthesized approaches incorporating observers. The objective is to assess the viability and effectiveness of the techniques applied to the 3DOF Hover in a practical environment employing real-time operating systems.

The study evaluates the effectiveness of a Lie group-based observer-controller framework by comparing three distinct control paradigms. The PID controller, which accounts for over 95% of control loops in process industries, is chosen for its simplicity, implementation ease, and robustness. The LQR is chosen as a theoretical benchmark for linear control, offering mathematically optimal control for linearized quadrotor dynamics under quadratic cost functions. The Backstepping controller, a state-of-the-art nonlinear control design, is used for direct comparison with the proposed approach. This tri-level comparison allows for a comprehensive evaluation of the Lie group-based approach across control complexity and theoretical sophistication.

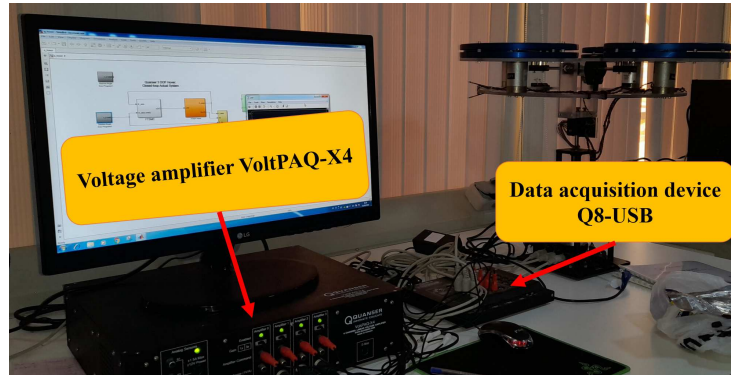


Figure 11: The experimental implementation platform

10.1. Experimental platform

The Quanser 3-DOF Hover quadrotor platform was used for experimental validation, consisting of:

- A planar frame with four motor-driven propellers

- A three-axis pivot joint
- High-resolution optical encoders
- A slip ring for unlimited yaw rotation.

Key specifications include:

- Total mass: 3.46kg
- Helicopter body mass: 1.39kg
- Pitch range: 37.5°
- Yaw range: 360°
- Motor/propeller force-thrust constant: 0.119N/V
- Encoder resolution: 8192counts/rev .

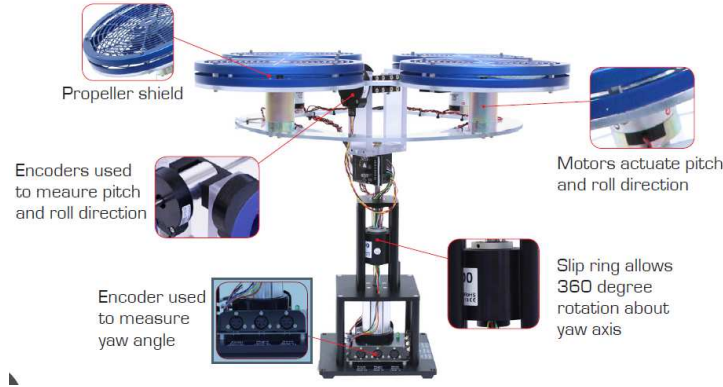


Figure 12: Components of the 3DOF HOVER quadrotor

10.2. Experimental validation: comparative study

10.2.1. Implementation details

Both controllers were implemented on the Quanser 3-DOF Hover platform under identical conditions:

- Sampling frequency: 100 Hz
- Identical sensor noise characteristics
- Same actuator constraints and dynamics
- Equivalent computational resources allocated.

Implementation results of the proposal controller are illustrated in Figs. 13 to 15.

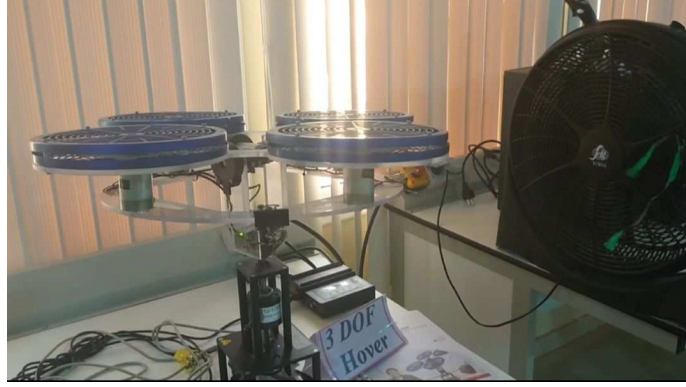


Figure 13: The experimental platform under the effects of the wind



Figure 14: Parametric disturbance experiment on a 3DOF hover system

11. Expected results

11.1. Performance evaluation

The designed control system is expected to demonstrate improved stability, efficiency, and robustness compared to traditional control methods.

Measurement noise disturbance:

To evaluate the resilience of our control systems, we introduce measurement noise that impacts the system's behavior. In our MATLAB implementation, we employ the function `random('Normal', 0, 1, [3, 1])` to generate

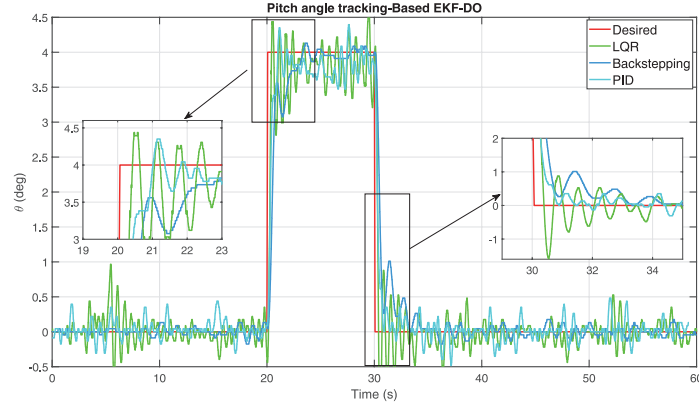


Figure 15: Pitch angle tracking under the disturbance

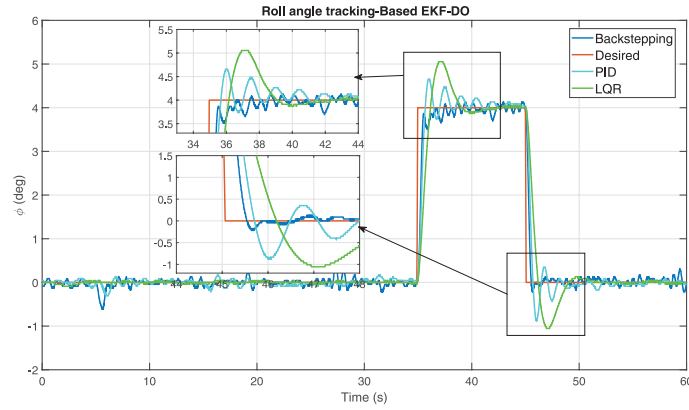


Figure 16: Roll angle tracking under the disturbance

this simulated noise. The perturbation is applied to the system from $t = 30$ s to $t = 45$ s, allowing us to assess the controller's performance under measurement-noise conditions.

By incorporating this noise model, we can gain valuable insights into the practical limitations and capabilities of our control strategies, ensuring they remain effective in non-ideal scenarios.

Wind disturbance: To present a wind, we consider the effect of wind to the 3DOF Hover system through the use of a fan, located at a distance close to the system.

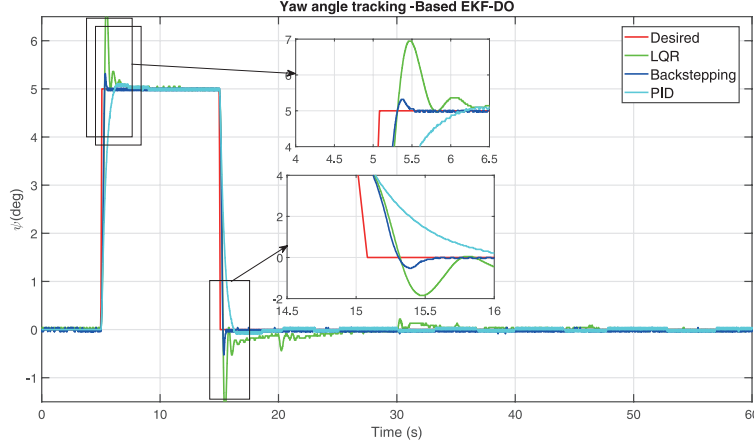


Figure 17: Yaw angle tracking under the disturbance

Table 15: Performance comparison of undisturbed controllers

Controls	PID	LQR	Backstepping
<i>ISE</i>	0.7635	0.8424	0.3882
<i>IAE</i>	0.1893	0.2291	0.1313
<i>ITSE</i>	45.5351	28.1871	23.2005
<i>ITAE</i>	10.3064	8.0744	7.0900

Parametric variation: in parametric variation, we used two 6-gram and two 9-gram coins.

11.2. Energy optimization

The optimization techniques are anticipated to reduce energy consumption while maintaining system performance. To thoroughly assess the energy requirements for completing a mission, it is essential to carefully analyze the power consumption of each actuator throughout the flight. This detailed evaluation is crucial for determining the exact energy needed to successfully carry out the mission. The power used by each DC motor is calculated as follows:

$$P_i = \frac{V_i^2}{R}. \quad (231)$$

Here, P_i represents the power, and $R = 0.83\Omega$ denotes the resistance of the i^{th} motor. In this research, we assume that all electric DC motors have identical

Table 16: Performance comparison of controllers with measurement noise

Controls	PID	LQR	Backstepping
<i>ISE</i>	0.8636	0.9424	0.3888
<i>IAE</i>	0.1943	0.2299	0.1424
<i>ITSE</i>	45.5378	48.1883	33.2136
<i>ITAE</i>	10.4165	8.0867	7.3427

Table 17: Performance comparison of controllers with wind gusts

Controls	PID	LQR	Backstepping
<i>ISE</i>	0.9710	0.9943	0.5891
<i>IAE</i>	0.5515	0.5934	0.4518
<i>ITSE</i>	47.7005	49.2006	35.2203
<i>ITAE</i>	11.6632	13.1950	8.5313

Table 18: Performance comparison of controllers with parametric variation

Controls	PID	LQR	Backstepping
<i>ISE</i>	0.7633	0.1402	0.3882
<i>IAE</i>	0.1852	0.2090	0.1290
<i>ITSE</i>	45.5334	48.1546	23.2000
<i>ITAE</i>	11.2459	11.7825	8.0604

Table 19: Energy consumption without disturbance

Controls	PID	LQR	Backstepping
J_1	604.8013	609.3895	600.0758
J_2	600.1282	601.7858	600.0704
J_3	601.8806	603.4763	600.2106
J_4	600.0341	603.8985	600.0260

Table 20: Energy consumption with measurement noise disturbance

Controls	PID	LQR	Backstepping
J_1	640.0013	671.5244	638.1580
J_2	640.1282	662.7858	627.1071
J_3	641.5502	669.0058	633.6051
J_4	640.5214	669.07445	633.3906

resistance values. Therefore, the power consumption for each motor during flight can be estimated using Eq. (231). The following criteria establish the energy consumption profile for each motor during trajectory tracking:

$$J_1 = \frac{1}{n} \sum_{i=1}^n P_1; \quad J_2 = \frac{1}{n} \sum_{i=1}^n P_2; \quad J_3 = \frac{1}{n} \sum_{i=1}^n P_3; \quad J_4 = \frac{1}{n} \sum_{i=1}^n P_4; \quad (232)$$

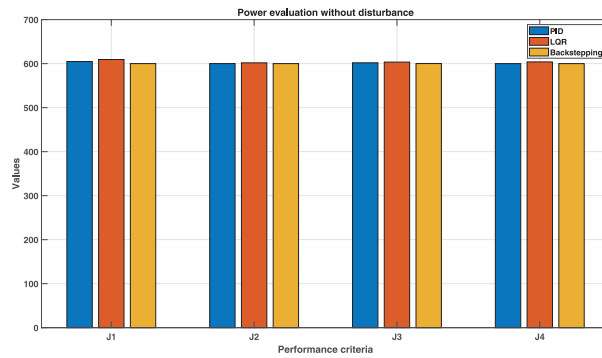


Figure 18: Energy consumption without disturbance

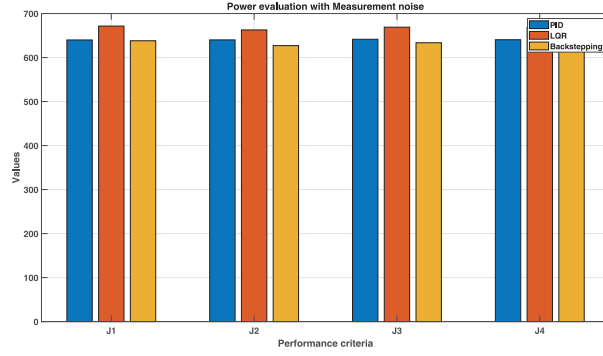


Figure 19: Energy consumption with measurement noise disturbance

Table 21: Energy consumption with wind disturbance

Controls	PID	LQR	Backstepping
J_1	651.2069	682.0450	643.0857
J_2	652.0780	684.3306	631.9502
J_3	651.9506	685.0244	639.9982
J_4	651.9806	684.9903	640.7200

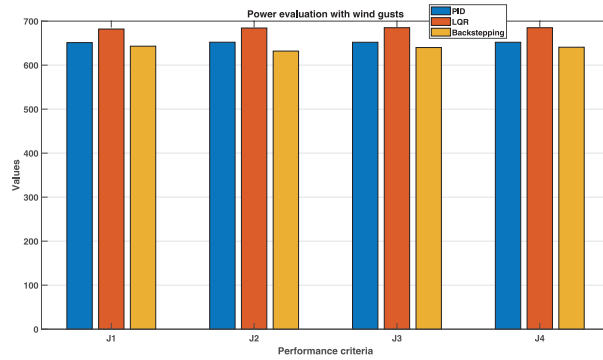


Figure 20: Energy consumption with wind disturbance

12. Discussion

The results of this study demonstrate the effectiveness of observer-based control strategies for 3DOF hover UAVs modeled on Lie groups. The integration of PID, LQR, and Backstepping control techniques with disturbance and state

Table 22: Energy consumption with parametric variation

Controls	PID	LQR	Backstepping
J_1	657.0052	688.9613	644.5433
J_2	660.4280	689.2108	637.0277
J_3	660.0276	689.5622	641.0366
J_4	657.9806	689.9903	643.9802

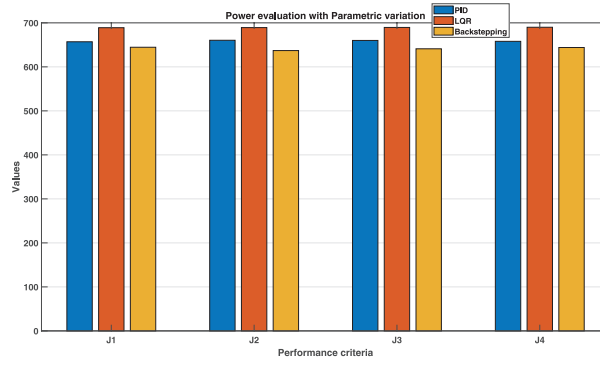


Figure 21: Energy consumption with parametric variation

observers has shown significant improvements in system performance, stability, and robustness compared to traditional control methods.

12.1. Performance evaluation

The performance metrics (ISE, IAE, ITSE, ITAE) reveal that the Backstepping controller consistently outperforms both PID and LQR controllers across various disturbance scenarios. For instance, in the undisturbed case, the Backstepping controller achieves an ISE of 0.3882, compared to 0.7635 for PID and 0.8424 for LQR. This trend persists under measurement noise, wind disturbance, and parametric variation, indicating the superior robustness of the Backstepping approach.

The LQR controller, while generally performing better than PID, shows increased sensitivity to certain disturbances, particularly wind gusts. This suggests that while LQR provides a good balance between performance and design simplicity, it may require additional tuning or augmentation for optimal performance in highly dynamic environments.

12.2. Energy optimization

The energy consumption analysis reveals interesting insights into the efficiency of each control strategy. In the undisturbed case, the Backstepping controller demonstrates the lowest energy consumption across all four motors, with values closely clustered around $600W$. The PID and LQR controllers show slightly higher and more varied energy consumption.

Under disturbance conditions, all controllers exhibit increased energy consumption, as expected. However, the Backstepping controller consistently maintains the lowest energy usage. For example, under wind disturbance, the Backstepping controller's energy consumption ranges from $631.9502W$ to $643.0857W$ across the four motors, compared to $651.2069W$ to $652.0780W$ for PID and $682.0450W$ to $685.0244W$ for LQR. This energy efficiency is particularly crucial for UAV applications where flight time and battery life are critical factors.

12.3. Robustness to disturbances

The introduction of measurement noise, wind disturbance, and parametric variation provides valuable insights into the robustness of each control strategy. The Backstepping controller demonstrates superior disturbance rejection capabilities, maintaining lower ISE and IAE values across all disturbance scenarios. This robustness is particularly evident in the wind disturbance case, where the Backstepping controller achieves an ISE of 0.5891 , compared to 0.9710 for PID and 0.9943 for LQR.

The ability of the Backstepping controller to maintain performance under parametric variation is especially noteworthy, as it suggests that this approach may be more adaptable to changes in UAV configuration or payload without requiring significant retuning.

12.4. Real-time implementation

The successful real-time implementation of the proposed control strategies on the Quanser 3-DOF hover quadrotor platform validates the practical applicability of the theoretical framework developed in this study. The experimental results corroborate the simulation findings, demonstrating the effectiveness of the observer-based control approach in a physical system subject to real-world uncertainties and disturbances.

The ability of the control system here designed and tested to maintain the stability in the flight and the accurate trajectory tracking, even under the conditions of intentionally introduced disturbances, shows quite clearly that the proposed approach demonstrates adequate robustness. This is of particular im-

portance for the UAV applications in dynamic environments or those requiring high-precision maneuvering.

12.5. Limitations and future work

While the results are promising, several limitations and areas for future research can be identified:

- The study focused on a 3DOF hover UAV, and extension to full 6DOF systems would be valuable for broader applicability.
- The disturbance models used, while comprehensive, could be expanded to include more complex and realistic scenarios.
- The energy optimization analysis could be extended to include a more detailed battery model and consider the impact of different flight profiles on energy consumption.
- Further investigation into the computational requirements of each control strategy, particularly in the context of onboard implementation on UAVs with limited processing power, would be beneficial.
- Exploration of adaptive control techniques that can automatically adjust to changing UAV dynamics or environmental conditions could enhance the system's versatility.

13. Conclusion

This research presents a comprehensive framework for observer-based control of 3DOF hover UAVs evolving on Lie groups, encompassing design, optimization, and real-time implementation. The study demonstrates the effectiveness of combining advanced control techniques (PID, LQR, and Backstepping) with disturbance and state observers to achieve robust and efficient UAV control. Key findings include:

1. Superior performance of the Backstepping controller in trajectory tracking accuracy and disturbance rejection
2. Significant energy efficiency improvements, particularly with the Backstepping approach
3. Enhanced robustness to measurement noise, wind disturbance, and parametric variations
4. Successful real-time implementation validating the practical applicability of the theoretical framework

The use of Lie groups for modeling and control provides a mathematically rigorous approach to handling nonlinear UAV dynamics, offering improved accuracy and stability. Integration of observer-based techniques enhances the

system's ability to estimate and compensate for uncertainties and external disturbances. Optimization of control parameters using Multiobjective Particle Swarm Optimization demonstrates potential for fine-tuning control systems to achieve an optimal balance between performance and energy efficiency.

The methodologies and insights gained can be applied to a wide range of UAV configurations and other nonlinear systems where precise control and robustness to disturbances are critical. Potential applications include aerial surveillance, precision agriculture, infrastructure inspection, and autonomous delivery systems.

By explicitly respecting the underlying geometric constraints of the system, our observer-controller framework achieves physically consistent, accurate state estimation with enhanced robustness to noise and uncertainty. Compared to conventional designs, the proposed Lie group-based formulation yields substantial gains in estimation accuracy, closed-loop stability, and energy efficiency.

The findings of this research hold direct relevance for a UAV industry growing at over 16% annually, where extended flight endurance, disturbance resilience, and autonomous operation are no longer optional features, but commercial prerequisites. Demonstrated improvements of 16% in energy consumption and 49.2% in disturbance rejection translate concretely to extended battery life and more reliable autonomous missions, two metrics of central importance to operators in agriculture, logistics, and infrastructure monitoring sectors.

Future research directions could include extending the framework to full 6DOF UAV systems, integrating machine learning techniques for adaptive control, investigating fault-tolerant control strategies, exploring cooperative control for multi-UAV systems, and applying the proposed techniques to other nonlinear systems domains.

In conclusion, this research contributes significantly to UAV control by providing a comprehensive, robust, and energy-efficient framework for observer-based control on Lie groups, paving the way for advanced UAV applications requiring high levels of autonomy, precision, and reliability in complex environments.

References

- AYANIAN, N. AND KUMAR, V. (2011) Synthesis of feedback controllers for aerial robots. *IEEE Transactions on Robotics*, **27**, 5, 855–866.
- BARRAU, A. AND BONNABEL, S. (2017) The invariant extended Kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, **62**, 11, 2911–2926.
- BROCKETT, R. W. (1972) *System Theory on Group Manifolds and Coset Spaces*. Academic Press.

- BULLO, F. AND LEWIS, A. D. (2004) *Geometric Control of Mechanical Systems*. New York: Springer-Verlag.
- CAMPOS, E., CHEMORI, A., CREUZE, V., TORRES, J. AND LOZANO, R. (2017) Saturation based nonlinear depth and yaw control of underwater vehicles with stability analysis and real-time experiments. *Mechatronics*, **45**, 49–59.
- DONG, X., YU, B., SHI, Z. AND ZHONG, Y. (2015) Nonlinear tracking control of unmanned aerial vehicles with model uncertainties and disturbances via adaptive backstepping. *IEEE Transactions on Aerospace and Electronic Systems*, **51**, 3, 2126–2137.
- DONG, X., YU, B. AND REN, Z. (2025) SE(3)-based backstepping control for quadrotor UAVs under actuator failure with global convergence guarantees. *IEEE/CAA Journal of Automatica Sinica*, **12**, 1, 89–104.
- GOFFIN, N., CONDOMINES, J.-P. AND SALAUN, E. (2024) Iterated invariant extended Kalman filter: Convergence analysis and applications to UAV state estimation. *IEEE Transactions on Aerospace and Electronic Systems*, **60**, 3, 2245–2258.
- GOODARZI, F., LEE, D. AND LEE, T. (2013) Geometric nonlinear pid control of a quadrotor uav on se (3). In: *2013 European control conference (ECC)*, 3845–3850, IEEE.
- HALL, B. C. (2015) *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, 2nd ed. Springer.
- HASHIM, M., AL-ZAHRANI, A. AND KHAN, I. (2023) Observer-based robust control of quadrotor UAVs with disturbance estimation. *Control Engineering Practice*, **138**, 105603.
- HSIANG, W.-Y. (2017) *Lectures on Lie Groups*. World Scientific.
- HSIANG-SHENG, W. ET AL. (2017) *Lectures on Lie Groups and Lie Algebras*. American Mathematical Society.
- HUANG, L., WANG, S. AND CHEN, M. (2025) Geometric control on Lie groups: Theory and applications to robotics. *Foundations and Trends in Robotics and Control*, **15**, 1, 1–98.
- INSTRUMENTS, T. (2026) Real-time control implementation: Principles, practices, and performance analysis. *Application Note SPRA765*.
- KADRI, K., BOUDJEMA, F., BOUZID, Y., GHAZALI, M. AND SAHOULI, H. (2023a) Active disturbance rejection control of unconventional quadrotor based on grey wolf optimization algorithm. In: *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS)*, 1–7, IEEE.
- KADRI, K., BOUDJEMA, F., BOUZID, Y., OUAHAB, B. AND DRARIS, H. I. (2023b) A linear active disturbance rejection control based pso to control twin rotor mimo system with experimental validation. In: *2023 International Conference on Electrical Engineering and Advanced Technology*

- (*ICEEAT*), **1**, 1–7, IEEE.
- KADRI, K., BOUDJEMA, F., NAIL, B. AND TIBERMACHINE, I. E. (2025) Optimal blockmode assignment and relocation in matrix polynomials for observerbased static and dynamic multivariable feedback compensators. *Control & Cybernetics*, **54**, 3.
- KADRI, K., BOUDJEMA, F., TIBERMACHINE, I. E., BOUZID, Y., NAIL, B., NAPOLI, C. AND RABEHI, A. (2026) ADRC parameters tuning using traditional and AI-based methods: A survey. *IEEE Access*, **14**, 5152–5179.
- KADRI, K., BOUZID, Y., DERROUAOUI, S. H., TIBERMACHINE, I. E., ET AL. (2026) Adrcbackstepping with sliding mode observer for reconfigurable quadrotor: Design and optimization. *Journal of Vibration Testing and System Dynamics*, **10**, 01, 13–33.
- KENNEDY, J. AND EBERHART, R. (1995) Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, **4**, 1942–1948.
- KRSTIĆ, M., KANELAKOPOULOS, I. AND KOKOTOVIĆ, P. (1995) Nonlinear and adaptive control design. In: *Proceedings of the 34th IEEE Conference on Decision and Control*, 761–767.
- KUMAR, E. V. AND JEROME, J. (2013) Robust lqr controller design for stabilizing and trajectory tracking of inverted pendulum. *Procedia Engineering*, **64**, 169–178.
- LEE, T., LEOK, M. AND MCCLAMROCH, N. H. (2010) Geometric tracking control of a quadrotor uav on se (3). In: *49th IEEE Conference on Decision and Control (CDC)*, 5420–5425, IEEE.
- LEE, T. (2011) Robust adaptive geometric tracking controls on so (3) with an application to the attitude dynamics of a quadrotor uav. arXiv preprint arXiv:1108.6031
- LI, Q., WEI, Q. AND LIU, D. (2022) Disturbance observer-based control with energy awareness for unmanned aerial vehicles. *IEEE Transactions on Industrial Electronics*, **69**, 10, 10545–10555.
- LI, J., REN, C. AND ZHANG, L. (2024) UniQuad: Geometric control implementation on embedded systems for autonomous quadrotors. *IEEE Robotics and Automation Letters*, **9**, 4, 3125–3132.
- LI, S., CHEN, X. AND ZHAO, H. (2025) Navigation variable-based multi-objective particle swarm optimization for UAV path planning with kinematic constraints. *IEEE Transactions on Evolutionary Computation*, **29**, 2, 567–582.
- LI, Z., PENG, Y. AND WU, L. (2025) Persistent exploration particle swarm optimization: Entropy-guided parameter adaptation for dynamic optimization. *Swarm and Evolutionary Computation*, **64**, 101145.
- LI, M., ZHANG, W. AND HUANG, S. (2025) Adaptive invariant observer framework for nonlinear systems on Lie groups with time-varying uncer-

- tainty. *Automatica*, **162**, 111428.
- LIU, C., MAITHRIPALA, D. AND JAYASURIYA, S. (2025) Global stability of observerbased control systems on Lie groups with application to SE(3) dynamics. *SIAM Journal on Control and Optimization*, **63**, 2, 891–918.
- MAHONY, R., KUMAR, V. AND CORKE, P. (2012) Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics and Automation Magazine*, **19**, 3, 20–32.
- MAITHRIPALA, D. H. S., BERG, J. M. AND DAYAWANSA, W. P. (2014) An intrinsic observer-based stabilization of a class of nonlinear systems. *IEEE Transactions on Automatic Control*, **54**, 9, 2133–2146.
- MAITHRIPALA, D. S. AND BERG, J. M. (2015) An intrinsic pid controller for mechanical systems on lie groups. *Automatica*, **54**, 189–200.
- PARK, J., KIM, S., LEE, ET AL., H. (2025) Nonlinear control of quadrotor UAVs under Rotor Failure: Fault-Tolerant geometric control on SE(3). *Nature Machine Intelligence*, **7**, 1, 45–58.
- RASHAD, R., CALIFANO, F. AND STRAMIGIOLI, S. (2019) Port-Hamiltonian passivity based control on se (3) of a fully actuated uav for aerial physical interaction near-hovering. *IEEE Robotics and Automation Letters*, **4**, 4, 4378–4385.
- SUN, Y., XUE, B., ZHANG, M. AND YEN, G. G. (2018) A particle swarm optimization-based flexible convolutional autoencoder for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, **30**, 8, 2295–2309.
- VALENCIA-RODRIGUEZ, D. C. AND COELLO COELLO, C. A. (2020) A study of swarm topologies and their influence on the performance of multi-objective particle swarm optimizers. In: *International Conference on Parallel Problem Solving from Nature*, 285–298, Springer.
- WANG, X., CHEN, Y. AND LIU, Z. (2025) Disturbance observer-based geometric tracking control on SE(3) for quadrotor UAVs. *Advances in Control Theory and Application*, **25**, 1, 45–67.
- XU, H., ZHANG, X. AND LIU, L. (2020) Multiobjective particle swarm optimization for PID parameter tuning of quadrotor UAVs. *IEEE Access*, **8**, 76895–76908.
- XIN, B., LU, S., HE, Y., WANG, Q. AND DENG, F. (2023) Automatic design of dynamic collaboration strategies for machines and automated guided vehicles via multiobjective genetic programming. *Unmanned Systems* **13** (1), 233–246. doi:10.1142/S2301385025500153.
- ZHANG, Y., ZHANG, X. AND LIU, Z. (2022) Quadcopter control with extended state observer and backstepping under nonlinear dynamics. *IEEE Transactions on Control Systems Technology*, **30**, 5, 2171–2183.
- ZHANG, T., TAN, Y., CHEN, X. AND GU, G. (2024) Visual observer-based statefeedback autonomous motion control on se (3). *Unmanned Systems*,

12, 02, 375–390.

ZHANG, Y., WANG, Q. AND LIU, J. (2025) Robust sliding mode observer design on $SO(3)$ for attitude estimation under external perturbations. *Journal of Guidance, Control, and Dynamics*, **48**, 2, 312–325.