

CALOS: Control-Affine Lyapunov On-manifold Safety Layer for Safe Deep Reinforcement Learning for Quadrotors

Fabrizio Cesareo, Sebastiano Mengozzi, Nicola Mimmo, Andrea Acquaviva

Department of Electrical, Electronic, and Information Engineering “Guglielmo Marconi” — DEI
University of Bologna, Italy

Abstract—Deep Reinforcement Learning has demonstrated remarkable capability in quadrotor control, yet learned policies offer no guarantee of respecting safety constraints during training or deployment. We present CALOS (Control-Affine Lyapunov On-manifold Safety), a runtime safety layer that enforces attitude constraints on a quadrotor without modifying the underlying learning algorithm. CALOS formulates four tilt-angle inequalities and a Lyapunov descent condition as a single quadratic program whose solution is the minimum-norm correction to the nominal torque output of the policy. The quadratic program is solved exactly via active-set enumeration over the three-dimensional torque space, with a computational cost low enough to enforce constraints in real time across thousands of parallel simulation environments, as required by modern massively parallel Deep Reinforcement Learning training. Evaluated on trajectory-tracking tasks in NVIDIA Isaac Lab, CALOS reduces lateral tracking error by 55–60% relative to an unconstrained Proximal Policy Optimization baseline while achieving zero attitude-constraint violations on the training trajectory. By restricting exploration to safe regions of the state space, the safety layer also accelerates training convergence and improves data efficiency without producing suboptimal policies.

I. INTRODUCTION

Deep Reinforcement Learning (DRL) policies trained in simulation can push quadrotors to the edge of their flight envelope [1], yet the resulting controllers inherit no classical stability guarantee [2]. Safe DRL [3] addresses this gap through two broad strategies: modifying the optimization criterion via Constrained Markov Decision Processes (CMDPs) [4], [5], or modifying the action at runtime via shielding [6] or control barrier functions [7]. A persistent tension remains: CMDP solvers can destabilize learning [8], while runtime filters applied during training risk correcting actions so aggressively that the gradient signal is degraded and the policy fails to improve.

A further limitation is architectural: many runtime approaches enforce a single certificate type and compose them sequentially. ATACOM [9], [10] projects actions onto the constraint manifold tangent space but lacks energy-based certificates, offering no guarantee on rotational energy dissipation. Lyapunov-based safe policy optimization [11] enforces energy dissipation but treats constraints independently, so satisfying one may violate another. The Barrier-Lyapunov Actor-Critic [12] integrates both but requires constrained actor updates and a backup controller, coupling safety enforcement to the learning algorithm. None jointly optimize over tilt

and Lyapunov constraints in a single step, leaving sequential composition as the only option, with the feasibility issues this entails.

This paper presents CALOS (Control-Affine Lyapunov On-manifold Safety), a runtime safety layer that addresses these limitations. CALOS unifies attitude-tilt constraints and a Lyapunov stability condition into a single quadratic program (QP), producing the minimum-norm torque correction at each control step. The layer is transparent to the learning algorithm, here Proximal Policy Optimization (PPO) [13], and compatible with any policy gradient method.

II. PROBLEM FORMULATION

We consider quadrotor trajectory tracking: a learning-based controller must follow a time-varying reference $p_d(t)$ while respecting attitude constraints. The quadrotor uses the Combined Thrust and Body Torques (CTBT) representation [14]: the translational and rotational dynamics are:

$$m\dot{v} = mg e_3 - T R e_3 + F_d, \quad (1)$$

$$J\dot{\Omega} = \tau - \Omega \times J\Omega, \quad (2)$$

where m is the mass, v the inertial velocity, R the rotation matrix from body to world frame, F_d the aerodynamic drag, $J \in \mathbb{R}^{3 \times 3}$ the inertia matrix, $\Omega \in \mathbb{R}^3$ the body angular velocity, and $\tau \in \mathbb{R}^3$ the body torque vector. The CTBT is input-affine which is essential for linear constraints on the input.

The DRL policy outputs $u_0 = (T, \tau_0)$, where T is the collective thrust and $\tau_0 \in \mathbb{R}^3$ is the nominal body torque. The safety layer receives τ_0 and the current state $x = (g_b, \Omega)$, comprising the body-frame gravity vector and angular velocity, and returns a corrected torque:

$$\tau = S(x, \tau_0), \quad (3)$$

where S denotes the safety operator, leaving thrust unmodified to preserve vertical authority and avoid coupling with altitude regulation.

III. SAFETY LAYER DESIGN

A. Tilt constraints and Predictive Tilt Projection

Attitude tilt is expressed via the gravity direction in body frame, $g_b = R^\top e_3$, where the subscript b denotes the body-frame representation, avoiding Euler-angle singularities. We

impose roll and pitch limits $\phi_{\max} = \theta_{\max} = 60^\circ$. Using a symplectic Euler discretization of $\dot{g}_b = g_b \times \Omega$ and the rotational dynamics, the predicted $g_{b,t+1}$ depends affinely on τ [9]:

$$g_{b,t+1} \approx g_{\text{const}}(x_t) + M(x_t) \tau, \quad (4)$$

where g_{const} collects all τ -independent terms, and $M(x_t) = \Delta t^2 [g_b]_\times J^{-1}$ is the sensitivity matrix mapping torque to changes in the body-gravity vector; $[g_b]_\times$ is the skew-symmetric matrix of g_b . Imposing $|g_{b,x}| \leq \sin \theta_{\max}$, $|g_{b,y}| \leq \sin \phi_{\max}$ yields four linear inequalities:

$$A_{\text{PTP}}(x) \tau \leq b_{\text{PTP}}(x), \quad (5)$$

where

$$A_{\text{PTP}} = \begin{bmatrix} M_x \\ -M_x \\ M_y \\ -M_y \end{bmatrix} \in \mathbb{R}^{4 \times 3}, \quad b_{\text{PTP}} = \begin{bmatrix} \sin \theta_{\max} - g_{\text{const},x} \\ \sin \theta_{\max} + g_{\text{const},x} \\ \sin \phi_{\max} - g_{\text{const},y} \\ \sin \phi_{\max} + g_{\text{const},y} \end{bmatrix} \in \mathbb{R}^4, \quad (6)$$

with M_x and M_y denoting the first and second rows of $M(x_t)$. The vector b_{PTP} encodes the tilt bounds offset by g_{const} .

The **Predictive Tilt Projection (PTP)** enforces (5) via an ATACOM-inspired [9] iterative correction.

Let $v_+(\tau) = \max\{A_{\text{PTP}} \tau - b_{\text{PTP}}, 0\}$; the torque is corrected as:

$$\tau \leftarrow \tau - A_{\text{PTP}}^\dagger k v_+(\tau) \quad (7)$$

where $k > 0$ is a tunable correction gain and $A_{\text{PTP}}^\dagger = A_{\text{PTP}}^\top (A_{\text{PTP}} A_{\text{PTP}}^\top + \lambda I)^{-1}$ is the damped pseudo-inverse of A_{PTP} , with damping factor $\lambda > 0$ ensuring numerical conditioning.

B. Lyapunov constraint and scaling

We define a Lyapunov candidate

$$V(x) = \frac{1}{2} q_g \|e_g\|^2 + \frac{1}{2} q_\omega \|\Omega\|^2 \quad (8)$$

on tilt error $e_g = g_b - g_{d,b}$ and angular velocity Ω , and enforce $\dot{V}(x, \tau) \leq -cV(x)$ with decay rate $c > 0$. Since the rotational dynamics are control-affine, $\dot{V}$ is affine in τ , yielding:

$$A_L(x) \tau \leq b_L(x), \quad (9)$$

where $A_L = q_\omega \Omega^\top J^{-1} \in \mathbb{R}^{1 \times 3}$ captures the sensitivity of rotational energy to applied torque, and $b_L = -cV(x) - B_L(x) \in \mathbb{R}$, with $B_L(x) = q_g e_g^\top (g_b \times \Omega) - q_\omega \Omega^\top J^{-1} (\Omega \times J\Omega)$ collecting the tilt-error coupling and the gyroscopic terms.

As a standalone operator, the **Lyapunov scaling**, enforces (9) by reducing torque magnitude: $\tau = s \tau_0$, $s \in [0, 1]$.

C. Sequential composition

A sequential **Cascade** composition of tilt projection, Lyapunov scaling and tilt projection check, does not guarantee joint feasibility. When the tracking error is large, $V(x)$ grows and scaling drives $s \rightarrow 0$, nullifying the torque when the controller needs maximum to recover the large error. This motivates our unified formulation.

D. CALOS: unified formulation

CALOS stacks (5) and (9) into a single system of $m=5$ linear constraints:

$$A(x) = \begin{bmatrix} A_{\text{PTP}}(x) \\ A_L(x) \end{bmatrix} \in \mathbb{R}^{5 \times 3}, \quad b(x) = \begin{bmatrix} b_{\text{PTP}}(x) \\ b_L(x) \end{bmatrix} \in \mathbb{R}^5. \quad (10)$$

The safety layer solves:

$$\tau^* = \arg \min_{\tau \in \mathbb{R}^3} \frac{1}{2} \|\tau - \tau_0\|^2 \quad \text{s.t.} \quad A(x) \tau \leq b(x), \quad (11)$$

which admits a unique solution whenever the feasible set is non-empty.

CALOS can be solved in two ways. **CALOS-P** is a projected approximation that solves (11) approximately by applying the same damped pseudo-inverse correction (7) to the unified system (A, b) , enforcing all five constraints jointly without guaranteeing the exact minimum-norm solution, trading exactness for speed in large batches, which directly reduces training time when the safety layer must be evaluated across thousands of parallel environments at every timestep.

CALOS-QP is an exact solver that exploits the fact that $\tau \in \mathbb{R}^3$ is three-dimensional, so at most three linearly independent constraints can be simultaneously active. This yields $\sum_{k=0}^3 \binom{5}{k} = 26$ candidate active sets. For each candidate I , the Karush–Kuhn–Tucker optimality conditions give $\tau = \tau_0 - A_I^\top \lambda$ and $(A_I A_I^\top) \lambda = A_I \tau_0 - b_I$ [15]. A candidate is feasible if the dual variables satisfy $\lambda \geq 0$ and the resulting torque satisfies all constraints $A \tau \leq b$; among all feasible candidates, we select the one minimizing $\|\tau - \tau_0\|^2$. If none exists, the uncorrected policy action τ_0 is applied and the actuators are clamped to their physical limits, ensuring that the system remains within hardware-safe bounds even when the QP has no feasible solution. The enumeration is fully parallelizable, essential for modern DRL training where the safety layer must scale to thousands of environments without becoming a bottleneck for the training pipeline.

IV. EXPERIMENTS

A. Setup

All agents are trained with PPO [13] using the DRL library SKRL on 4096 parallel environments in NVIDIA Isaac Lab for $\sim 10^9$ timesteps. The policy is a two-layer MLP 64 units each, with ELU activation and Gaussian output. The observation $o_t \in \mathbb{R}^{15}$ comprises body-frame velocity v_b , angular velocity Ω_b , projected gravity g_b , desired position $p_{d,b}$, and desired velocity $v_{d,b}$, all in $\mathbb{R}^3$. A reward curriculum progressively tightens the tracking precision from 0.50 m to 0.15 m throughout training, requiring increasingly accurate position following as the policy improves.

Six variants are compared: **PPO**, **PTP**, **Lyapunov scaling**, **Cascade**, **CALOS-P**, **CALOS-QP**. We evaluated the six variants over 200 episodes on three different trajectories. The first is the training Lissajous $\mathcal{L}_1$, with amplitude of 1 m and period of 5 s. The second is a harder Lissajous $\mathcal{L}_2$, with amplitude of 1.5 m and 43% higher peak velocity than the training reference. The third is a circle $\mathcal{C}$ with radius $r = 1$ m and angular rate $\omega_c = 1.777$ rad/s, chosen so that the tangential

speed matches the peak velocity of the training reference. $\mathcal{L}_2$ and $\mathcal{C}$ are *never seen during training* and are tested with a large initial position offset. A mean lateral error exceeding 0.20 m is considered a tracking failure.

B. Tracking Performance

TABLE I
MEAN LATERAL ERROR $\bar{e}_{xy}$ [M] ACROSS EVALUATION CONDITIONS.

Agent	$\mathcal{L}_1$	$\mathcal{L}_2$ (unseen)	$\mathcal{C}$ (unseen)
PPO	0.157	0.171	0.165
PTP	0.081	0.107	0.040
Lyapunov	0.127	0.256	0.569
Cascade	0.108	0.243	0.352
CALOS-P	0.062	0.076	0.043
CALOS-QP	0.069	0.113	0.042

Table I summarizes tracking accuracy. CALOS-P and CALOS-QP reduce lateral error by 55–60% on $\mathcal{L}_1$ relative to PPO. On the unseen trajectories with large initial offset, Lyapunov and Cascade fail to track the reference due to the torque nullification discussed in Section III, while both CALOS variants maintain errors below 0.08 m.

Figure 1 shows results on the circular trajectory $\mathcal{C}$. The constant curvature of this reference produces a steady-state attitude regime once the initial transient has settled, making it straightforward to distinguish the transient recovery phase from nominal tracking and to identify differences in constraint enforcement across methods. The trajectory plot (top) confirms that CALOS-QP tracks the reference closely while PPO exhibits visible drift. In the attitude plot (bottom), PTP is also shown to highlight the effect of the Lyapunov component: CALOS-QP extends PTP by adding the energy dissipation constraint (9) to the unified QP, and this additional term visibly accelerates the convergence of roll and pitch toward the steady-state regime. PTP, which enforces only tilt bounds, exhibits larger oscillations and a slower settling, because it has no mechanism to actively dissipate rotational energy during the recovery phase. CALOS-QP, by jointly enforcing both tilt and Lyapunov constraints, damps the angular transient more aggressively while still respecting the attitude bounds.

C. Safety Analysis

Table II reports attitude safety metrics on $\mathcal{L}_1$ evaluated with a large initial position offset, a condition never encountered during training, designed to stress-test the safety layer under extreme tracking error. The roll and pitch constraint enforced by CALOS is set at 60° .

CALOS-P violations last at most 3 consecutive steps, occurring only during the first simulation steps when the controller is correcting the large initial error; once the attitude enters the feasible region, no further violations occur.

In contrast, Lyapunov violations reach a maximum tilt of 180° , lasting up to 27 consecutive steps. On $\mathcal{L}_1$ *without* an initial offset, CALOS-QP achieves zero roll-violation episodes, demonstrating that the safety layer fully enforces the attitude constraints under nominal conditions. The residual

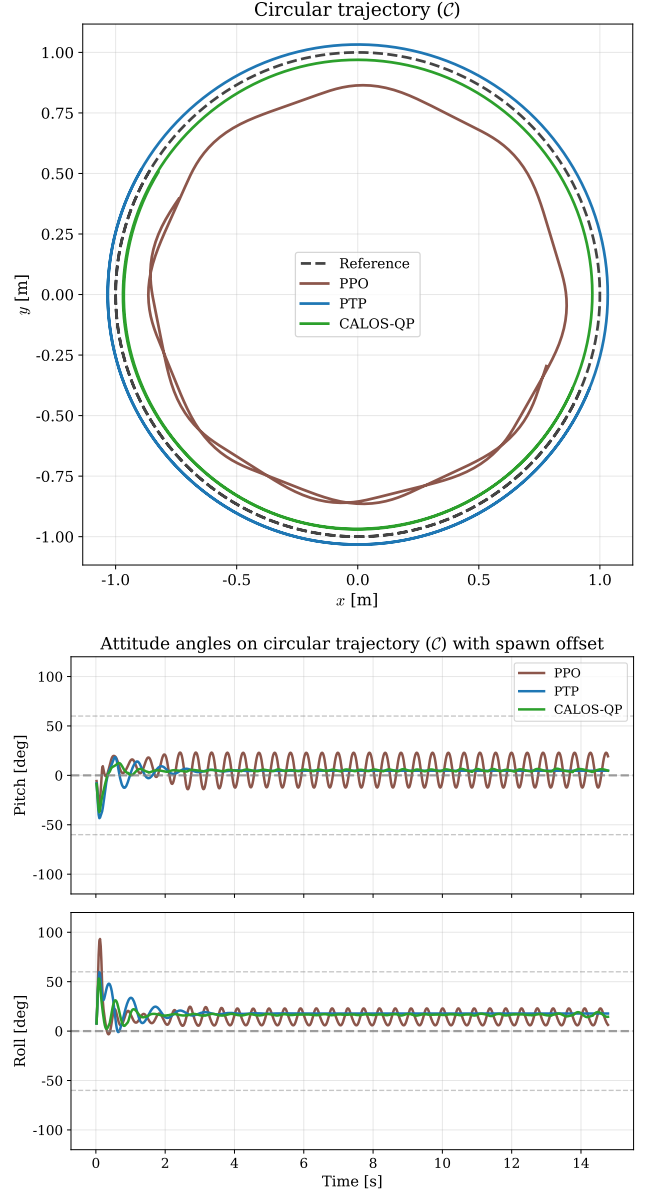


Fig. 1. Circular trajectory $\mathcal{C}$ with large initial offset. Top: xy -plane tracking (reference dashed). Bottom: roll and pitch; dashed lines mark the $\pm 60^\circ$ constraint. PTP is included to show the effect of the Lyapunov constraint: CALOS-QP settles faster due to active energy dissipation.

TABLE II
ROLL-VIOLATION STATISTICS ON $\mathcal{L}_1$ WITH LARGE INITIAL OFFSET (200 EPISODES, 60° LIMIT). AVG AND MAX REFER TO CONSECUTIVE STEPS IN VIOLATION.

Agent	$\max \phi $	Viol. eps.	Avg steps	Max steps
PPO	116.9°	4096	5.0	6
PTP	68.7°	1122	1.8	3
Lyapunov	180.0°	2988	5.2	27
Cascade	175.4°	1599	3.2	14
CALOS-P	71.5°	4096	2.6	3
CALOS-QP	99.9°	3069	4.3	9

violations observed under large initial offsets could be further reduced by including such high-error conditions in the training distribution.

D. Internalized Safe Behavior

The policy trained with CALOS retains safer and more accurate behavior even without the safety layer at test time, as shown in Table III: the mean lateral error is 55–74% lower than PPO. The constraint projection prevents unrecoverable attitudes during training, so all samples come from states where meaningful learning occurs. This restricts exploration to safe regions, accelerating convergence and improving data efficiency. Moreover, CALOS formulation preserves sufficient freedom within the feasible set, avoiding suboptimal policies: the policy internalizes the constraint boundaries as part of its nominal behavior while matching or exceeding the tracking performance of the unconstrained baseline.

TABLE III
TRACKING WITH THE SAFETY LAYER DISABLED AT TEST TIME.

Traj.	Agent	$\bar{e}_{xy}$ [m]	$\bar{R}$
$\mathcal{L}_1$	PPO	0.157	93.2
	CALOS-P (layer off)	0.062	100.9
	CALOS-QP (layer off)	0.067	100.9
$\mathcal{C}$	PPO	0.165	90.1
	CALOS-P (layer off)	0.043	99.0
	CALOS-QP (layer off)	0.041	100.3

E. Why Sequential Methods Fail

The failure of Lyapunov scaling and Cascade under large tracking errors is due to the multiplicative structure: $\tau = s\tau_0$ with $s \rightarrow 0$ when $V(x)$ is large, nullifying the torque precisely when maximum authority is needed. CALOS encodes the Lyapunov condition as $A_L(x)\tau \leq b_L(x)$, defining in $\mathbb{R}^3$ the set of all torque vectors whose component along A_L does not exceed the dissipation budget b_L . The QP can project the torque to satisfy dissipation while preserving a large component along the recovery direction.

Across all experimental conditions—all agents, trajectories, with and without initial offset—the CALOS-QP solver *always* found a feasible solution to (11). The fallback to τ_0 , i.e., applying the uncorrected policy action, was never triggered, confirming that the joint tilt-Lyapunov constraint set remained non-empty throughout all tests.

V. CONCLUSIONS

We presented CALOS, a runtime safety layer that unifies tilt and Lyapunov constraints into a single low-dimensional QP for quadrotor attitude control under DRL. The method enforces constraints in real time with minimal intervention—achieving zero attitude violations on the training trajectory and at most 3 consecutive violation steps under large perturbations—while reducing lateral tracking error by 55–74% relative to unconstrained PPO. The QP feasible set was never empty across all tests, meaning the fallback to the uncorrected policy action

was never triggered. Moreover, policies trained with CALOS internalize safe behavior, retaining comparable tracking improvement even with the safety layer disabled, confirming that the constrained exploration improves convergence and data efficiency without sacrificing optimality.

Future work includes: (i) learned, task-aware Lyapunov certificates [16], [17]: the current quadratic certificate drives toward the zero-tilt equilibrium, causing unnecessary interventions when the drone tracks at a non-zero but feasible attitude; a learned certificate that decreases toward the current reference would eliminate this; (ii) extensions to non-control-affine dynamics, since higher-fidelity models with rotor-speed dynamics or blade-flapping effects break the affine assumption that CALOS requires; (iii) sim-to-real transfer with domain randomization and data-driven safety filters [18] to handle model uncertainty on physical hardware.

REFERENCES

- [1] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, *et al.*, “Champion-level drone racing using deep reinforcement learning,” *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.
- [2] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, 3 ed., 2002.
- [3] J. García and F. Fernández, “A comprehensive survey on safe reinforcement learning,” *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.
- [4] E. Altman, *Constrained Markov Decision Processes*. CRC Press, 1999.
- [5] Y. Chow, O. Nachum, E. Duenez-Guzman, *et al.*, “A lyapunov-based approach to safe reinforcement learning,” in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [6] M. Alshiekh, R. Bloem, R. Ehlers, *et al.*, “Safe reinforcement learning via shielding,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI Press, 2018.
- [7] R. Cheng, G. Orosz, R. M. Murray, *et al.*, “End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks,” in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI Press, 2019.
- [8] C. Tessler, D. J. Mankowitz, and S. Mannor, “Reward constrained policy optimization,” in *International Conference on Learning Representations (ICLR)*, 2019.
- [9] P. Liu, D. Tateo, H. Bou-Ammar, *et al.*, “Robot reinforcement learning on the constraint manifold,” in *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.
- [10] P. Liu, H. Bou-Ammar, J. Peters, *et al.*, “Safe reinforcement learning on the constraint manifold: Theory and applications,” *arXiv preprint*, vol. arXiv:2404.09080, 2024.
- [11] Y. Chow, O. Nachum, A. Faust, *et al.*, “Lyapunov-based safe policy optimization for continuous control,” in *Reinforcement Learning for Real Life (RL4RealLife) Workshop at ICML*, (Long Beach, California, USA), 2019.
- [12] L. Zhao, K. Gatsis, and A. Papachristodoulou, “Stable and safe reinforcement learning via a barrier-lyapunov actor-critic approach,” *arXiv preprint arXiv:2304.04066v3*, 2023. arXiv:2304.04066v3.
- [13] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, *et al.*, “Proximal policy optimization algorithms,” *arXiv preprint*, vol. arXiv:1707.06347, 2017.
- [14] R. Mahony, V. Kumar, and P. Corke, “Modeling, estimation, and control of quadrotor,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [15] G. Cimini and A. Bemporad, “Exact complexity certification of active-set methods for quadratic programming,” *IEEE Transactions on Automatic Control*, vol. 62, pp. 6094–6108, Dec. 2017.
- [16] H. Dai, B. Landry, L. Yang, *et al.*, “Lyapunov-stable neural-network control,” in *Robotics: Science and Systems*, 2021.
- [17] J. Wu, A. Clark, Y. Vorobeychik, and W. Wang, “Neural Lyapunov control for discrete-time systems,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.
- [18] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, *et al.*, “Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems,” *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.