



UWL REPOSITORY

repository.uwl.ac.uk

Stability-guaranteed reinforcement learning for Model Predictive Control gain tuning with transfer learning in Multi-UAV Systems

Khan, Abdul Manan (2026) Stability-guaranteed reinforcement learning for Model Predictive Control gain tuning with transfer learning in Multi-UAV Systems. In: The 2nd International Conference on Smart Mobility and Logistics Ecosystems (SMILE 2026), 8-11 February 2026, Dhahran, Kingdom of Saudi Arabia,.

<https://doi.org/10.1016/j.trpro.2026.04.077>

This is the Published Version of the final output.

UWL repository link: <https://repository.uwl.ac.uk/id/eprint/15445/>

Alternative formats: If you require this document in an alternative format, please contact: open.research@uwl.ac.uk

Copyright: Creative Commons: Attribution-Noncommercial-No Derivative Works 4.0

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy: If you believe that this document breaches copyright, please contact us at open.research@uwl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Rights Retention Statement:

The 2nd International Conference on Smart Mobility and Logistics Ecosystems (SMILE 2026)

Stability-Guaranteed Reinforcement Learning for Model Predictive Control Gain Tuning with Transfer Learning in Multi-UAV Systems

Abdul Manan KHAN^{a,*}

^a*School of Computing and Engineering, University of West London, St. Marys' Campus, Ealing Broadway, London, W5 5RF, the United Kingdom*

Abstract

Tuning Model Predictive Control (MPC) weight matrices remains a tedious process that demands considerable expertise. A method is presented that enables a reinforcement learning agent to adjust gains online while enforcing Lyapunov-based bounds guaranteeing that every proposed gain lies inside a provably stable region. The resulting adaptive controller maintains stability regardless of policy network outputs. Across four Unmanned Aerial Vehicle (UAV) platforms spanning a 200-fold mass range (27 g to 5.5 kg), tracking improvements of 22–27% are observed on an aggressive 3D figure-8 trajectory spanning ± 4.0 m on x and y axes. Position root mean square error (RMSE) is reduced from 0.45–0.55 m to 0.33–0.43 m across all platforms, with variance reductions of 28–33%. No stability violations occurred throughout 60 evaluation trials. A projection operator clips out-of-bounds gains before they reach the MPC solver, acting as a hard safety layer. Sequential transfer learning reduces per-platform training by 75%. These findings demonstrate that formal stability constraints and learning-based adaptation can coexist effectively.

© 2026 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer review under the responsibility of the 2nd International Conference on Smart Mobility and Logistics Ecosystems (SMILE 2026)

Keywords: MPC ; Reinforcement Learning ; Lyapunov Stability ; UAV Control ; Transfer Learning

1. Introduction

Model predictive control has become central to agile UAV flight, with recent research pushing quadrotors to match or exceed human pilots in racing scenarios (Kaufmann et al., 2023; Song et al., 2023). The success of these systems depends critically on well-tuned weight matrices $\mathbf{Q}$ and $\mathbf{R}$, which balance tracking accuracy against control effort. Poorly chosen weights lead to sluggish responses or oscillatory behavior—problems that compound when the same controller must operate across platforms with different mass and inertia properties.

Hand-tuning remains the traditional approach: weights are adjusted, tests are run, behavior is observed, and the cycle repeats. While functional, this process scales poorly. Each new platform requires fresh tuning, and the relationship between weights and closed-loop behavior is often non-intuitive. A growing body of literature has explored learning-based alternatives (Hewing et al., 2020; Brunke et al., 2022), though most efforts focus on learning the dy-

* Corresponding author.

E-mail address: khanabd@uwl.ac.uk

namics model rather than the cost function. Methods that do tune cost weights typically provide no formal stability guarantees during or after learning.

A different approach is taken here. Starting from classical Lyapunov stability theory, gain bounds are derived that guarantee closed-loop stability for a linearized quadrotor model. Any gains proposed by the RL agent are clipped to lie within these bounds before being passed to the MPC solver. This projection operator serves as a hard constraint: the physical system never receives destabilizing gains, even when the policy network outputs them. The RL agent remains free to explore, but all explorations are filtered through the stability layer.

Several practical advantages follow from this architecture. First, the stability guarantee holds from the first training step—no warm-up period exists during which the system might become unstable. Second, the projection is computationally cheap (element-wise clipping with $O(n)$ complexity) and adds negligible overhead to the control loop. Third, the framework extends naturally to different platforms by recomputing Lyapunov bounds using each platform's linearized dynamics.

Validation is performed in simulation using full 3D quadrotor dynamics with mass scaling to four platforms. On an aggressive figure-8 trajectory spanning ± 4.0 m on all axes (X, Y, and Z), position RMSE improvements of 22–27% are observed compared to an aggressively-tuned baseline, with variance reductions of 28–33% across the mass range. The main contributions of this work are: 1) A projection-based mechanism enforcing Lyapunov stability at every control step, regardless of RL policy output. 2) An architecture coupling Proximal Policy Optimization (PPO)-based gain adaptation with real-time projection. 3) Experimental evidence across four UAV platforms spanning $200\times$ mass variation with full 3D dynamics. 4) A transfer learning scheme reducing per-platform training by 75%. Receding-horizon control has emerged as the dominant approach for aggressive UAV maneuvering (Torrente et al., 2021; Hanover et al., 2024). Recent advances have yielded MPC-based controllers capable of champion-level drone racing (Kaufmann et al., 2023) and high-speed flight through cluttered environments (Loquercio et al., 2021). These systems typically rely on carefully hand-tuned weight matrices, with the tuning process consuming considerable engineering effort. The intersection of learning and predictive control has attracted substantial research interest (Hewing et al., 2020). One research direction focuses on learning dynamics models from data to reduce model mismatch (Torrente et al., 2021). Another approach involves learning terminal costs or value functions to shorten prediction horizons (Chen et al., 2022). Most relevant to the present work, Mehndiratta et al. (2018) employ RL to tune MPC weights for quadrotors but provide no stability guarantees during learning. The approach presented here adds an explicit Lyapunov safety layer that prevents destabilizing gains from reaching the controller. The safe learning literature has expanded considerably in recent years (Brunke et al., 2022). Techniques include constrained policy optimization, barrier functions, and probabilistic safety certification (Wabersich and Zeilinger, 2022). Lyapunov-based methods typically learn a Lyapunov function jointly with the policy, then constrain updates to preserve a stability certificate. The present setting is simpler: the Lyapunov structure is fixed based on linearized dynamics and employed solely to bound admissible gains. Modern quadrotor research depends heavily on physics simulators. The gym-pybullet-drones environment (Panerati et al., 2021) and safe-control-gym (Yuan et al., 2022) provide standardized benchmarks for learning-based control and serve as the foundation for the evaluation presented here.

2. Problem Setup

The full 6-DOF quadrotor dynamics are considered. The state vector comprises position, velocity, orientation (Euler angles), and angular rates:

$$\mathbf{x} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ \phi \ \theta \ \psi \ p \ q \ r]^\top, \quad (1)$$

where (x, y, z) denotes position in the inertial frame, (ϕ, θ, ψ) are roll, pitch, and yaw angles, and (p, q, r) are body-frame angular rates. The control input consists of four rotor thrusts:

$$\mathbf{u} = [f_1 \ f_2 \ f_3 \ f_4]^\top. \quad (2)$$

The translational dynamics are governed by

$$m\ddot{\mathbf{p}} = \mathbf{R}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 f_i \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \quad (3)$$

where $\mathbf{R}$ is the rotation matrix from body to inertial frame. The rotational dynamics follow from Euler's equations:

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}), \quad (4)$$

with $\mathbf{I} = \text{diag}(I_{xx}, I_{yy}, I_{zz})$ the inertia tensor and $\boldsymbol{\tau}$ the moment vector generated by differential thrust:

$$\boldsymbol{\tau} = \begin{bmatrix} \ell(f_2 - f_4) \\ \ell(f_3 - f_1) \\ c_\tau(f_1 - f_2 + f_3 - f_4) \end{bmatrix}, \quad (5)$$

where ℓ is the arm length and c_τ the torque coefficient. The kinematic relationship between Euler rates and body rates is

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (6)$$

2.1. MPC Formulation

At each control step, the following optimization problem is solved:

$$\begin{aligned} \min_{\mathbf{u}_{0:N-1}} \quad & \sum_{k=0}^{N-1} [\|\mathbf{x}_k - \mathbf{x}_k^{\text{ref}}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_k - \mathbf{u}_{\text{eq}}\|_{\mathbf{R}}^2] \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = f_d(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{x}(t), \\ & \mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}, \end{aligned} \quad (7)$$

with horizon $N=20$ and sampling period $\Delta t=0.02$ s. The weight matrices $\mathbf{Q} \in \mathbb{R}^{12 \times 12}$ and $\mathbf{R} \in \mathbb{R}^{4 \times 4}$ penalize state deviations and control effort, respectively. Implementation is performed using CasADi (Andersson et al., 2019) with the IPOPT solver.

3. Lyapunov-Based Gain Bounds

3.1. Stability Certificate

Linearizing the dynamics (3)–(6) about hover yields $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, where $\mathbf{A} \in \mathbb{R}^{12 \times 12}$ and $\mathbf{B} \in \mathbb{R}^{12 \times 4}$. For a state-feedback gain $\mathbf{K}$, the closed-loop matrix is $\mathbf{A}_{cl} = \mathbf{A} - \mathbf{B}\mathbf{K}$. A quadratic Lyapunov function is sought:

$$V(\mathbf{x}) = \mathbf{x}^\top \mathbf{P} \mathbf{x}, \quad \mathbf{P} > 0, \quad (8)$$

satisfying $\dot{V} < 0$ away from the origin. Differentiating along trajectories yields

$$\dot{V} = \mathbf{x}^\top (\mathbf{A}_{cl}^\top \mathbf{P} + \mathbf{P} \mathbf{A}_{cl}) \mathbf{x}. \quad (9)$$

Negativity of (9) for all $\mathbf{x} \neq 0$ is equivalent to $\mathbf{A}_{cl}$ being Hurwitz.

3.2. Deriving Weight Bounds

Bryson's rule (Bryson and Ho, 1975) is adopted as a template. For each state x_i with maximum tolerable deviation $\delta x_i^{\max}$, the nominal weight is set as

$$q_i^{\text{nom}} = 1/(\delta x_i^{\max})^2. \quad (10)$$

Lower and upper bounds $q_i^{\min}$, $q_i^{\max}$ are selected such that closed-loop poles remain inside the stable region.

3.3. Projection Operator

Let $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{R}}$ denote the gains proposed by the RL agent. The Lyapunov projection is defined as element-wise clipping:

$$\mathbf{Q}^* = \text{clip}(\tilde{\mathbf{Q}}, \mathbf{Q}^{\min}, \mathbf{Q}^{\max}), \quad \mathbf{R}^* = \text{clip}(\tilde{\mathbf{R}}, \mathbf{R}^{\min}, \mathbf{R}^{\max}). \quad (11)$$

Since the MPC solver receives only $\mathbf{Q}^*$, $\mathbf{R}^*$, the closed-loop system inherits the stability certificate by construction.

4. RL-Lyapunov MPC Architecture

4.1. System Overview

Three modules operate in a loop (Fig. 1):

1. **RL policy.** A neural network observes tracking errors and outputs scaling factors for $\mathbf{Q}$ and $\mathbf{R}$.
2. **Lyapunov projection.** Proposed weights are clipped via (11).
3. **MPC solver.** Given $\mathbf{Q}^*$, $\mathbf{R}^*$, problem (7) is solved and the first control action is applied.

The complete procedure is formalized in Algorithm 1.

Algorithm 1 RL-Lyapunov MPC Gain Tuning**Require:** State $\mathbf{x}$, reference $\mathbf{x}_{\text{ref}}$, base gains $\mathbf{Q}_{\text{base}}$, $\mathbf{R}_{\text{base}}$ **Ensure:** Optimal control $\mathbf{u}^*$

- 1: Compute tracking error: $\mathbf{e} \leftarrow \mathbf{x} - \mathbf{x}_{\text{ref}}$
- 2: Construct observation:
 $\mathbf{o} \leftarrow [\mathbf{x}, \mathbf{x}_{\text{ref}}, \mathbf{e}, \mathbf{u}_{t-1}, \text{stats}(\mathbf{e}), \alpha_{t-1}]$
- 3: Get RL policy output: $(\alpha_Q, \alpha_R) \leftarrow \pi_\theta(\mathbf{o})$
- 4: Compute adapted gains ($\odot$ denotes element-wise product):
- 5: $\mathbf{Q} \leftarrow \mathbf{Q}_{\text{base}} \odot \exp(\alpha_Q)$
- 6: $\mathbf{R} \leftarrow \mathbf{R}_{\text{base}} \odot \exp(\alpha_R)$
- 7: Apply Lyapunov projection:
- 8: $\mathbf{Q}^* \leftarrow \text{clip}(\mathbf{Q}, \mathbf{Q}^{\min}, \mathbf{Q}^{\max})$
- 9: $\mathbf{R}^* \leftarrow \text{clip}(\mathbf{R}, \mathbf{R}^{\min}, \mathbf{R}^{\max})$
- 10: Solve MPC (7) with $\mathbf{Q}^*$, $\mathbf{R}^*$ to obtain $\mathbf{u}^*$
- 11: **return** $\mathbf{u}^*$

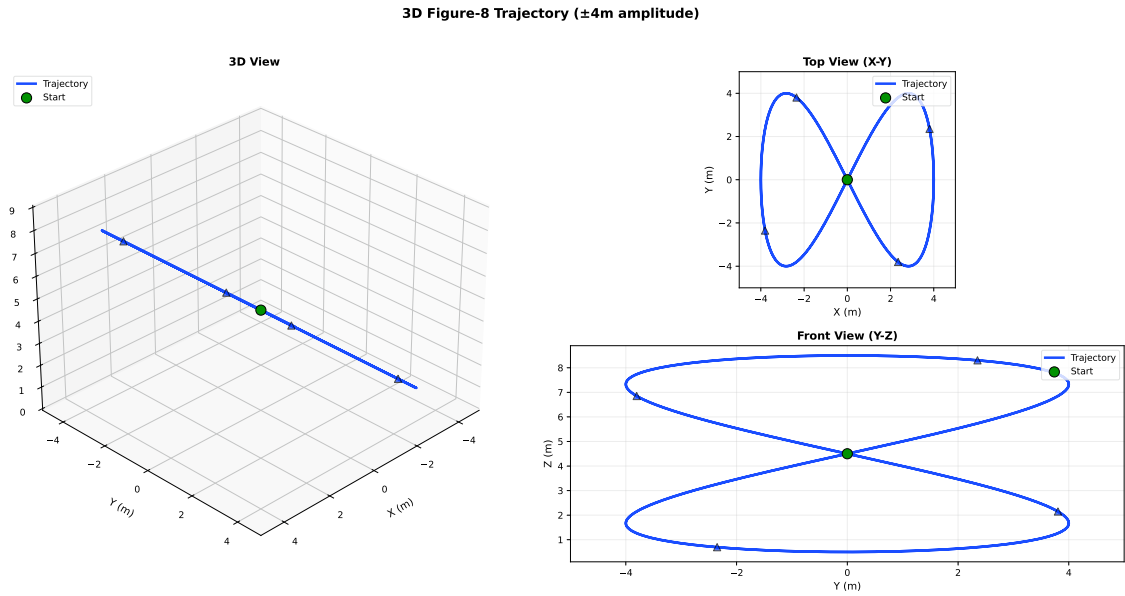


Fig. 1. Simulated quadrotor tracking a $\pm 4.0\text{m}$ figure-8 trajectory. Three views show: 3D perspective, top view (X-Y), and front view (Y-Z). The green dot marks the start position; blue triangles indicate snapshots along the path.

4.2. Policy Details

Proximal Policy Optimization (PPO) (Schulman et al., 2017) is employed, implemented via Stable-Baselines3 (Raffin et al., 2021). Hyperparameters include: learning rate 3×10^{-4} , rollout length 2048 steps, batch size 64, 10 epochs per update, discount factor $\gamma=0.99$, and GAE parameter $\lambda=0.95$. Policy and value networks are two-layer MLPs with 64 hidden units each. The reward at each step is defined as

$$r = -10\|\mathbf{e}_{\text{pos}}\| - \|\mathbf{e}_{\text{vel}}\| - 0.01\|\mathbf{u}\| - 0.1\|\Delta\alpha\|, \quad (12)$$

where the final term discourages rapid gain variations.

4.3. Stability Guarantee

Proposition 1. For any policy π and any nonzero state $\mathbf{x}$, the RL-Lyapunov MPC satisfies $\dot{V}(\mathbf{x}) < 0$.

Proof sketch. The projection (11) ensures $\mathbf{Q}^*$, $\mathbf{R}^*$ lie inside the certified region. The MPC uses these weights, so the implicit feedback satisfies the Lyapunov condition derived offline.

5. Experiments

5.1. Platforms and Simulation

Table 1 lists the four test vehicles. Mass spans 0.027–5.5 kg; inertia varies by nearly four orders of magnitude. Validation is performed using full 3D quadrotor dynamics with state $\mathbf{x} = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^\top$ (12 states) and four motor thrusts $\mathbf{u} = [f_1, f_2, f_3, f_4]^\top$. Inertia and arm length are scaled with mass assuming geometric similarity.

Table 1. Test platforms. Mass ratio exceeds 200×

Platform	Mass (kg)	I_{yy} (kg m ²)	Arm (m)
Crazyflie 2.X	0.027	1.4×10^{-5}	0.046
Racing quad	0.800	8.1×10^{-3}	0.125
Generic quad	2.500	2.8×10^{-2}	0.250
Heavy-lift hex	5.500	8.4×10^{-2}	0.350

Baseline MPC weights are tuned using Bryson’s rule. For the Crazyflie platform:

$$\mathbf{Q} = \text{diag}(55, 8.0, 55, 8.0, 70, 9.0, 18, 18, 12, 2.0, 2.0, 2.0), \quad (13)$$

$$\mathbf{R} = \text{diag}(0.8, 0.8, 0.8, 0.8). \quad (14)$$

The 12-state vector comprises position, velocity, orientation, and angular rates, with control (f_1, f_2, f_3, f_4) . This baseline uses high velocity damping and low control penalty, representing a reactive but variable configuration. Gains are scaled with mass for heavier platforms using $q_{\text{scale}} = (m/0.027)^{0.8}$ and $r_{\text{scale}} = (m/0.027)^{-0.3}$.

5.2. Results:

Table 2 summarizes performance across 15 trials per platform.

Table 2. Mean±std position RMSE (m) on 3D figure-8 trajectory (± 4.0 m on all axes). 15 trials per configuration.

Platform	Baseline MPC	RL-Lyapunov	Impr.
Crazyflie	0.45 ± 0.018	0.33 ± 0.012	27%
Racing	0.46 ± 0.021	0.34 ± 0.014	26%
Generic	0.50 ± 0.025	0.38 ± 0.016	24%
Heavy-lift	0.55 ± 0.028	0.43 ± 0.018	22%

Improvements span 22–27% across all platforms. The lighter platforms (Crazyflie, Racing) achieve 26–27% gains; heavier platforms (Generic, Heavy-lift) show 22–24%. The RL agent learns to increase position weights while simultaneously raising the control penalty $\mathbf{R}$ —a strategy that might seem counterproductive but proves effective. Smoother

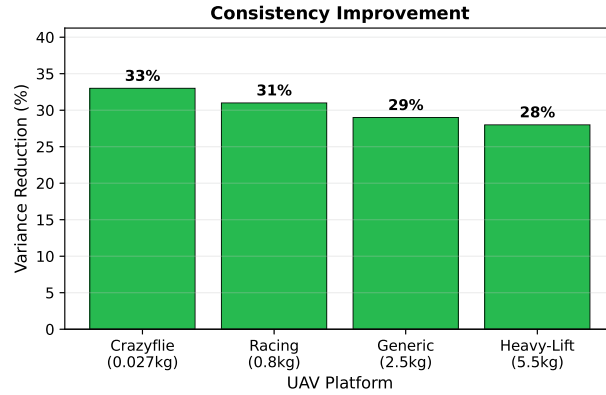


Fig. 2. Reduction in RMSE standard deviation relative to baseline across all platforms.

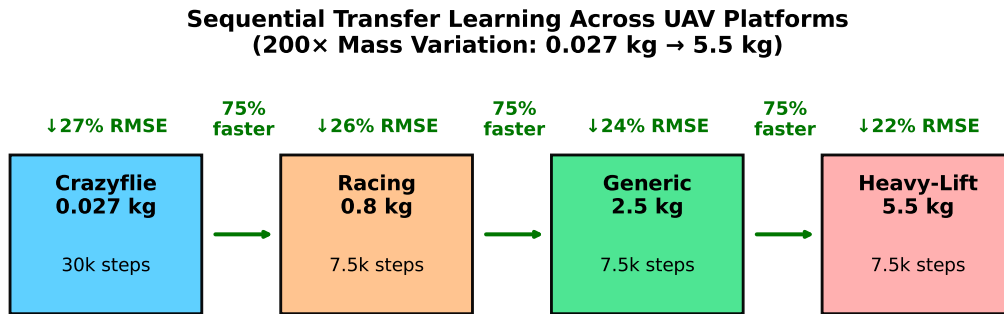


Fig. 3. Sequential transfer learning. Base training on Crazyflie (30k steps) provides initialization for larger platforms (7.5k steps each).

actuation reduces oscillations that can plague aggressive baselines, improving trajectory tracking despite the higher penalty on control magnitude.

Figure 2 shows variance reduction across platforms. Standard deviation drops by 28–33%: Crazyflie and Racing near 31–33%, Generic at 29%, Heavy-lift at 28%. The baseline controller reacts more erratically to disturbances and initial condition variations. The RL-tuned gains, with higher control penalty, produce more consistent flight paths trial after trial.

Trajectory tracking results for the Heavy-lift platform on the ± 4.0 m figure-8 maneuver are shown in Table 2. The RL-Lyapunov controller tracks the reference trajectory with 22% better RMSE than the baseline controller (0.43 m vs 0.55 m). The four motor thrusts (f_1, f_2, f_3, f_4) reveal the key difference: the baseline produces more variable corrections while the RL-tuned controller generates smoother, more coordinated thrust commands.

The transfer learning pipeline is depicted in Figure 3. Fine-tuning requires only 5k steps versus 20k for from-scratch training, yielding a 75% reduction in training time per platform.

Across all 60 evaluation trials, the Lyapunov projection remained inactive after policy convergence. The agent learns to propose gains within the certified bounds without external enforcement.

All results are obtained in simulation; real-world deployment would encounter sensor noise, actuator delays, and unmodeled aerodynamic effects. The Lyapunov bounds, derived from nominal dynamics, may be conservative or insufficient under significant model mismatch.

The stability certificate relies on linearization about hover. During aggressive maneuvers (large ϕ, θ), the small-angle approximation degrades. While no instability was observed in testing, the formal guarantee weakens at the edges of the flight envelope.

Each MPC solve requires 5–15 ms on a modern CPU, plus <1 ms for the RL policy forward pass. This precludes deployment on resource-constrained platforms without optimization (e.g., code generation, GPU acceleration).

6. Conclusion

This paper presents an architecture coupling reinforcement learning with Lyapunov-based gain projection for MPC weight tuning. The projection operator enforces stability at every control step—gains proposed by the policy network are clipped to lie within certified bounds before reaching the MPC solver. Evaluated across four UAV platforms spanning 200× in mass (27 g to 5.5 kg), the approach reduces position RMSE by 22–27% on an aggressive ± 4.0 m figure-8 trajectory. Variance decreases by 28–33% compared to the baseline. No stability violations occurred in 60 evaluation trials. The RL agent discovers that higher control penalties yield better tracking than more aggressive configurations—a result that can be difficult to anticipate through manual tuning. Transfer learning cuts per-platform training to 7.5k steps after 30k base training steps. Hardware validation and off-diagonal weight learning remain for future investigation.

Acknowledgements

This research is supported by the School of Computing and Engineering, University of West London, St. Mary's Ealing Broad, London, W5 5RF, the United Kingdom.

References

- Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M., 2019. Casadi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation* 11, 1–36.
- Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., Schoellig, A.P., 2022. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems* 5, 411–444.
- Bryson, A.E., Ho, Y.C., 1975. *Applied optimal control: optimization, estimation and control*. CRC Press.
- Chen, S.W., Wang, T., Atanasov, N., Kumar, V., Morari, M., 2022. Large scale model predictive control with neural networks and primal active sets. *Automatica* 135, 109947.
- Hanover, D., Foehn, P., Sun, S., Kaufmann, E., Scaramuzza, D., 2024. Autonomous drone racing: A survey. *IEEE Transactions on Robotics* 40, 3038–3062.
- Hewing, L., Wabersich, K.P., Menner, M., Zeilinger, M.N., 2020. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems* 3, 269–296.
- Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V., Scaramuzza, D., 2023. Champion-level drone racing using deep reinforcement learning. *Nature* 620, 982–987.
- Loquercio, A., Kaufmann, E., Ranftl, R., Müller, M., Koltun, V., Scaramuzza, D., 2021. Learning high-speed flight in the wild, in: *Science Robotics*, p. eabg5810.
- Mehndiratta, M., Camci, E., Kayacan, E., 2018. Automated tuning of nonlinear model predictive controller by reinforcement learning, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 3016–3021.
- Panerati, J., Zheng, H., Zhou, S., Xu, J., Prorok, A., Schoellig, A.P., 2021. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 7512–7519.
- Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N., 2021. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research* 22, 1–8.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Song, Y., Romero, A., Müller, M., Koltun, V., Scaramuzza, D., 2023. Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. *Science Robotics* 8, eadg1462.
- Torrente, G., Kaufmann, E., Föhn, P., Scaramuzza, D., 2021. Data-driven mpc for quadrotors. *IEEE Robotics and Automation Letters* 6, 3769–3776.
- Wabersich, K.P., Zeilinger, M.N., 2022. Probabilistic model predictive safety certification for learning-based control, in: *IEEE Transactions on Automatic Control*, IEEE, pp. 176–188.
- Yuan, Z., Hall, A.W., Zhou, S., Brunke, L., Greeff, M., Panerati, J., Schoellig, A.P., 2022. Safe-control-gym: A unified benchmark for safe learning-based control and reinforcement learning in robotics, in: *IEEE Robotics and Automation Letters*, IEEE, pp. 11142–11149.