

Toward Safe and Trustworthy Aerial Robotics Through Hybrid MPC–RL

Rudolf Reiter, Angel Romero, and Davide Scaramuzza
Robotics and Perception Group, University of Zurich, Switzerland

Abstract—Safe and trustworthy autonomous flight requires control policies that are flexible enough to handle complex, dynamic environments and structured enough to offer guarantees on stability and constraint satisfaction. Model predictive control (MPC) and reinforcement learning (RL) have each achieved remarkable results in aerial robotics, yet their individual limitations motivate a combined approach. In this paper, we review the complementary strengths of MPC and RL and argue that their synthesis is a key enabler for the next generation of safe, resilient aerial systems. Concretely, we discuss the Actor-Critic Model Predictive Control (AC-MPC) framework [1], which embeds a differentiable MPC layer within an actor-critic RL architecture. We show that this integration achieves state-of-the-art performance in agile drone racing at up to 21 m/s while exhibiting superior robustness to out-of-distribution conditions compared to purely model-free policies. Drawing on a broader survey of MPC–RL combination methods [2], we place AC-MPC in a wider methodological context and identify open challenges for safe and trustworthy aerial autonomy.

Keywords: model predictive control, reinforcement learning, quadrotor, drone racing, safe autonomy

I. INTRODUCTION

Autonomous aerial robots are increasingly deployed in safety-critical applications, including infrastructure inspection, search and rescue, and precision agriculture. In all of these scenarios, a failure of the control system can have severe consequences, posing a strong demand for controllers that are not only high-performing but also verifiable and robust. At the same time, agile flight tasks, such as navigating through tight indoor environments or racing through gates at high speed, cf. Fig. 1, require policies that push the vehicle to its dynamic limits.

Two paradigms have dominated autonomous flight research over the past decade. Model Predictive Control (MPC) solves a short-horizon optimal control problem at every time step, explicitly accounting for the system dynamics and constraints [2]. This makes MPC interpretable and well-suited for safety-critical applications, but the quality of the resulting policy depends heavily on model accuracy and the manual design of the cost function. Reinforcement Learning (RL), on the other hand, learns a feedback policy directly from task-level rewards through trial and error, without the need for a hand-crafted cost function or a high-fidelity model [1]. RL has achieved superhuman lap times in drone racing [3], yet it remains difficult

This work was supported by the European Union’s Horizon Europe Research and Innovation Programme under grant agreement No. 101120732 (AUTOASSESS) and the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).



Fig. 1. Drone-racing as shown in [1]. Merging MPC and RL offers great advantages for safe AI-driven navigation.

to verify and is susceptible to degradation in out-of-distribution states.

The complementarity of these approaches is well-recognised [2], where weaknesses of one method are often strengths of the other. This has motivated a rapidly growing body of research on MPC–RL combinations that aim to simultaneously leverage the online replanning and constraint-handling capabilities of MPC together with the task-level optimisation and exploration capabilities of RL. In this paper, we review this broader landscape and focus on the Actor-Critic MPC (AC-MPC) framework [1] as a concrete instantiation validated in real-world agile flight.

II. BACKGROUND: MPC, RL, AND THEIR SYNTHESIS

A. Model Predictive Control

MPC solves, at each time step, a finite-horizon optimal control problem of the form

$$\min_{u_{0:N-1}} \sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N) \quad \text{s.t.} \quad x_{k+1} = f(x_k, u_k), \quad (1)$$

where $x_k \in \mathcal{X}$ and $u_k \in \mathcal{U}$ denote the state and control input, ℓ is a stage cost, and V_f is a terminal cost [2]. Only the first optimal control action is applied, and the optimisation is repeated at the next step [4]. This receding-horizon strategy provides implicit feedback and robustness to disturbances. A common special case is tracking MPC, where the cost penalises the deviation from a pre-planned reference trajectory. While

straightforward, this can be suboptimal when the system is driven to its limits, because the planning step may not produce a time-optimal reference in the first place [5], [6].

B. Reinforcement Learning

RL frames control as a Markov decision process (MDP) in which an agent learns a policy $\pi : \mathcal{X} \rightarrow \mathcal{U}$ by interacting with an environment and maximising cumulative rewards. Modern deep RL methods have achieved impressive results on continuous control tasks, including agile drone racing [1]. Crucially, RL optimises the task objective directly, without requiring an explicit reference trajectory, and can handle sparse and non-differentiable reward signals.

The major drawback of model-free RL is its lack of structure: the policy is a black-box neural network that provides no guarantees on constraint satisfaction and can degrade in out-of-distribution states [2]. This makes direct deployment in safety-critical settings challenging without additional safeguards.

C. Taxonomy of Combination Methods

The survey in [2] classifies MPC–RL combinations along two main axes: *how* MPC is incorporated into the RL framework and *what role* it plays. Key categories include:

- **MPC as expert actor:** An MPC controller provides demonstrations that guide the RL policy, as in guided policy search.
- **MPC within the deployed policy:** A differentiable MPC layer is embedded inside the actor network, enabling end-to-end training while retaining predictive structure [1].
- **MPC as a critic:** The value function is represented by, or learned from, an MPC-derived terminal cost, providing theoretically grounded credit assignment.
- **MPC for posterior safety:** A safety filter based on MPC is applied on top of an RL policy, projecting unsafe actions into a constraint-satisfying set.

Across all categories, the synthesis of MPC and RL consistently outperforms each standalone method [2] on many relevant control tasks, yielding better sample efficiency, stronger generalisation, and more reliable real-world deployment.

D. Comparative Properties

Table I summarises the key properties of standalone MPC, standalone model-free RL, and the AC-MPC combination along dimensions that are directly relevant to safe aerial autonomy. The entries reflect the broader analysis in [2] as well as the empirical findings from agile drone racing experiments in [1].

The table highlights that AC-MPC successfully combines the task-level optimisation strength of RL with the sample efficiency, online adaptability, and partial interpretability of MPC. The AC-MPC work in [1] can account for constraints only with quadratic penalties in the cost formulation. More recent frameworks, are able to include constraints in a differentiable MPC setting [7].

III. ACTOR-CRITIC MODEL PREDICTIVE CONTROL

A. Architecture

The AC-MPC framework [1] belongs to the “MPC within the deployed policy” category. The core idea is to replace the

TABLE I
COMPARISON OF MPC, MODEL-FREE RL, AND AC-MPC ALONG
DIMENSIONS RELEVANT TO SAFE AERIAL AUTONOMY. RATINGS: + GOOD,
O MODERATE, - POOR.

Property	MPC	RL	AC-MPC
Sample efficiency	+	-	+
Task performance	o	+	+
Out-of-distr. robustness	+	-	+
Constraint handling	+	-	o
Engineering effort	-	o	o
Online adapt.	+	-	+
Interpretability	+	-	o

output layer of the actor network with a differentiable MPC module. The actor consists of two components:

- 1) A *neural cost map* that takes the current observation o_t and outputs the cost matrices (Q_k, R_k) for the MPC problem (1).
- 2) A *differentiable MPC* that uses these cost matrices together with a model of the quadrotor dynamics to compute the optimal control action.

Because the MPC is differentiable with respect to its cost parameters, policy gradients propagate back through the MPC into the neural cost map, enabling fully end-to-end training via standard actor-critic RL without requiring expert demonstrations. At deployment time, the control commands are drawn directly from the MPC solution, which ensures that the quadrotor’s dynamics are respected over the planning horizon [1].

This design introduces a two-timescale structure: the MPC handles short-term predictive planning, while the critic manages long-horizon value estimation. This is a key advantage over vanilla actor-critic RL, where the actor is a randomly initialised feedforward network with no domain-specific structure or physics-based prior.

B. Model Predictive Value Expansion

A further contribution is Model Predictive Value Expansion (MPVE) [1]. The short-horizon state-action rollouts produced by the differentiable MPC during every forward pass are reused to generate additional training signal for the critic. This improves sample efficiency without introducing additional policy calls and is most beneficial for small values of the temporal-difference parameter λ .

C. Value Function and MPC Cost

An interesting empirical finding reported in [1] is that the Hessian of the learned value function closely matches the quadratic cost matrices generated by the neural cost map when evaluated around hover equilibria. This provides empirical evidence for a theoretical connection between the optimal value function of the MDP and the MPC cost function, which aligns with the theoretical analysis in [2] regarding conditions under which MPC and RL solve equivalent problems.

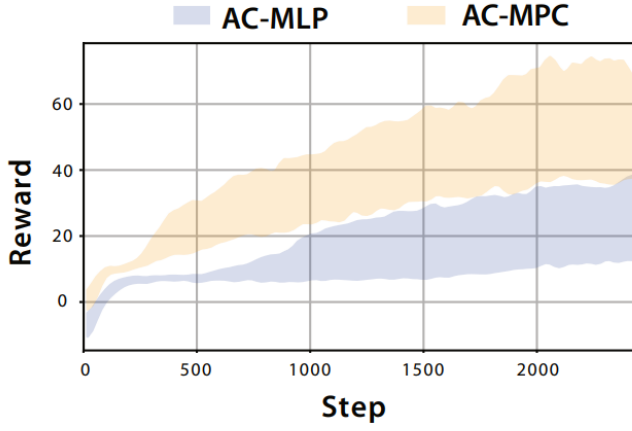


Fig. 2. Comparison of an MPC actor versus an MLP actor during the training.

IV. EXPERIMENTAL RESULTS: AGILE DRONE RACING

Drone racing is an ideal benchmark for agile flight controllers because it requires pushing the quadrotor to its physical limits while reliably passing through a sequence of gates at high speed. Small model mismatches, aerodynamic disturbances, or state estimation errors can all cause crashes, making the task both physically and algorithmically demanding [1].

Fig. 2 compares AC-MPC against the AC-MLP representative baseline. The ACMPC prior helps to much faster learn competitive racing policies. Tracking MPC achieves moderate success rates because conservative thrust limits are required to maintain tracking feasibility. Model Predictive Contouring Control (MPCC) improves upon this by directly optimising path progress, but still requires significant manual tuning per track. The model-free RL baseline (AC-MLP) achieves competitive lap times and high success rates, but is more sensitive to changes in the quadrotor’s dynamic parameters. AC-MPC matches the lap time and success rate of the RL baseline while offering substantially better robustness to out-of-distribution conditions, including variations in vehicle mass, inertia, and actuator limits [1].

The trajectory in Fig. 3 shows that AC-MPC exploits the full flight envelope: speed peaks between gates and decreases near each gate, reflecting an aggressive but controlled racing strategy. In real-world experiments, the maximum recorded speed is 21 m/s, on par with the best human racing pilots [1], [3].

Robustness to dynamic uncertainty is particularly important for deployment. In ablation experiments, AC-MPC maintained high success rates under $\pm 30\%$ changes in vehicle mass and $\pm 50\%$ changes in body-rate limits without any re-training. This resilience stems from the MPC’s access to an online-updated dynamics model: even with a fixed neural cost map, the MPC can adapt its control sequence to compensate for changed parameters within the planning horizon. This is in contrast to model-free policies, which have no mechanism for such online adaptation.

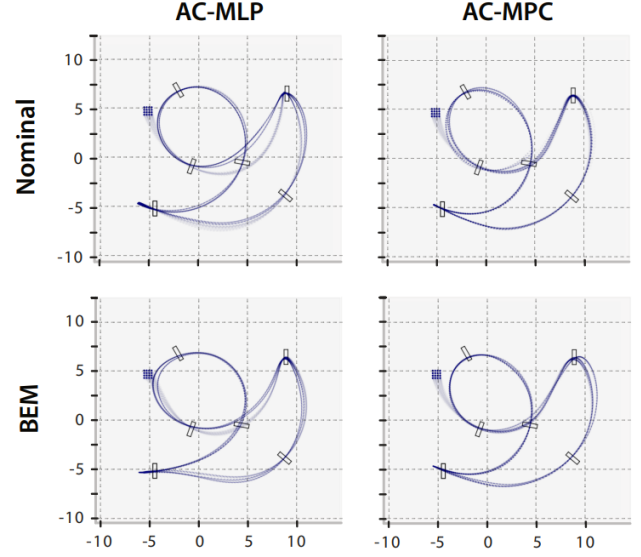


Fig. 3. Comparison of an MPC actor versus an MLP actor trajectories.

A. Real-World Validation

Demonstrating agile autonomous flight in the real world is particularly challenging because the gap between simulation and reality is large for aggressive manoeuvres at high speed. Aerodynamic effects, motor delays, and state estimation noise all introduce disturbances that are difficult to capture in simulation. Despite this, AC-MPC successfully transferred from simulation to hardware without any additional fine-tuning, achieving consistently high gate-passing success rates across multiple tracks.

The history of autonomous drone racing underscores how demanding this benchmark is. Early approaches used polynomial minimum-snap trajectories tracked by a geometric controller. Later, time-optimal planners were introduced, but their trajectories saturated the actuators so aggressively that even small disturbances caused crashes. Model Predictive Contouring Control (MPCC) then improved lap times by directly optimising path progress, but required extensive per-track tuning. Only with the advent of model-free RL did a control strategy emerge that could consistently outperform human pilots [1]. AC-MPC reaches the same performance level while retaining the interpretable, model-driven structure of MPC, which is a prerequisite for any certification pathway in safety-critical aerial applications.

V. DISCUSSION: TOWARDS SAFE AND TRUSTWORTHY AERIAL AUTONOMY

A. Hybrid AI as a Design Engine for Safety

A central theme of this workshop is the role of models in AI: as enabler of high-performance autonomy and as a potential source of unpredictability that must be managed through careful algorithm design. The MPC–RL synthesis reviewed here represents one concrete way to address this tension. By embedding a physically motivated optimisation problem within the policy, AC-MPC provides a degree of

structural interpretability that purely data-driven policies lack. The differentiable MPC layer acts as a structured prior that constrains the policy’s actions to physically feasible trajectories, reducing the risk of unsafe behaviour outside the training distribution.

From the broader perspective of [2], the field has converged on several strategies for infusing safety into learned policies: formal constraint satisfaction via MPC-based safety filters, terminal value functions derived from Lyapunov analysis, and architectures that separate learned task objectives from model-based stability guarantees. Each strategy offers a different trade-off between flexibility, performance, and the strength of the safety guarantee.

B. Open Challenges

Several challenges remain before MPC–RL systems can be deployed at scale in safety-critical aerial applications.

Verifiability. The neural cost map in AC-MPC is still a black-box component whose outputs are difficult to certify formally. Integrating verification methods – such as interval arithmetic or Lipschitz-based bounds – with differentiable MPC architectures is an important open direction [2].

Model accuracy. The MPC component relies on a nominal rigid-body dynamics model. While domain randomisation helps to bridge the sim-to-real gap, significant aerodynamic effects are not easily captured in simple models. Online model adaptation, as discussed in the aligned MPC-RL model structure [2], is a promising direction.

Long-horizon planning. The short MPC horizon is intentionally limited for tractability. For tasks requiring long-horizon reasoning – such as exploration or multi-vehicle coordination – hierarchical architectures that pair a long-horizon RL planner with a short-horizon MPC executor represent a promising avenue.

Ethical and regulatory aspects. As AI-driven aerial systems enter public airspace, accountability, explainability, and regulatory compliance become pressing concerns. The partial interpretability of MPC-based architectures – where an operator can inspect the neural cost function – is a step in the right direction. Formalising this into aviation certification frameworks remains an important and open problem.

VI. CONCLUSION

We have argued that the synthesis of Model Predictive Control and Reinforcement Learning is a key enabler for safe, resilient, and trustworthy aerial autonomy. The AC-MPC framework [1] demonstrates that embedding a differentiable MPC within an actor-critic RL architecture achieves superhuman performance in agile drone racing while providing superior robustness to out-of-distribution conditions compared to purely model-free policies. Grounded in a broader landscape of MPC–RL combination methods introduced by [2], we identified open challenges related to verifiability, model accuracy, long-horizon planning, and regulatory compliance.

We hope that this work will stimulate interdisciplinary collaboration between the optimal control, machine learning,

and aerial robotics communities – precisely the kind of research needed to build aerial systems that are not only fast and capable, but also safe and worthy of public trust.

ACKNOWLEDGEMENTS

This work was supported by the European Union’s Horizon Europe Research and Innovation Programme under grant agreement No. 101120732 (AUTOASSESS) and the European Research Council (ERC) under grant agreement No. 864042 (AGILEFLIGHT).

REFERENCES

- [1] A. Romero, E. Aljalbout, Y. Song, and D. Scaramuzza, “Actor-critic model predictive control: Differentiable optimization meets reinforcement learning for agile flight,” *IEEE Transactions on Robotics*, 2025. To appear. Source code: https://github.com/uzh-rpg/acmpc_public.
- [2] R. Reiter, J. Hoffmann, D. Reinhardt, F. Messerer, K. Baumgärtner, S. Sawant, J. Bödecker, M. Diehl, and S. Gros, “Synthesis of model predictive control and reinforcement learning: Survey and classification,” *Annual Reviews in Control*, vol. 61, p. 101045, 2026.
- [3] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, “Champion-level drone racing using deep reinforcement learning,” *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.
- [4] M. Diehl, H. G. Bock, and J. P. Schlöder, “A real-time iteration scheme for nonlinear optimization in optimal feedback control,” *SIAM Journal on Control and Optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.
- [5] A. Romero, R. Penicka, and D. Scaramuzza, “Time-optimal online replanning for agile quadrotor flight,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 7730–7737, July 2022.
- [6] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, “Model predictive contouring control for time-optimal quadrotor flight,” *IEEE Transactions on Robotics*, vol. PP, pp. 1–17, 12 2022.
- [7] L. Fichtner, D. Reinhardt, J. Hoffmann, F. Airalidi, J. Frey, J. K. Hromatko, K. Baumgaertner, M. Amria, R. Reiter, and S. Sawant, “leap-c/leap-c: v0.1.0-alpha,” Oct. 2025.