Decentralized Aerial Manipulation of a Cable-Suspended Load Using Multi-Agent Reinforcement Learning

Jack Zeng[1], Andreu Matoses Gimenez[1], Eugene Vinitsky[2], Javier Alonso-Mora[1], Sihao Sun[1]
[1]Delft University of Technology, [2]NYU Tandon School of Engineering

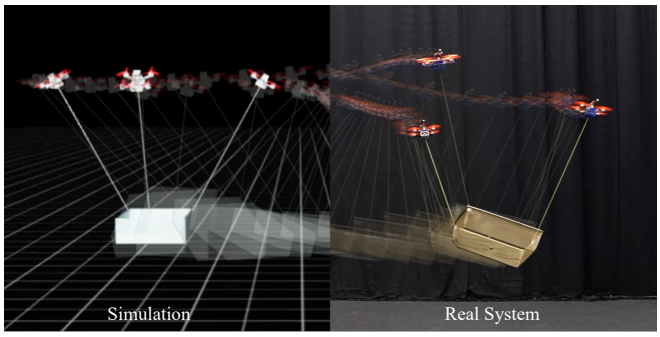

Fig. 1: Multi-MAV lifting system performing full-pose control of a cable-suspended load. Left: simulation environment used to train the decentralized outer-loop control policy. Right: policy transferred to the real system. We refer readers to the following link for the full paper, the methodology, and the video of the real-world experiments: `https://autonomousrobots.nl/paper_websites/aerial-manipulation-marl`

Autonomous Micro Aerial Vehicles (MAVs) offer great capability for transporting slung loads to dangerous and remote locations [1]. While a single low-cost MAV has limited payload capacity, collaborative teams of MAVs can transport significantly heavier loads. In addition, by connecting each MAV with the load at different points using tethers, the full pose of the load can be controlled by changing the position of the MAVs, yielding a cooperative cable-suspended manipulation solution, which shows great potential for aerial-based construction, inspection, and resecuring [2]–[6].

To coordinate and control MAV fleets, the state-of-the-art method [6] employs a centralized framework that accurately captures the strong dynamical coupling between the MAVs and the suspended load. This ensures safety and stability while addressing the significant underactuation inherent to cable-suspended systems, preventing actuator saturations and reciprocal collisions. However, using centralized control strategies for such systems suffers from critical drawbacks: computational complexity tends to scale exponentially with the number of agents for many approaches, rendering real-time control infeasible for larger teams with a centralized scheme [6], [7]. In addition, dependence on global state information and centralized communication is often impractical due to limits on sensors and communication bandwidth. A plausible solution, decentralization, remains an open challenge to effectively coordinate MAV fleets due to partial observability, limited communication bandwidth, and decision-making under strong dynamical coupling between agents while co-manipulating an object.

In this work, we present the first decentralized algorithm to achieve a real-world demonstrated full-pose manipulation of a cable-suspended payload using a team of MAVs. Our method leverages multi-agent reinforcement learning (MARL) and **does not require any inter-agent communication**. Instead, each agent only takes their own state and identity, the load pose, and the target load pose as observations. We train the policy through MARL in a centralized training with decentralized execution (CTDE) paradigm using multi-agent proximal policy optimization (MAPPO) [8]. Each MAV learns to communicate implicitly through the load pose information. To fill the sim-to-real gap in this highly dynamic cooperative task, we design the action space of the reinforcement learning (RL) policy as reference linear accelerations and body rates of the MAV and combine the RL policy with a low-level controller based on incremental nonlinear dynamic inversion (INDI) [9]–[11]. The low-level controller follows the linear acceleration command with the body rate reference as the feedforward commands, ensuring agile and smooth control maneuvers during the cooperative manipulation.

Our method enables zero-shot transfer of the policy from simulation to real-world deployment to achieve full-pose control accuracy comparable to the state-of-the-art centralized controller [6], and is deployed fully onboard. In addition, experiments with real MAVs demonstrate that our method remains robust under load model uncertainties, operates effectively in heterogeneous agent settings where one MAV uses a different controller, and remains functional even when one of the MAVs completely fails. **Our core contributions are as follows:**

- The first method to achieve fully decentralized and onboard-deployed cooperative aerial manipulation in experiments with real MAVs, without any inter-agent communication.
- A novel action space design for MAVs manipulating a cable-suspended load, together with a robust low-level controller, enabling successful zero-shot sim-to-real transfer.
- First demonstration of robust full-pose control of the cable-suspended load under heterogeneous conditions and even under complete in-flight failure of an MAV.

I. Real-world experiment results

Setpoint tracking We demonstrate agile pose control of three MAVs with a cable-suspended load, tracking a 2m displacement with (30°, -20°, -90°) commands. Figure 3 compares our decentralized method with centralized NMPC [6]. Despite decentralization, our method achieves similar performance with position and attitude RMSEs of 0.52m (vs. 0.45m) and 22.93° (vs. 16.24°). Since NMPC tracks trajectories while we track poses, its RMSE is smaller during transients. Time-to-target (error $< 0.10\,\mathrm{m}/10°$) is 6.84,s for NMPC vs. 8.36,s for ours, with final RMSEs of 0.05m/4.02° vs. 0.04m/5.78°. With 4 MAVs (no cable slack), we achieve RMSEs of 0.92 m and 42.67°; the higher error likely arises from overconstraint and complex cable dynamics [12]. For computation, both methods run onboard a Raspberry Pi5 (2.4GHz ARM Cortex-A76). Our inference takes **6ms** at 100Hz vs. NMPC's **78ms** at 10Hz. Crucially, NMPC scales poorly (174ms/267ms for 5/6 agents), while ours remains **constant with team size**.

Robustness to load mismatch We attach 0.216kg objects (15.4% load mass), including freely moving items perturbing inertia. Despite no randomization during training, tracking remains strong (0.63 m vs. 0.60 m; 26.93° vs. 26.49° RMSE). Low-level feedback compensates disturbances, showing robustness to model uncertainties (Figure2B).

Heterogeneous agents Though trained for homogeneous MAVs, our policy works with heterogeneous ones. In a hover test, one MAV used a model-based controller [11] with shifted setpoints: first pulling the load 0.7 m outward, then pushing it 0.3 m inward. As shown in Figure 2A, the RL-controlled MAVs compensate for load deviations since the policy depends only on load pose, unlike a fully observable policy, which fails under these conditions.

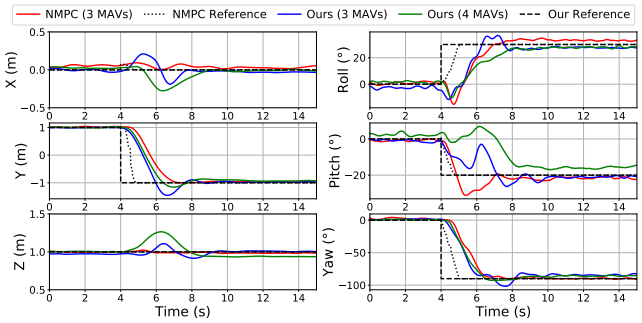

Fig. 3: Time series of pose tracking results comparing our method and a centralized NMPC method [6]. Our method also includes a setup with 4 MAVs.

In-flight failure Combining heterogeneity and robustness also enables fault tolerance. We turned off one MAV, leaving two to control the load. Although orientation about their connecting line is unactuated and the failed MAV hangs as a disturbance, our method maintains 5 DoF control: yawing -180°, descending 0.5 m, and maneuvering 1 m laterally. Results appear in Figure 2C. As in the heterogeneous case, independence from agent states prevents instability, unlike the fully observable policy which fails.

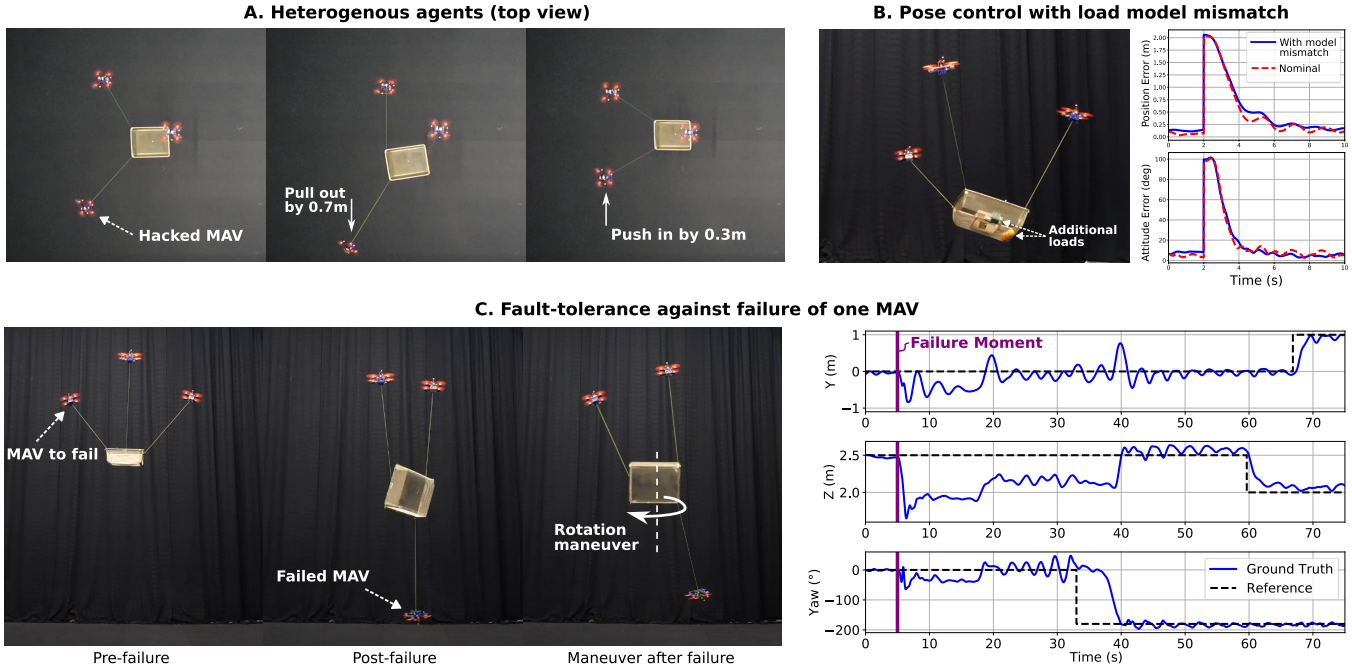

Fig. 2: Real-world experiments. (A) Snapshot of the test with heterogeneous agents in which one MAV is manually controlled (hacked) to pull out and push in, and the other two MAVs counteract the interference of the hacked MAV. (B) Snapshot of the test where additional load is added to the original load, and the pose error with and without such model mismatch. (C) Snapshot of the case where one MAV fails in flight and the remaining two MAVs manage to control the load.

II. Ablation Studies

We compare our selected observation and action spaces with alternatives in simulation for safety. The Agilicious flight stack is used with the Gazebo simulator [13] and RotorS [14] plugins, which add sensor noise, aerodynamic disturbances, and system latencies in a ROS environment. All policies are trained for 1 billion environment steps (10 h) and evaluated 10 times in Gazebo.

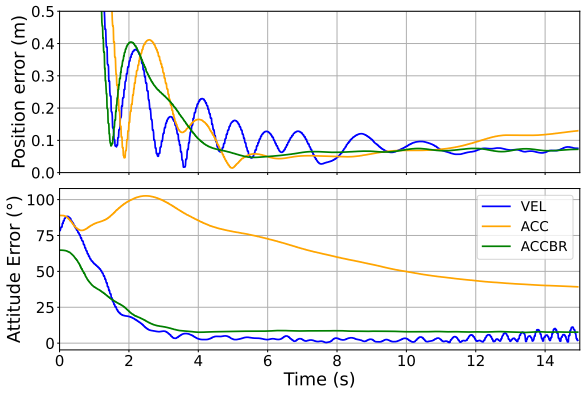

Fig. 4: Positional and attitude errors comparing different action spaces at test time in the Gazebo environment.

Action space	Pos RMSE	Att RMSE
ACCBR	0.64 ± 0.00	33.87 ± 0.91
CTBR*	NaN	NaN
ACC	**0.54 ± 0.00**	87.89 ± 1.85
VEL	0.56 ± 0.06	**25.74 ± 1.49**

*Not able to take off

TABLE I: Pose tracking RMSEs of different action spaces at test time in the Gazebo environment.

Action space comparison We compare ACCBR with three action spaces: velocity (VEL), linear acceleration (ACC), and collective thrust with body rates (CTBR). ACCBR, VEL, and ACC share the same low-level controllers that compensate for disturbances, while CTBR feeds directly into the INDI controller without compensation.

As shown in Table I, VEL performs best, followed by ACCBR, while ACC fails to track load orientation. Notably, the widely used CTBR [15], [16] fails to learn, Since CTBR directly commands collective thrust without leveraging the proposed low-level controller's disturbance compensation, making unpredictable cable forces too difficult to handle without force sensing.

However, while VEL achieves lower RMSE, Figure 4 shows it causes **hazardous oscillations**. ACCBR offers more stable hovering despite higher initial errors, making it safer and preferable for stability-critical tasks like inspection or delivery.

Observation space To benchmark the decentralized policy's performance, we compare three observation space cases: (1) the fully observable case with global state $\boldsymbol{s} = [\boldsymbol{x}_L, \boldsymbol{x}_G, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3]$, (2) an augmented partial observability case where each MAV i also receives the load twist and other MAVs' positions ("Partial augmented") $\boldsymbol{o}_i = [\boldsymbol{x}_L, \boldsymbol{x}_G, \boldsymbol{p}_{j_1}, \boldsymbol{p}_{j_2}, \boldsymbol{x}_i, \boldsymbol{e}_i]$ with $\boldsymbol{p}_{j_1}, \boldsymbol{p}_{j_2}$ representing the neighboring agents' positions, and (3) the partially observable case. For partially observable cases, we include observation histories ($H = 3$) to improve state estimation and decision-making under uncertainty [17]. Figure 5 reveals comparable convergence across all configurations, indicating that load pose alone serves as a sufficient statistic for implicit MAV coordination, while the full global state contains redundant elements.

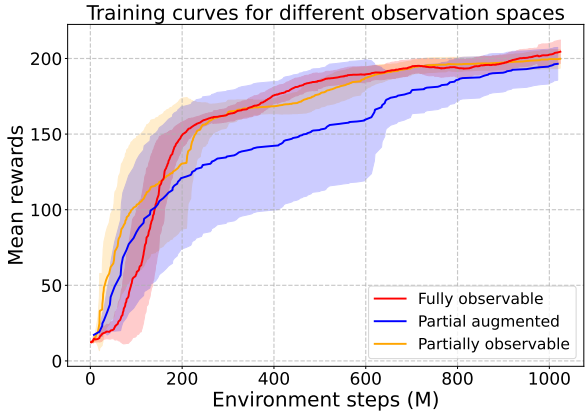

Fig. 5: Training curves of fully observable, partial augmented, and partially observable observation spaces.

III. Method Overview

Our method (Figure 6) uses MARL to train an outer-loop policy that generates reference accelerations and body rates from local ego-MAV observations, robot ID, payload, and goal pose. A low-level INDI controller tracks these references. During training, a centralized critic uses the full state, but at execution only local observations are used. Experience is shared across actors to learn a single policy, enabling centralized training and decentralized, onboard execution with zero-shot sim-to-real transfer.

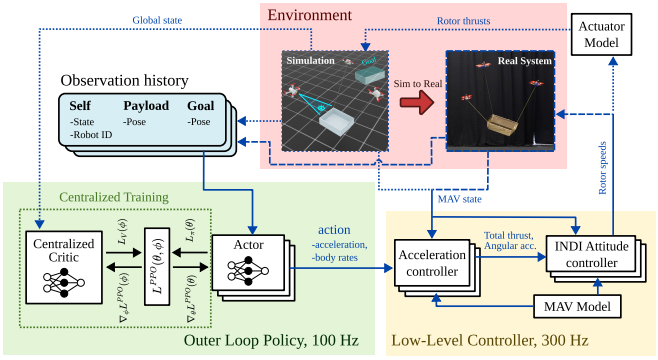

Fig. 6: Overview of our method: dotted lines show training-only components, dashed lines deployment-only, and solid lines both. We use MARL to train a shared outer-loop policy that generates acceleration and body rate references from local observations, tracked by an INDI controller.

References

[1] E. N. Barmpounakis, E. I. Vlahogianni, and J. C. Golias, "Unmanned aerial aircraft systems for transportation engineering: Current practice and future challenges," *International Journal of Transportation Science and Technology*, vol. 5, no. 3, pp. 111–122, 2016.

[2] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," *rn*, vol. 1, no. r2, r3, 2013.

[3] T. Lee, "Geometric control of quadrotor uavs transporting a cable-suspended rigid body," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 255–264, 2017.

[4] J. Geng, P. Singla, and J. W. Langelaan, "Load-distribution-based trajectory planning and control for a multilift system," *Journal of Aerospace Information Systems*, vol. 19, no. 5, pp. 366–381, 2022.

[5] G. Li and G. Loianno, "Nonlinear model predictive control for cooperative transportation and manipulation of cable suspended payloads with multiple quadrotors," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 5034–5041.

[6] S. Sun, X. Wang, D. Sanalitro, A. Franchi, M. Tognon, and J. Alonso-Mora, "Agile and cooperative aerial manipulation of a cable-suspended load," *arXiv preprint arXiv:2501.18802*, 2025.

[7] L. Bakule and M. Papik, "Decentralized control and communication," *Annual Reviews in Control*, vol. 36, no. 1, pp. 1–10, 2012.

[8] C. Yu, A. Velu, E. Vinitsky, *et al.*, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in neural information processing systems*, vol. 35, pp. 24 611–24 624, 2022.

[9] E. J. Smeur, Q. Chu, and G. C. De Croon, "Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 450–461, 2016.

[10] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, 2020.

[11] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, 2022.

[12] G. Li, X. Liu, and G. Loianno, "Rotortm: A flexible simulator for aerial transportation and manipulation," *IEEE Transactions on Robotics*, vol. 40, pp. 831–850, 2023.

[13] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, IEEE, 2004, pp. 2149–2154.

[14] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "Rotors—a modular gazebo mav simulator framework," *Robot Operating System (ROS) The Complete Reference (Volume 1)*, pp. 595–625, 2016.

[15] 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.

[16] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Science Robotics*, vol. 8, no. 82, eadg1462, 2023.

[17] M. J. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps.," in *AAAI fall symposia*, vol. 45, 2015, p. 141.