Multi-MIGHTY: Decentralized Multiagent Trajectory Planning with Hermite Splines

Kota Kondo[1], Juan Rached[1], Blake Edwards[1], Mason B. Peterson[1], Lucas Jia[1], Lili Sun[1], Shanti Visurakapalli[1], Joel Abu[1], Janet Weng[1], Yuwei Wu[2], Vijay Kumar[2], and Jonathan P. How[1]

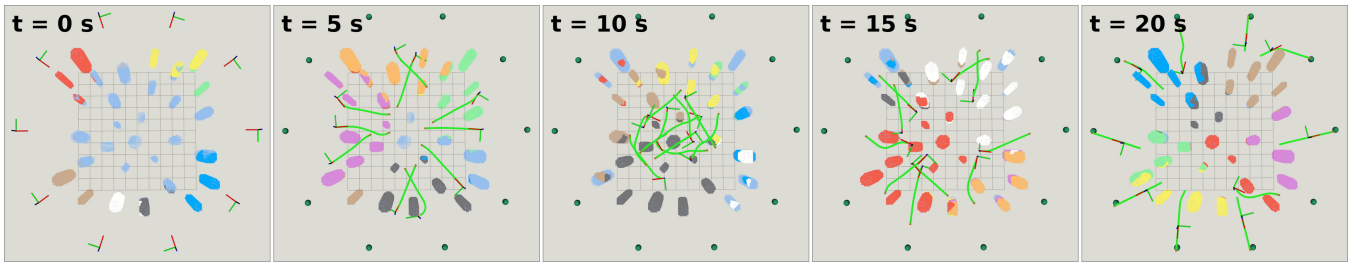


Fig. 1. 10-agent position-exchange simulation. Agents start on a circle ($t = 0\,\mathrm{s}$), fly through the congested center while avoiding each other's broadcast committed trajectories (green curves), and reach their goals (green spheres) ($t = 20\,\mathrm{s}$).

***Abstract*— Teams of robots operating in a shared, confined environment must generate deconflicted, feasible trajectories onboard and in real time. We present Multi-MIGHTY, a decentralized multiagent extension of MIGHTY, a Hermite spline-based planner that performs joint spatiotemporal optimization while fully leveraging the continuous search space of a spline. Each agent replans asynchronously onboard, broadcasts its committed trajectory, and treats teammates' committed trajectories as moving obstacles through a smooth inter-agent avoidance penalty, so the single-agent planner's efficiency and trajectory quality transfer directly to the team. The resulting open-source MIGHTY system spans single- and multiagent operation, aerial and ground platforms, position-command/exchange and exploration tasks, and both simulation and hardware: we demonstrate a 10-agent simulated position exchange, hardware position exchanges with 4 UAVs and 5 ground rovers, and 3-robot indoor explorations in simulation and hardware. The system operated robustly across all of these settings, reliably deconflicting the agents, and in hardware all perception, planning, and control run onboard in real time.**

Supplementary Material

Code: https://github.com/mit-acl/mighty.git

I. Introduction

Emerging robotics applications such as cooperative inspection, search and rescue, exploration, and transport increasingly require *teams* of robots, aerial and ground alike, to operate in close proximity within a shared environment. In these tightly coupled settings, each robot's motion directly constrains its teammates', and a central challenge is scaling single-robot planning intelligence to the team: each robot must replan onboard at high frequency while remaining deconflicted with every other agent.

This work was supported in part by DSTA and ARL grant W911NF-21-2-0150.

[1]Massachusetts Institute of Technology (kkondo@mit.edu).
[2]University of Pennsylvania.

Single-agent trajectory planning. Hard-constraint planners [1], [2] explicitly enforce safety but require computationally expensive solvers. Soft-constraint planners such as EGO-Planner [3], RAPTOR [4], and MINCO-based methods [5], [6] converge quickly, but either decouple spatial and temporal optimization or restrict the search to a low-dimensional trajectory class. Our recent planner **MIGHTY** [7] addresses both limitations with a quintic Hermite-spline parameterization in which knot positions, velocities, accelerations, and segment durations are all explicit decision variables, enabling joint spatiotemporal optimization over the full continuous search space of a spline with direct local control of higher-order dynamics. MIGHTY reduces computation time by 9.3% and travel time by 13.1% over state-of-the-art baselines with a 100% success rate, and completes hardware flights at speeds up to $6.7\,\mathrm{m/s}$ [7].

Multiagent trajectory planning. Decentralized, asynchronous replanning with trajectory sharing has emerged as a practical and scalable coordination paradigm. MADER [8] introduced asynchronous decentralized planning in dynamic environments with an optimization–check–recheck deconfliction scheme; Robust MADER [9] added a delay-check step that guarantees safety under communication delays; and PUMA [10] removed the shared-global-frame assumption via uncertainty-aware planning and online frame alignment. EGO-Swarm [11] and EGO-Swarm2 [12] scale soft-constraint B-spline and MINCO planners to swarms with the same lightweight trajectory-broadcast principle. While this line of work has focused on aerial swarms, the same deconfliction problem arises for ground-robot teams navigating confined indoor environments, and these systems inherit both the strengths and the representation limitations of their underlying single-agent planners.

Contributions. This extended abstract presents **Multi-**

MIGHTY, which brings Hermite-spline joint spatiotemporal optimization to multiagent teams:

1) **Multi-MIGHTY**: a decentralized, asynchronous multiagent planner in which every agent runs MIGHTY onboard and deconflicts through a smooth inter-agent penalty on broadcast committed trajectories, adding no new solver machinery (Sec. II).
2) **Position exchange**: robust coordination of 10 agents in simulation, and of 4 unmanned aerial vehicles (UAVs) and 5 ground rovers in hardware (Sec. III-A).
3) **Collaborative exploration**: indoor exploration with 3 ground robots in simulation and hardware, with all perception, planning, and control onboard (Sec. III-B).

Multi-MIGHTY thus provides an open-source planning system that spans **single- and multiagent** operation, **aerial and ground** platforms, **position-command/exchange and exploration** tasks, and both **simulation and hardware**.

II. Multi-MIGHTY

A. Background: The MIGHTY Planner

MIGHTY [7] parameterizes a trajectory as a degree-5 Hermite spline with M segments. Segment $s \in \{0, \dots, M-1\}$ has duration $T_s > 0$ and is defined on a normalized time $\tau_s = (t - t_s)/T_s \in [0, 1]$, where t is time and $t_s = \sum_{r=0}^{s-1} T_r$ is the start time of segment s:

$$\mathbf{x}_s(\tau_s) = \sum_{k=0}^{5} h_k(\tau_s)\, \mathbf{H}_{s,k}, \tag{1}$$

where $\mathbf{x}_s(\tau_s) \in \mathbb{R}^3$ is the position along segment s, h_k are the quintic Hermite basis functions, and the coefficients $\mathbf{H}_{s,k} \in \mathbb{R}^3$ collect the knot positions, velocities, and accelerations $(\mathbf{p}_i, \mathbf{v}_i, \mathbf{a}_i)$, $i = 0, \dots, M$, scaled by T_s [7]. Because adjacent segments share knot values, the spline is $\mathcal{C}^2$ continuous by construction, and every knot state is an explicit, locally controllable decision variable. MIGHTY jointly optimizes all interior knot states and segment durations, $\mathbf{z} = \left[\mathbf{p}_1^\top, \mathbf{v}_1^\top, \mathbf{a}_1^\top, \dots, T_0, \dots, T_{M-1}\right]^\top$, with fixed boundary states, using diffeomorphic reparameterizations to keep the problem unconstrained (e.g., $T_s = e^{\sigma_s} > 0$ with unconstrained $\sigma_s \in \mathbb{R}$) and duration-scaled derivative variables for numerical conditioning. The objective

$$\begin{aligned} J = w_T \sum_s T_s + w_{\text{smooth}} \tilde{J}_{\text{smooth}} + w_{\text{SFC}} J_{\text{SFC}} \\ + w_v J_v + w_a J_a + w_j J_j \end{aligned} \tag{2}$$

is a weighted sum of cost terms: the first term penalizes the total trajectory time, $\tilde{J}_{\text{smooth}}$ is the integrated squared jerk, J_{SFC} penalizes leaving the safe flight corridor (SFC), and J_v, J_a, and J_j penalize violations of the velocity, acceleration, and jerk limits, respectively; w_T, w_{smooth}, w_{SFC}, w_v, w_a, and w_j are the corresponding scalar weights. Costs are evaluated efficiently in the Bézier basis and their gradients are pulled back to the Hermite variables in closed form, so the resulting unconstrained problem is solved with L-BFGS [13] in milliseconds; dynamic obstacles are handled by an additional sampled penalty on the predicted obstacle motion [7].

B. Decentralized Multiagent Extension

Multi-MIGHTY scales this planner to a team of N agents without a central coordinator. Each agent runs its own MIGHTY instance onboard and replans continuously. Whenever an agent commits to a new trajectory, it immediately broadcasts that trajectory to the team. The message is compact: the Hermite knot states, segment durations, and trajectory start time fully determine Eq. (1), so teammates can evaluate the sender's future position at any query time.

Received committed trajectories enter the optimization as time-parameterized moving obstacles. Let $\mathbf{k}_j(t)$ denote the position at time t of the most recently received committed trajectory of teammate j; since every agent tracks its committed trajectory, $\mathbf{k}_j(t)$ is a prediction of teammate j's future motion. Mirroring the dynamic-obstacle cost of [7], agent i augments Eq. (2) with the inter-agent avoidance penalty

$$J_{\text{agent}} = w_{\text{agent}} \sum_{j \neq i} \int_0^{T_{\text{tot}}} \left(C_{\text{agent}}^2 - \|\mathbf{x}(t) - \mathbf{k}_j(t)\|^2\right)_+^3 dt, \tag{3}$$

where w_{agent} is a scalar weight, $\mathbf{x}(t)$ is the position of agent i's own candidate trajectory at time t, evaluated via Eq. (1), $T_{\text{tot}} = \sum_s T_s$ is the total trajectory duration, and $(\cdot)_+ = \max(0, \cdot)$. The constant C_{agent} is a soft activation radius: the penalty is nonzero only when the predicted distance to a teammate falls below C_{agent}, which provides the desired separation margin. Like MIGHTY's other integral costs, Eq. (3) is evaluated numerically. The candidate trajectory is sampled at a fixed set of time instants, and the integral is approximated by trapezoidal quadrature over those samples. MIGHTY already provides closed-form gradients for any cost defined on such trajectory samples [7], so a teammate's committed trajectory is handled exactly like a dynamic obstacle in the single-agent planner.

Replanning is fully asynchronous, and Multi-MIGHTY omits explicit deconfliction rounds such as the optimization–check–recheck scheme of MADER [8] or the delay check of Robust MADER [9]. Safety is instead maintained through high-frequency replanning, fast trajectory broadcast, and the separation margin built into C_{agent}. As Sec. III shows, this lightweight scheme provided robust separation across our simulation and hardware experiments.

III. Results

We validate the MIGHTY system along four axes: **agents** (single [7] and multiagent), **platform** (aerial and ground robots), **task** (position command/exchange and collaborative exploration), and **setting** (simulation and hardware). **The system operated robustly across every combination of these axes**, reliably deconflicting the agents, and in all hardware experiments perception, planning, and control run fully onboard in real time.

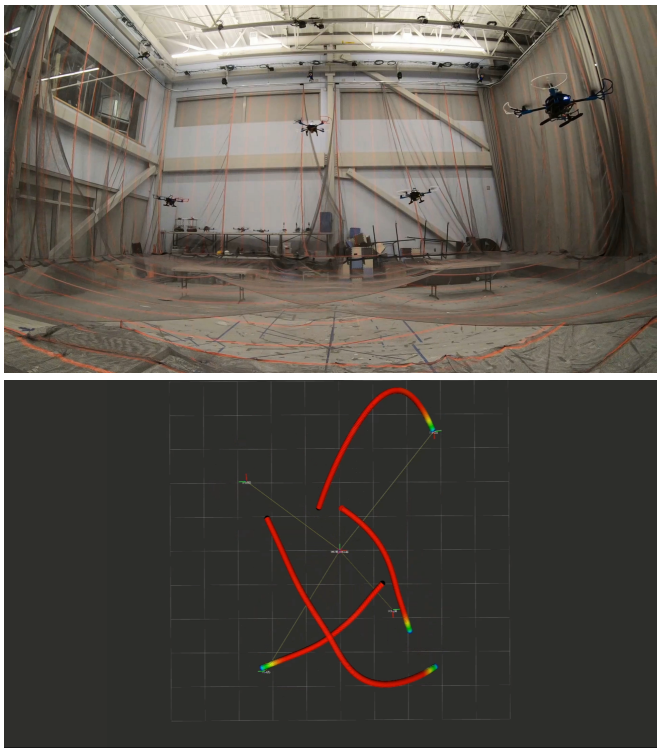


Fig. 2. Four-agent experiment with MIGHTY as the onboard planner, showing the real-world deployment (top) and the RViz visualization of the planned trajectories and occupancy maps for all agents (bottom).

A. Position Exchange

Simulation (10 agents): Ten simulated UAVs start on a circle and exchange antipodal positions, forcing all trajectories through the congested center of the environment (Fig. 1). Each agent runs as an independent process with its own planner instance and broadcasts its committed trajectory as it replans. All agents smoothly yield to one another through the center and reliably reach their goals.

Hardware (4 UAVs): Each vehicle carries a Livox Mid-360 LiDAR with DLIO [14] for localization and an Intel NUC 13 that runs perception and planning, with PX4 [15] on a Pixhawk flight controller [16] for low-level control; committed trajectories are exchanged over a wireless network. Four UAVs repeatedly exchange positions across the shared flight space (Fig. 2), robustly deconflicting throughout the flights.

Hardware (5 ground rovers): We repeat the position-exchange task with five wheeled rovers (Fig. 3): the rovers start on a circle and exchange positions, crossing the congested center simultaneously; Fig. 4 shows the corresponding RViz views. Because the rovers are nonholonomic differential-drive platforms, each ground robot tracks its committed MIGHTY trajectory with an onboard model predictive controller (MPC): a receding-horizon problem over unicycle dynamics that follows the trajectory's position, heading, and velocity references subject to velocity and acceleration bounds, and outputs velocity commands. The team robustly deconflicts across repeated runs, confirming that the same planner and deconfliction scheme transfer directly from aerial to ground platforms.



Fig. 3. Five-rover position-exchange hardware experiment (snapshots 5 s apart). The rovers start on a circle, cross the congested center while avoiding each other's broadcast committed trajectories, and reach their goals.

B. Collaborative Exploration

Beyond point-to-point tasks, we deploy Multi-MIGHTY on ground robots collaboratively exploring indoor environments, in both simulation and hardware. The robots build occupancy maps onboard, plan toward goals within a designated exploration area, and deconflict via broadcast committed trajectories.

Fig. 5 shows three ground robots exploring an indoor office environment. Using their onboard perception, the robots avoid the environment and each other while covering the exploration area.

IV. Conclusion

We presented Multi-MIGHTY, a decentralized multiagent extension of the Hermite spline-based planner MIGHTY in which agents deconflict by broadcasting committed trajectories and penalizing proximity to teammates within the same fast, unconstrained spatiotemporal optimization. Together

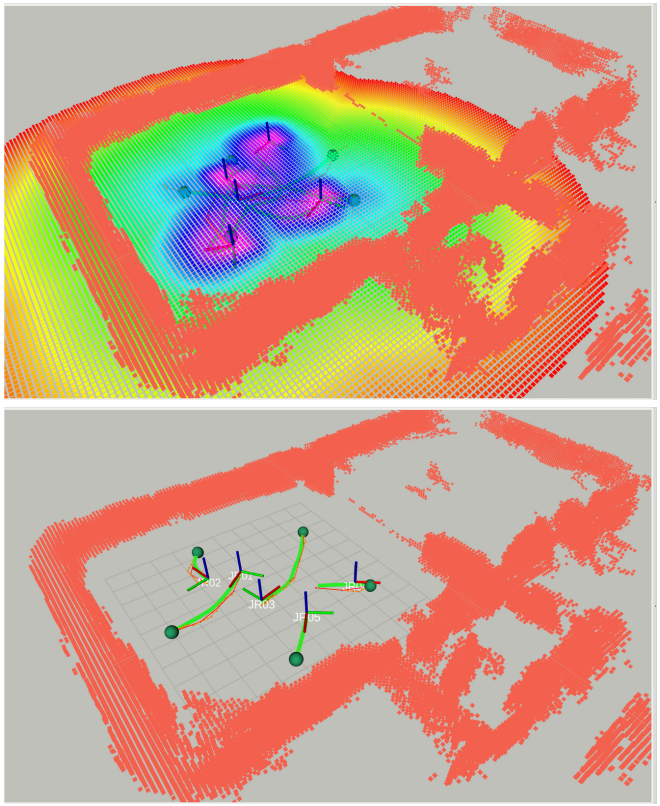


Fig. 4. RViz visualization of the five-rover position-exchange experiment: heatmap of the planning cost field over the environment (top), and the rovers' committed trajectories (green) toward their goals (green spheres) within the onboard occupancy map (bottom).

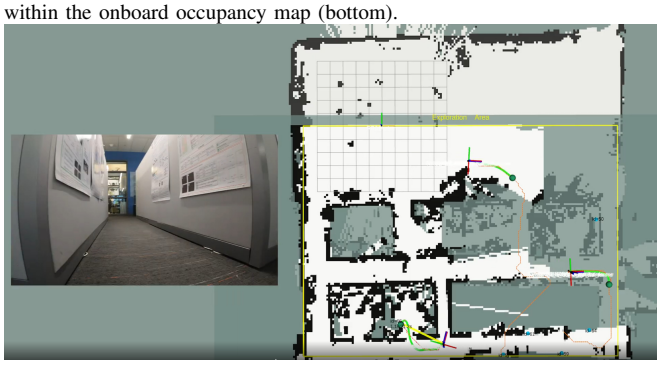


Fig. 5. Three-agent multi-agent MIGHTY experiment with ground robots in an indoor environment. The left panel shows one robot's onboard camera view, and the right panel shows the RViz visualization with the occupancy map, the exploration area (yellow), and the agents' trajectories.

with the single-agent planner [7], the open-source MIGHTY system was validated across single- and multiagent operation, aerial and ground platforms, position-command/exchange and collaborative-exploration tasks, and both simulation and hardware. These experiments, including a 10-agent simulated position exchange, hardware position exchanges with 4 UAVs and 5 ground rovers, and 3-robot indoor explorations, ran robustly with all sensing, planning, and control onboard.

REFERENCES

[1] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *RA-L*, 2017.

[2] K. Kondo, M. Peterson, N. Rober, J. R. Viso, L. Jia, J. Chen, H. Merton, and J. P. How, "Dynus: Uncertainty-aware trajectory planner in dynamic unknown environments," *ArXiv*, 2025.

[3] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "EGO-Planner: An ESDF-Free Gradient-Based Local Planner for Quadrotors," *IEEE RA-L*, 2021.

[4] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *T-RO*, 2021.

[5] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE TR-O*, 2022.

[6] Y. Ren, F. Zhu, G. Lu, Y. Cai, L. Yin, F. Kong, J. Lin, N. Chen, and F. Zhang, "Safety-assured high-speed navigation for mavs," *Science Robotics*, 2025.

[7] K. Kondo, Y. Wu, V. Kumar, and J. P. How, "MIGHTY: Hermite spline-based efficient trajectory planning," *IEEE Robotics and Automation Letters*, 2026. Accepted.

[8] J. Tordesillas and J. P. How, "MADER: Trajectory planner in multiagent and dynamic environments," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 463–476, 2022.

[9] K. Kondo, J. Tordesillas, R. Figueroa, J. Rached, J. Merkel, P. C. Lusk, and J. P. How, "Robust MADER: Decentralized and asynchronous multiagent trajectory planner robust to communication delay," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1687–1693, 2023.

[10] K. Kondo, C. T. Tewari, M. B. Peterson, A. Thomas, J. Kinnari, A. Tagliabue, and J. P. How, "PUMA: Fully decentralized uncertainty-aware multiagent trajectory planner with real-time image segmentation-based frame alignment," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[11] X. Zhou, J. Zhu, H. Zhou, C. Xu, and F. Gao, "Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments," in *IEEE ICRA*, 2021.

[12] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, 2022.

[13] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, 1989.

[14] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," in *IEEE ICRA*, 2023.

[15] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *IEEE ICRA*, 2015.

[16] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," in *IEEE ICRA*, 2011.