# Safe Navigation of Quadrotor Teams to Labeled Goals in Limited Workspaces

Sarah Tang, Justin Thomas, and Vijay Kumar

GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA
{sytang,jut,kumar}@seas.upenn.edu

**Abstract.** In this work, we solve the *labeled multi-robot planning problem*. Most proposed algorithms to date have modeled robots as kinematic or kinodynamic agents in planar environments, making them impractical for real-world systems. Here, we present experiments to validate a centralized multi-robot planning and trajectory generation method that explicitly accounts for robots with higher-order dynamics. First, we demonstrate successful execution of solution trajectories. Next, we verify the robustness of the robots' trajectory tracking to unmodeled external disturbances, in particular, the aerodynamic interactions between co-planar neighbors. Finally, we apply our algorithm to navigating quadrotors away from the downwash of their neighbors to improve safety in three-dimensional workspaces.

**Keywords:** aerial robotics, trajectory generation, multi-robot planning

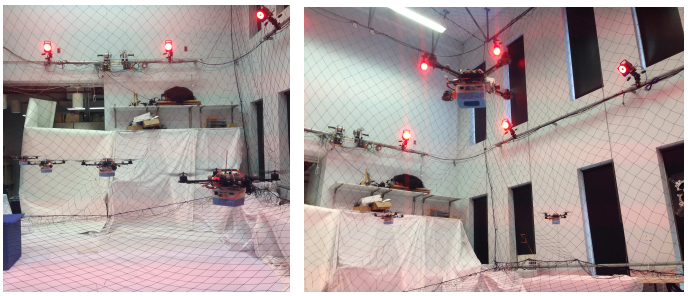
## 1 Introduction

Multi-robot teams have been used to complete many complex tasks, including warehouse management [1], package delivery [2], and surveillance [3]. These applications often require each agent to safely and quickly navigate to goal locations to complete tasks, where tasks are non-interchangable between robots. This is called the *labeled multi-robot planning problem*.

Many approaches have been presented for solving this problem, including variations on traditional single-robot planners [4], graph-search [5], optimization-based [6], and rule-based [7]. However, algorithmic guarantees often assume robots are perfect kinematic or kinodynamic agents. In reality, robots often have higher-order dynamics that could potentially make motions planned by kinematic algorithms infeasible. This problem of generating feasible, optimal, collision-free trajectories for dynamic robots has mostly been neglected.

As a result, experimental validation of multi-robot algorithms has been limited to simulation or teams of four to five ground robots that can be approximated as kinematic, such as the Pioneer [8], iRobot Create [9], Dr. Robot Jaguar Lite [10], or other similar custom platforms [11] [12]. These robots move at relatively slow velocities in planar environments, and so, results on these platforms do not necessarily indicate applicability to fast-moving vehicles with higher-order dynamics, such as quadrotors.

The final publication is available at Springer via https://doi.org/10.1007/978-3-319-60916-4_34.



(a) (b)

Fig. 1: A multi-robot team of five quadrotors.

A number of multi-robot planning algorithms have been successfully implemented on quadrotors. For example, Alonso-Mora *et al.* [13] propose methods for collision avoidance using Velocity Obstacles, with experimental validation using two robots. Omidshafiei *et al.* [14] propose a Decentralized Partially Observable Semi-Markov Decision Process framework for planning and include experiments using four quadrotors. Yet, even in these applications, quadrotors move relatively slowly and operate as planar systems at a common altitude.

Similar efforts have been made in the *unlabeled* multi-robot planning domain, where robots can swap goals. Turpin *et al.* [3] demonstrate navigation of six small vehicles through an obstacle-filled space and Mohta *et al.* [15] demonstrate six quadrotors cooperatively completing an outdoor surveillance mission. However, these approaches are not suitable for applications like package delivery, where robots are non-interchangeable.

This work aims to experimentally validate a centralized multi-robot planning and trajectory generation algorithm that explicitly accounts for robot dynamics. Fig. 1 shows an image of our experimental testbed.

We present results from three experiments. First, we execute solution trajectories for a team of four robots for problems with various levels of congestion in a limited workspace. We demonstrate that our algorithm successfully accounts for the fourth-order, underactuated, dynamics of the quadrotor and plans safe, yet fast, motions. Next, we test the robots' robustness to aerodynamic interactions between co-planar neighbors. Finally, we apply our algorithm to quadrotor teams operating in the full three-dimensional workspace. Specifically, we show that our algorithm is able to improve trajectory tracking by navigating robots away from the downwash of their neighbors.

## 2 Technical Approach

Consider a team of robots operating in an obstacle-free two-dimensional workspace. Each robot is contained in a disk of radius $R$ and has $n^{th}$-order dynamics:

$$\mathbf{x}_i^{(n)}(t) = \mathbf{u}_i(t) \tag{1}$$

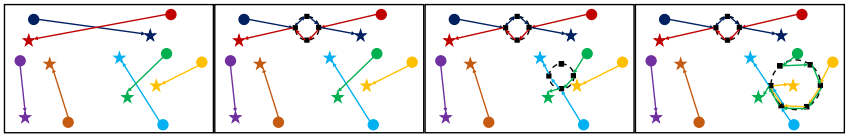
$\mathbf{x}_i$, $\mathbf{u}_i$ are the position and input of robot $i$, respectively. Let $\mathbf{X}_i$ denote the *state*:

$$\mathbf{X}_i = [\mathbf{x}_i \;\; \dot{\mathbf{x}}_i \;\; ... \;\; \mathbf{x}_i^{(n-1)}]^T \tag{2}$$

We define the *labeled multi-robot planning problem* as follows: given $N$ robots, indexed using $i \in [1, N]$, with start positions $\mathbf{s}_i \in \mathbb{R}^2$ and goal positions $\mathbf{g}_i \in \mathbb{R}^2$, find trajectories $\gamma_i(t) : \mathbb{R} \to \mathbb{R}^2$ that navigate all robots safely from start to goal.

Our proposed algorithm solves this problem in two steps. In the *motion planning* step, we find a safe motion plan for each robot, $\mathcal{M}_i = \{\mathcal{T}, \mathcal{X}_{i,des}\}$. $\mathcal{T} = \{t_0, t_1, ..., t_m\}$ is a set of times and $\mathbf{X}^t_{des,i} = \{\mathbf{X}^{t_0}_{des,i}, \mathbf{X}^{t_1}_{des,i}, ..., \mathbf{X}^{t_1}_{des,m}\}$ is the set of corresponding desired states. For each $\mathbf{X}^{t_j}_{des,i}$, the desired position must be specified, however, higher derivatives can be unspecified. While all robots have unique sets of desired states, they share a common $\mathcal{T} = \{t_0, t_1, ...\}$.

We find this motion plan using the OMP_CHOP algorithm [16], summarized in Fig. 2. Each robot's initial motion plan is a straight-line trajectory to its goal. The algorithm iteratively resolves inter-robot collisions by constructing *Circular HOlding Patterns* (CHOPs). Each CHOP consists of a series of waypoints navigating the affected robots in a collision-free manner to their goals. The final solution, Fig. 2c, allows robots to take direct trajectories to their goals when possible and navigates them into CHOPs through congested areas. Algorithm details can be found in [16].



(a) Iteration 0 (b) Iteration 1 (c) Iteration 2 (d) Final motion plan

Fig. 2: Illustration of motion planning algorithm. Robots must navigate from start positions, circles, to corresponding goals, stars of the same color. Fig. 2b shows a CHOP between red and navy robots, with waypoints as black squares. In Fig. 2c, a CHOP is constructed between blue and green robots, causing a collision. Fig. 2d refines blue, green, and yellow robots' plans to one CHOP.

In the *trajectory generation* step, we transform motion plans, $\mathcal{M}_i$, to trajectories, $\gamma_i(t)$. Each $\gamma_i(t)$ is a piecewise polynomial, $\gamma_i(t) = [x_i(t) \;\; y_i(t)]^T$, where:

$$x_i(t) = \begin{cases} \sum_{k=0}^{2n-1} c^i_{k,1,x} t^k, & t_0 \le t < t_1 \\ \sum_{k=0}^{2n-1} c^i_{k,2,x} t^k, & t_1 \le t < t_2 \\ ... \\ \sum_{k=0}^{2n-1} c^i_{k,m,x} t^k, & t_{m-1} \le t \le t_m \end{cases} \tag{3}$$

$y_i(t)$ is defined analogously. Let vector $\mathbf{d}_i$ contain the unknown coefficients:

$$\begin{aligned} \mathbf{d}_i = \big[&c^i_{0,1,x} \;\; c^i_{1,1,x} \;\; ... \;\; c^i_{2n-1,1,x} \;\; c^i_{0,2,x} \;\; c^i_{1,2,x} \;\; ... \;\; c^i_{2n-2,m,x} \;\; c^i_{2n-1,m,x} \\ &c^i_{0,1,y} \;\; c^i_{1,1,y} \;\; ... \;\; c^i_{2n-2,m,y} \;\; c^i_{2n-1,m,y}\big]^T \end{aligned} \tag{4}$$

We solve for $\mathbf{d}_i$ using a Quadratic Program (QP) that minimizes the quadratic cost functional:

$$J_i = \int_{t_0}^{t_m} \left|\left| \frac{d^n}{dt^n} \gamma_i(t) \right|\right|_2^2 dt \tag{5}$$

subject to:

1. Waypoint constraints: Trajectory begins and ends at rest at start and goal.
2. Continuity constraints: Trajectory is at least $\mathcal{C}^{n-1}$ everywhere.
3. Safety constraints: All robots' trajectories are mutually collision-free.
4. Workspace constraints: Trajectory remains in the given workspace.

Details of the QP problem formulation can be found in [17]. This proposed algorithm is guaranteed to be safe and complete.

## 3 Experimental Architecture

We apply our algorithm to a team of quadrotors. Adopting the dynamic model presented by Mellinger *et al.* [18], let $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ be unit coordinate axes of an inertial reference frame and $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ be a body frame fixed to a robot. Let $m$ represent the robot's mass, $\mathcal{I}$ represent the inertia matrix, and $g$ represent the gravity constant. Let $\mathbf{r} \in \mathbb{R}^3$ denote the position of the robot's center of mass, $R \in SO(3)$ denote the world-to-body rotation matrix describing its orientation, and $\mathbf{\Omega} \in \mathbb{R}^3$ denote its body-frame angular velocity. Finally, the inputs are represented by a thrust force of magnitude $f \in \mathbb{R}$ and a moment vector $\mathbf{M} \in \mathbb{R}^3$ expressed in body-frame components. The vehicle dynamics are given by:

$$\begin{aligned} m(\ddot{\mathbf{r}} + g\mathbf{e}_3) &= fR\mathbf{e}_3 \\ \mathcal{I}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathcal{I}\mathbf{\Omega} &= \mathbf{M} \end{aligned} \tag{6}$$

Robots navigate using the controller in [19]. The desired thrust is:

$$f = (k_x \mathbf{e}_x + k_v \mathbf{e}_v + mg\mathbf{e}_3 + m\ddot{\mathbf{r}}_d) \cdot R\mathbf{e}_3 \equiv \mathbf{f}_{des} \cdot R\mathbf{e}_3 \tag{7}$$

$\mathbf{e}_x$ and $\mathbf{e}_v$ are the position and velocity errors, respectively, $k_x$ and $k_v$ are positive gains, and $\ddot{\mathbf{r}}_d$ is a feed-forward acceleration. The desired moments are:

$$\mathbf{M} = k_R \mathbf{e}_R + k_{\mathbf{\Omega}} \mathbf{e}_{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathcal{I}\mathbf{\Omega} - \mathcal{I}\left(\hat{\mathbf{\Omega}} R^T R_c \mathbf{\Omega}_c - R^T R_c \dot{\mathbf{\Omega}}_c\right) \tag{8}$$

$\mathbf{e}_R$ and $\mathbf{e}_{\mathbf{\Omega}}$ are attitude and angular velocity error terms, respectively, $k_R$ and $k_{\mathbf{\Omega}}$ are positive gains, and $\dot{\mathbf{\Omega}}_c$ is a commanded angular velocity calculated from the desired trajectory. This controller is provably exponentially stable as long as the initial attitude error is less than $\pi/2$. Further details can be found in [19].

Mellinger *et al.* [18] show that quadrotors are $4^{th}$ order *differentially flat* systems, with flat outputs $[\mathbf{r} \quad \psi]^T$, where $\psi$ is the yaw angle of the quadrotor. As a result, a set of sufficiently smooth time-parametrized trajectories for $x, y, z$

and $\psi$ of the quadrotor is sufficient to calculate the feed-forward and commanded terms necessary for the controller. Given a labeled multi-robot planning problem, we use the algorithm described in Section 2, with $n = 4$, to generate minimum-snap trajectories in the $x$ and $y$ dimensions. We then maintain $\psi = 0$ and a constant altitude $z$ for all time.

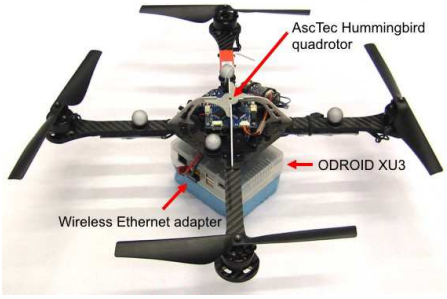



Fig. 3: Ascending Technologies Hummingbird quadrotor used for experiments.

Our experimental platform is the Ascending Technologies Hummingbird[1], pictured in Fig. 3. The robot has a rotor-tip-to-rotor-tip distance of 54 cm.

Fig. 4 illustrates the architecture of our experimental testbed. Each robot carries an ODROID XU3 computer, a wireless ethernet adaptor, and a USB serial adaptor. This approach is a departure from previous multi-robot experiments, which used ZigBee modules for communication and lacked the ability to communicate rich or large amounts of information (ex. images). The position and velocity of each robot is obtained from a Vicon[2] motion capture system at 100 Hz. The robots are operating in a workspace of about 3.5 m by 3.5 m in the $xy$-directions and 1.5 m in height. The total mass of the quadrotor, with onboard equipment, is 703 g.

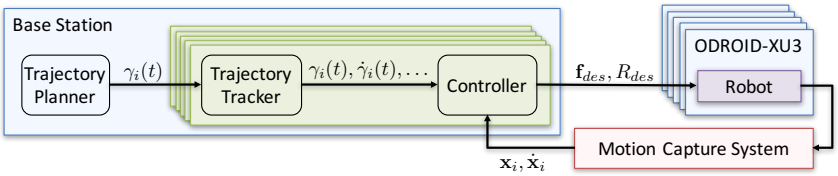



Fig. 4: Architecture of the experimental system. A base station runs our trajectory planner along with a trajectory tracker and controller for each robot. A desired force vector ($\mathbf{f}_{des}$) and attitude ($R_{des}$) are computed and sent to the robot's low-level controller. Finally, the state of the robot is observed using a motion capture system.

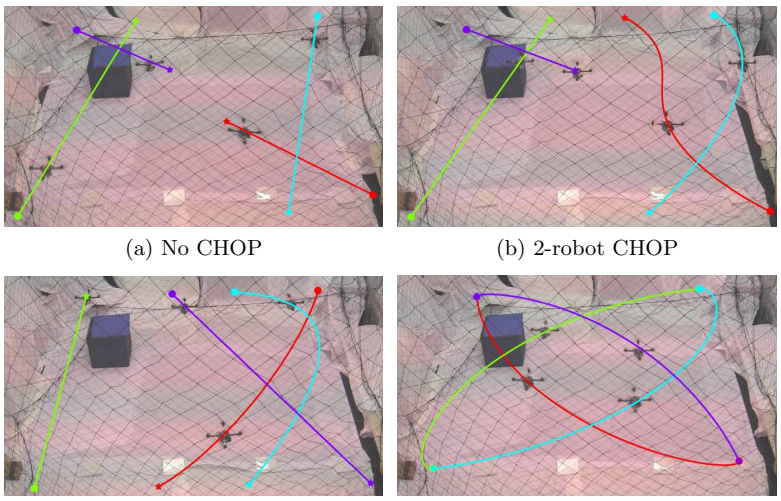
[1] http://www.asctec.de/en/
[2] http://www.vicon.com/

## 4 Results

In this section, we present results from our experiments, with video at https://youtu.be/XfC6zvE1uIc. For these experiments, robots were flown at a constant altitude of 1.5 m, and we only analyze motion in the plane.

### 4.1 Experimental Verification of Proposed Algorithm

We first validate our algorithm's ability to generate safe trajectories for the team under different geometries of start and goal locations. We executed solution trajectory sets to four different problems for a four-robot team. Each scenario demands different subsets of the team to enter a CHOP to safely navigate to their goals. These examples are pictured in Figs. 5a - 5d.



(a) No CHOP (b) 2-robot CHOP

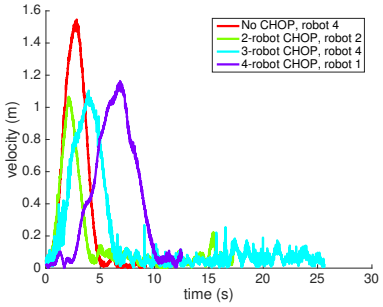
(c) 3-robot CHOP (d) 4-robot CHOP

Fig. 5: Execution of four-robot trajectory sets, single trial. Robots must navigate from start positions, circles, to assigned goals, stars of the same color.

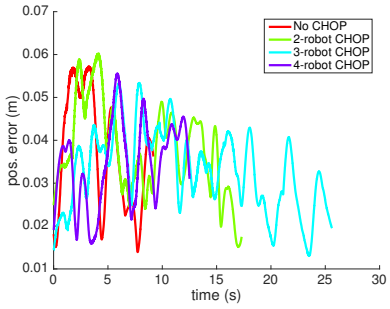
In Fig. 5a, while robots' straight-line paths intersect, their trajectories' time parameterizations are such that they do not collide and no holding patterns are necessary. Fig. 5b shifts the goal of the red robot. As a result, the red and blue robots must navigate around each other. Figs. 5c and 5d show three and four robots in a holding pattern, respectively. As can be expected, holding patterns involving more robots take up more of the available workspace.

Fig. 6a displays the magnitude of the robots' velocities in the plane ("forward velocity") over time for the fastest robot in each solution set. In each set, a robot exceeds 1 m/s, and in the most aggressive case, a robot reaches a maximum velocity of almost 1.6 m/s.

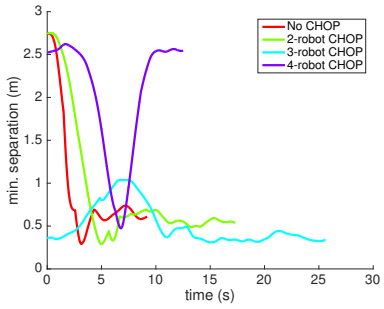
We reliably executed each solution set over five trials. Fig. 6b plots the magnitude of the position error in the plane ("planar error") averaged across all robots




(a) Forward velocity of fastest robot.




(b) Avg. position error.




(c) Min. robot separation.

Fig. 6: Performance statistics for four-robot solution sets, single trial.

within one representative trial. Fig. 6c reports the minimum tip-to-tip separation between robots. The average position error is consistently less than 6 cm, and in each scenario, robots' propellers come within 46 cm (0.85 body-lengths).

### 4.2 Robustness to Unmodeled Dynamics

Next, we test the robustness of our system to unmodeled dynamics by observing trajectory tracking error when executing various holding patterns. This is an indicator of the planned trajectories' optimality with respect to robot dynamics.

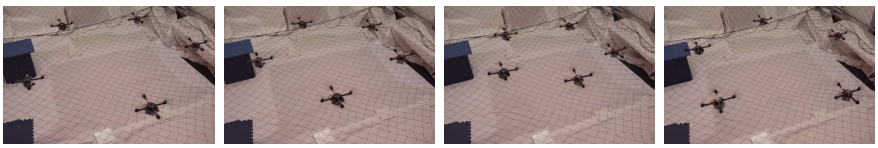
Table 1: Trajectory errors for unmodeled dynamics experiments over five trials.

| Experiment | Robots | Max. vel. (m/s) | Min. sep. (m) | Avg. error (m) | Max. err. (m) |
|---|---|---|---|---|---|
| 1 | 3 robots | 0.61 | 0.69 | 0.045 | 0.12 |
| 2 | 3 robots | 1.4 | 0.70 | 0.047 | 0.12 |
| 3 | 3 robots | 1.9 | 0.67 | 0.053 | 0.14 |
| 4 | 5 robots | 1.1 | 0.28 | 0.039 | 0.14 |

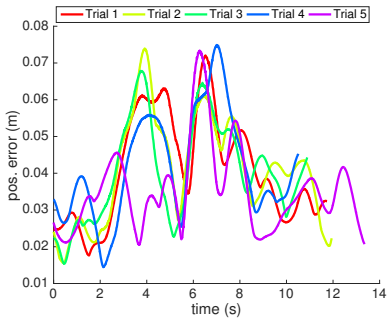
Table 1 characterizes CHOPs with the maximum forward velocity and the minimum separation between robots' propellers. We report planar error averaged across all robots in all trials and the maximum error of any robot over all trials.

Based on the three-robot experiments, as the robots' maximum velocities increase, both the average and maximum errors increase. This is likely caused

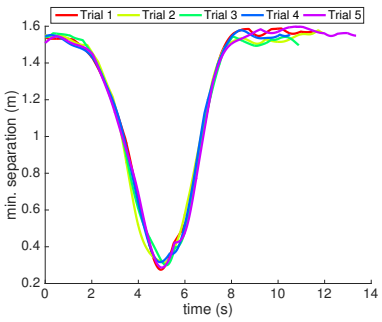
by both the increased difficulty in trajectory tracking at higher velocities and the increase in aerodynamic disturbances from neighboring vehicles as propeller speeds increase. In the most aggressive holding pattern, robots reach maximum velocities of almost 2 m/s. However, the average trajectory error is still never greater than 6 cm. Further, the maximum tracking error is less than 15 cm.



(a) Frame 1. (b) Frame 2. (c) Frame 3. (d) Frame 4.




(e) Avg. position error over time.




(f) Min. propeller separation over time.

Fig. 7: Five robots executing the holding pattern in Table 1, Experiment 4.

We were also able to successfully execute a CHOP with five robots, as shown in Fig. 7, safely within the workspace limits. Even with neighboring propellers coming within 28 cm (0.52 body-lengths) of each other, the average and maximum errors are still less than 4 cm and 15 cm, respectively. Figs. 7a - 7d show snapshots of the quadrotors in flight.

Further, Fig. 7e plots the magnitude of the position error averaged across all robots for each trial. The error is generally around 2-3 cm and always less than 8 cm. Fig. 7f displays the minimum rotor-tip-to-rotor-tip distance between any pair of robots. While Table 1 reflects little difference in average errors across experiments, we see from the time-based plots that there is in fact an increase in position error as robots move closer to each other.

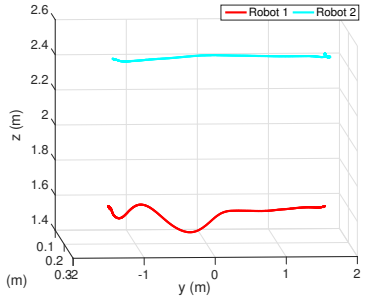
## 5 Application: Avoiding Downwash in Three-Dimensional Workspaces for Improved Trajectory Tracking

We finally apply our planar multi-robot trajectory generation algorithm to a three-dimensional workspace. A tempting solution to the labeled planning problem for quadrotors is to simply stagger the vehicles' altitudes and allow them to move directly to their designated goals. While this is theoretically possible, the

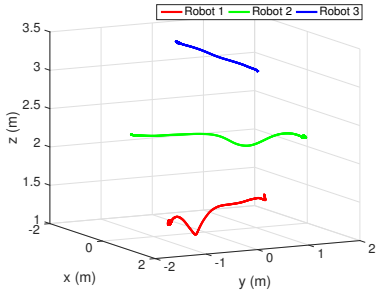
downwash from robots at higher altitudes could perturb robots at lower altitudes in unpredictable ways, potentially making this solution unsafe.

In fact, Mellinger *et al.* [20] characterize the downwash effect for Hummingbird quadrotors to be "most concentrated in a cylindrical region with a radius of approximately 0.5 m extending to a height of 1.5 m below the quadrotor. This cylindrical region bounds the volume where the $z$ displacement error is greater than 5 cm." Thus, even when executing trajectories at different altitudes, it could be advantageous for quadrotors to avoid flying directly below neighbors.

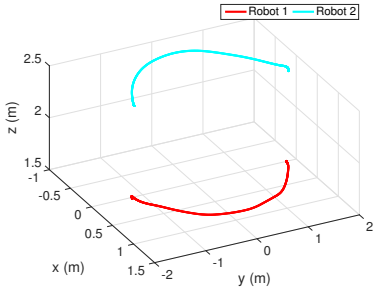
We conducted experiments with teams of two and three robots. We assign robots' start and goal positions in the full three-dimensional space. Each robot's start and goal are at the same altitude and no pair of robots share starts or goals with the same planar coordinates. First, we allow robots to take minimum-snap straight-line trajectories to their goals. Next, we apply our proposed algorithm as if all robots were operating on a common plane. Each robot then executes this CHOP trajectory at its originally assigned altitude. In this way, we use the CHOP to ensure that vehicles do not "collide" with the downwash of their neighbors. We executed five trials of each trajectory set.



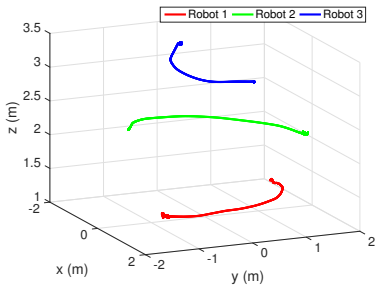

(a) 2 robots, straight-line trajectories.




(b) 3 robots, straight-line trajectories.




(c) 2 robots, one CHOP.




(d) 3 robots, one CHOP.

Fig. 8: Execution of straight-line trajectories vs. holding pattern, single trial.

Fig. 8 displays the actual trajectories from one representative trial of each experiment. Figs. 8a - 8b show tracking of the straight-line trajectories. In each

Table 2: Vertical error for three-dimensional experiments over five trials.

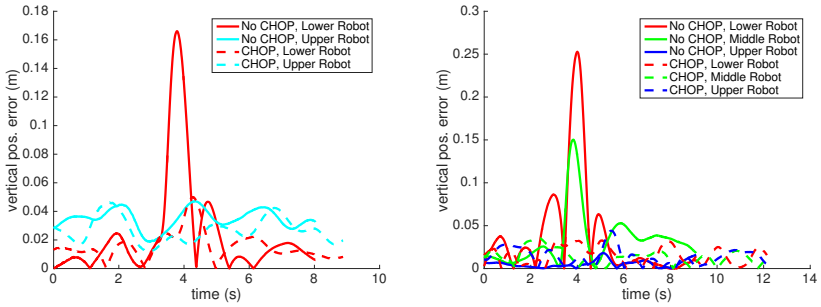
| Robots | Max. vel. (m/s) | Vert. sep. (m) | Not single CHOP errors Avg. (m) | Not single CHOP errors Max. (m) | Single CHOP errors Avg. (m) | Single CHOP errors Max. (m) |
|---|---|---|---|---|---|---|
| 2 | 1.3 | 0.90 | 0.020 | 0.17 | 0.016 | 0.050 |
| 3 | 1.3 | 1.0 | 0.026 | 0.27 | 0.017 | 0.12 |

case, there is a significant disturbance in the vertical direction for robots at lower altitudes. Figs. 8c - 8d show the proposed CHOP trajectories. The vertical trajectory error significantly decreases.

Table 2 lists the maximum forward velocities and vertical separations between robots for each experiment, along with error statistics averaged over five trials of each trajectory set. For each experiment, the vertical separation between robots was about 1 m, within the 1.5 m height of the disturbance region characterized by Mellinger *et al.* [20]. In each case, the maximum vertical error decreases dramatically by over 10 cm after the introduction of the CHOP to the robots' trajectories. The average vertical error decreases as well.

Fig. 9 displays the absolute value of the vertical error over time for a representative trial of each experiment. Fig. 9a shows that when two robots are executing straight-line trajectories, the higher robot's vertical error remains approximately 4 cm. However, the lower robot has a large vertical error at around 4 s, when it crosses paths with its neighbor. When robots execute a CHOP, the lower robot's vertical error decreases to around 4 cm throughout its trajectory.

Similarly, Fig. 9b shows that in the three-robot experiment, while executing straight-line trajectories, the middle and lower robots have large vertical errors at around 4 s. The middle robot is perturbed around 15 cm, while the lower robot, subject to disturbances from both neighbors above it, is perturbed vertically by over 25 cm. However, when all three robots execute a single CHOP, vertical errors for all robots remain at below 5 cm throughout their trajectories.




(a) 2-robot experiment. (b) 3-robot experiment.

Fig. 9: Absolute vertical error over time in three-dimensional experiments, single trial.

Table 3 lists the average and maximum planar errors of robots in each experiment over five trials. We confirm that coordination of all robots in a CHOP does not significantly affect planar errors.

Table 3: Planar error for three-dimensional experiments over five trials.

| Robots | Max. vel. (m/s) | Vert. sep. (m) | Not single CHOP errors | | Single CHOP errors | |
|---|---|---|---|---|---|---|
| | | | Avg. (m) | Max. (m) | Avg. (m) | Max. (m) |
| 2 | 1.3 | 0.90 | 0.040 | 0.10 | 0.037 | 0.099 |
| 3 | 1.3 | 1.0 | 0.042 | 0.14 | 0.038 | 0.16 |

Overall, these results suggest that for quadrotor teams, simply staggering altitudes of vehicles might not be a realistic solution because of the large vertical perturbations that occur as robots cross their neighbors. As a result, coordination between vehicles might still be necessary, particularly when the vertical separation between robots is not large enough to safely allow for deviations. Experiments show that our algorithm is a feasible solution to this problem.

## 6 Conclusions

In this work, we present experimental validation of a centralized labeled multi-robot planning and trajectory generation algorithm. Unlike methods that model robots as kinematic agents, our algorithm returns optimal trajectories with respect to the robots' dynamics. Experiments verify that solution trajectories are safe and practical for a quadrotor team. We further show that robots can robustly track these trajectories even when operating in close proximity. Finally, we demonstrate the successful application of our algorithm to three-dimensional workspaces. We believe these experimental insights will allow for development of larger, more complex quadrotor teams that may be used for tasks such as package delivery.

**Acknowledgments.** We gratefully acknowledge the support of ONR grants N00014-09-1-1051 and N00014-09-1-103, ARL grantW911NF-08-2-0004, ARO grant W911NF-13-1-0350, and Exyn Technologies. Sarah Tang is supported by NSF Research Fellowship Grant No. DGE-1321851.

## References

1. Enright, J.J., Wurman, P.R.: Optimization and coordinated autonomy in mobile fulfillment systems. In: AAAI Conference on Artificial Intelligence. (2011)
2. Forbes: Meet amazon prime air, a delivery-by-aerial-drone project (December 2013)
3. Turpin, M., Mohta, K., Michael, N., Kumar, V.: Goal assignment and trajectory planning for large teams of interchangeable robots. Autonomous Robots **37**(4) (2014) 401–415
4. Goldenberg, M., Felner, A., Stern, R., Sharon, G., Sturtevant, N., Holte, R.C., Schaeffer, J.: Enhanced partial expansion A*. Journal of Artificial Intelligence Research **50**(1) (2014) 141–187
5. Wagner, G., Choset, H.: Subdimensional expansion for multirobot path planning. Artificial Intelligence **219** (2015) 1–24

6. Yu, J., Rus, D.: An effective algorithmic framework for near optimal multi-robot path planning. In: The International Symposium on Robotics Research (ISRR). (2015)
7. Luna, R., Bekris, K.E.: Push and swap: Fast cooperative path-finding with completeness guarantees. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI). (2011) 294–300
8. Bennewitz, M., Burgard, W., Thrun, S.: Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. Robotics and Autonomous Systems **41**(2) (2002) 89–99
9. Desaraju, V., How, J.P.: Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees. In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA). (2011) 4956–4961
10. Wiktor, A., Scobee, D., Messenger, S., Clark, C.: Decentralized and complete multi-robot motion planning in confined spaces. In: Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2014) 1168–1175
11. Clark, C.M., Bretl, T., Rock, S.M.: Applying Kinodynamic Randomized Motion Planning with a Dynamic Priority System to Multi-Robot Space Systems. In: Proceedings of the 2002 IEEE Aerospace Conference. (March 2002)
12. Pallottino, L., Scordio, V.G., Frazzoli, E., Bicchi, A.: Decentralized cooperative policy for conflict resolution in multi-vehicle systems. IEEE Transactions on Robotics **23** (2007) 1170–1183
13. Alonso-Mora, J., Naegeli, T., Siegwart, R., Beardsley, P.: Collision avoidance for aerial vehicles in multi-agent scenarios. Autonomous Robots **39**(1) (2015) 101–121
14. Omidshafiei, S., akbar Agha-mohammadi, A., Amato, C., How, J.P.: Decentralized control of partially observable markov decision processes using belief space macro-actions. In: IEEE International Conference on Robotics and Automation (ICRA). (2015)
15. Mohta, K., Turpin, M., Kushleyev, A., Mellinger, D., Michael, N., Kumar, V.: Quadcloud: A rapid response force with quadrotor teams. In: The 14th International Symposium on Experimental Robotics (ISER), Springer International Publishing (2014) 577–590
16. Tang, S., Kumar, V.: A complete algorithm for generating safe trajectories for multi-robot teams. In: International Symposium on Robotics Research. (2015)
17. Tang, S., Kumar, V.: Safe and complete trajectory generation for large teams of robots with higher-order dynamics. In: Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (October 2016)
18. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA). (2011) 2520 – 2525
19. Lee, T., Leok, M., McClamroch, N.H.: Control of complex maneuvers for a quadrotor UAV using geometric methods on SE(3). In: Asian Journal of Control. Volume 15. (2011) 391–408
20. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The GRASP multiple micro-UAV testbed. IEEE Robotics Automation Magazine **17**(3) (Sept 2010) 56–65