

Multi-Robot Trajectory Generation for an Aerial Payload Transport System

Sarah Tang, Koushil Sreenath, and Vijay Kumar

Abstract In this work, we consider the problem of planning safe, feasible trajectories for a team of quadrotors with slung-loads operating in an obstacle-free, three-dimensional workspace. We are particularly interested in generating *dynamic trajectories* – trajectories where robots’ payloads are allowed to swing in accordance with the system’s natural dynamics — for fast, agile, coordinated payload transportation. This capability is applicable to tasks such as construction, where a single crane performing sequential tasks could be replaced by multiple quadrotors performing tasks in parallel for increased efficiency. We model this problem as a *labeled multi-robot planning problem*, where robots must navigate payloads from given start positions to fixed, non-interchangeable goal positions. Our system presents three novel challenges: (1.) Each vehicle has eight degrees-of-freedom, significantly increasing the size of the team’s joint state space. (2.) Each vehicle is a nonlinear, 6th-order dynamical system with four degrees of under-actuation. (3.) Each vehicle is a multi-body system. We present a safe and complete Quadratic Programming solution and validate its practicality with experiments containing up to nine quadrotors.

1 Motivation

In the past decades, many algorithms for solving the *labeled multi-robot planning problem* (MPP) have been proposed. In this paradigm, robots begin at fixed start positions and must safely navigate to assigned goal positions. We assume our system is *centralized* and operating in an obstacle-free, three-dimensional workspace.

Many MPP algorithms have focused on first-order robots, as unfortunately, even the problem of navigating multiple disks is NP-Hard [23]. In particular, the size of

S. Tang
GRASP Lab, University of Pennsylvania, e-mail: sytang@seas.upenn.edu

K. Sreenath
University of California, Berkeley, e-mail: koushils@berkeley.edu

V. Kumar
GRASP Lab, University of Pennsylvania, e-mail: kumar@seas.upenn.edu

the system’s joint state space grows exponentially with the number of agents. While search-based algorithms [10] can find optimal motion plans in the joint state, they quickly become computationally intractable for even modest-sized teams. Performance can be improved with approaches like an Integer Linear Program formulation [30], rule-based [16] methods, or decomposition into sub-problems [26, 11]. These *kinematic* approaches are often complete and either optimal or near-optimal. However, they specify robots’ motions in constant-velocity segments (for example, coordinated movements along edges of a graph-based workspace discretization), which can contain sharp turns infeasible for dynamic systems.

Continuous techniques that also optimize a velocity profile, such as prioritization [2], path-velocity decompositions [19], sampling-based approaches [22] or optimization formulations [12] are extendable to high-order systems. However, these techniques have been less amicable to decoupling and often require joint optimization of all robots’ trajectories. As each robot’s individual state space is also larger, the problem becomes intractable even more quickly.

A hybrid approach can potentially leverage the model simplicity of kinematic planners and the dynamic feasibility of continuous ones. Our previous work presents Hold or take Optimal Plan (HOOP), a centralized, two-dimensional trajectory generation algorithm for higher-order holonomic systems [29]. A discrete planner first finds a safe, but dynamically infeasible, piecewise linear trajectory for each robot. These are subsequently used to partition the workspace into *safe corridors* — a series of overlapping convex regions. Single-robot trajectory generation methods [5, 7, 15] can then be leveraged to independently plan dynamically feasible trajectories for each robot within its respective corridor. The discrete planner and trajectory generator can be independently modified based on application needs. This paradigm has been experimentally validated for quadrotor platforms [20, 29].

However, applications like warehouse tasks or construction could require robots with more degrees of freedom and coupled rigid bodies. We consider the problem completing a set of slung-load transportation tasks with quadrotor teams. This problem is representative of several challenges. First, each vehicle contains eight degrees-of-freedom, exacerbating the exponential growth of the joint configuration space. Second, each agent is a 6th-order, nonlinear dynamical system (ie. the quadrotor’s moment input actuates the payload’s 6th derivative) with four degrees of under-actuation. Finally, each robot is a multi-body system.

A number of works explore aerial payload manipulation. For example, [1] and [14] use quadrotor teams to construct building-like structures. However, payloads are rigidly held underneath the vehicle and do not require modeling as a separate body. A number of approaches have been proposed for single-quadrotor trajectory generation with slung-loads. Past works have typically focused on eliminating load swing during transport [3, 6]. However, this conservative approach severely compromises vehicles’ agility. More recently, the development of geometric controllers that can stabilize large payload angles [25] has inspired the design of planners that leverage the system’s natural payload swings during obstacle avoidance [8, 27]. Controllers for cooperative manipulation, where multiple robots transport a shared payload, have also been explored [24]. To the best of our knowledge,

this is the first work that plans trajectories for multi-body vehicles with complex dynamics in a multi-robot setting.

The contributions of this work include the extension of HOOP [29], a two-dimensional, single-body trajectory generation algorithm, to the quadrotor-with-payload system and the formulation of the trajectory generation problem as a Quadratic Program (in contrast with a previous Mixed Integer Quadratic Program formulation [27]). We provide experimental validation of our algorithm on a nine-robot team. Notably, while the algorithm reverts to swing-free trajectories in the worst case, experimental results show that planned trajectories can be very dynamic.

2 Dynamic Modeling

To begin, we formulate a dynamic model for the quadrotor-with-load system using variables in Table 1. We model the load as a point-mass and the cable as massless.

Table 1: Variables for the quadrotor-with-payload system.

\mathcal{I}, \mathcal{B}	World, body frame of quad	$m_Q, m_L \in \mathbb{R}$	Mass of quad, load
J	Inertia tensor of quad	$r_Q, r_L \in \mathbb{R}$	Radius of quad, load
$f \in \mathbb{R}, \mathbf{M} \in \mathbb{R}^3$	Thrust, moment input in \mathcal{B} to quad	$\mathbf{x}_Q, \mathbf{x}_L \in \mathbb{R}^3$	Position of quad, load, in \mathcal{I}
$\mathbf{p} \in \mathbb{S}^2$	Unit vector from quad to load, in \mathcal{I}	$\boldsymbol{\omega} \in \mathbb{R}^3$	Angular velocity of load, in \mathcal{I}
$R \in SO(3)$	Rotation matrix of quad from \mathcal{B} to \mathcal{I}	$\boldsymbol{\Omega} \in \mathbb{R}^3$	Angular velocity of quad, in \mathcal{B}
$\psi \in SO(2)$	Quad yaw angle	$l \in \mathbb{R}$	Length of cable

Let \mathcal{I} with axes $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$ be an inertial world frame and \mathcal{B} with axes $\{\mathbf{b}_x, \mathbf{b}_y, \mathbf{b}_z\}$ be a body-fixed frame. The system's configuration evolves on $SE(3) \times \mathbb{S}^2$ with state:

$$\mathbf{x}(t) = [\mathbf{x}(t)_L^\top \dot{\mathbf{x}}(t)_L^\top \mathbf{p}(t)^\top \boldsymbol{\omega}(t)^\top R(t) \boldsymbol{\Omega}(t)^\top]^\top.$$

The system's input is a thrust magnitude and moment:

$$\mathbf{u}(t) = [f(t) \mathbf{M}(t)^\top]^\top.$$

We represent the system in a *coordinate-free* manner, that is, we do not rely on local parameterizations of the payload orientation, \mathbf{p} , and the quadrotor attitude, R . We can apply the Lagrange D'Alembert principle to find the system dynamics [25]:

$$\begin{aligned} \frac{d}{dt} \mathbf{x}_L(t) &= \dot{\mathbf{x}}_L(t) \\ (m_Q + m_L)(\ddot{\mathbf{x}}_L(t) + g\mathbf{e}_z) &= (\mathbf{p}(t) \cdot f(t)R(t)\mathbf{e}_z - m_Q l(\dot{\mathbf{p}}(t) \cdot \dot{\mathbf{p}}(t)))\mathbf{p}(t) \\ \dot{\mathbf{p}}(t) &= \boldsymbol{\omega}(t) \times \mathbf{p}(t) \\ m_Q l \dot{\boldsymbol{\omega}}(t) &= -\mathbf{p}(t) \times f(t)R(t)\mathbf{e}_z \\ \dot{R}(t) &= R(t)\hat{\boldsymbol{\Omega}}(t) \\ \dot{\boldsymbol{\Omega}}(t) &= J^{-1}(\mathbf{M}(t) - \boldsymbol{\Omega}(t) \times J\boldsymbol{\Omega}(t)). \end{aligned} \tag{1}$$

3 Problem Definition

Consider a team of N quadrotors, each carrying a suspended payload. Let superscripts $i \in \mathcal{I}_N = \{i \in \mathbb{N} \mid i \leq N\}$ denote variables associated with robot i . Robots must navigate from start positions, $\mathbf{s} = \{\mathbf{s}^1, \mathbf{s}^2, \dots, \mathbf{s}^N\} \in \mathbb{R}^{3N}$, to assigned goal positions, $\mathbf{g} = \{\mathbf{g}^1, \mathbf{g}^2, \dots, \mathbf{g}^N\} \in \mathbb{R}^{3N}$. We use \mathbf{r} to represent arbitrary positions in \mathbb{R}^3 . Let $\mathcal{B}_L(\mathbf{r}), \mathcal{B}_Q(\mathbf{r})$ denote a sphere centered at \mathbf{r} with radius r_L, r_Q , respectively. Let p denote a line segment $p_i = \{\mathbf{r} \mid \mathbf{r} = \mathbf{x}_L^i - \mathbf{p}^i \tau, \tau \in [0, l]\}$. Each vehicle's extent is described by the non-convex set $\mathcal{Q}(\mathbf{x}^i) = \mathcal{B}_L(\mathbf{x}_L^i) \cup \mathcal{B}_Q(\mathbf{x}_Q^i) \cup p^i$. We assume $r_Q > r_L$.

Let $\mathbf{X} = [(\mathbf{x}^1)^\top (\mathbf{x}^2)^\top (\mathbf{x}^3)^\top \dots (\mathbf{x}^N)^\top]^\top$ denote the joint state space. A general expression of the MPP is:

$$\mathbf{X}^*(t) = \underset{\mathbf{X}(t)}{\operatorname{argmin}} \int_0^{T_f} \mathcal{L}(t, \mathbf{X}(t), \dot{\mathbf{X}}(t), \dots, \mathbf{X}^{(n)}(t)) dt \quad (2)$$

subject to:

1. Dynamic constraints — $\mathbf{X}(t)$ is *dynamically feasible*:

$$\begin{aligned} &\forall i \in \mathcal{I}_N, k \in \mathcal{I}_K, \text{ where } K \text{ is a chosen parameter,} \\ &\exists \mathbf{u}^i(t) \text{ s.t. } \mathbf{x}^i(t), \mathbf{u}^i(t) \text{ satisfy Eq. 1, } \left\| \frac{d^k}{dt^k} \mathbf{x}_L^i(t) \right\|_2 \leq \delta_{k, \max}. \end{aligned} \quad (3)$$

We use limits $\delta_{\max} = \{\delta_{k, \max} \mid k \in \mathcal{I}_K\}$ to reflect actuator constraints.

2. Task constraints — each payload begins and ends at the designated locations:

$$\begin{aligned} &\forall i \in \mathcal{I}_N, \\ &\mathbf{x}^i(0) = [(\mathbf{s}^i)^\top \mathbf{0}^\top - \mathbf{e}_z^\top \mathbf{0}^\top \mathbb{I} \mathbf{0}^\top]^\top, \quad \mathbf{x}^i(T_f) = [(\mathbf{g}^i)^\top \mathbf{0}^\top - \mathbf{e}_z^\top \mathbf{0}^\top \mathbb{I} \mathbf{0}^\top]^\top. \end{aligned} \quad (4)$$

3. Safety constraints — robots do not collide with each other:

$$\forall t \in [0, T_f], i \in \mathcal{I}_N, j \in \mathcal{I}_N \neq i, \quad \mathcal{Q}(\mathbf{x}^i(t)) \cap \mathcal{Q}(\mathbf{x}^j(t)) = \emptyset. \quad (5)$$

$\mathcal{L}(t, \mathbf{X}(t), \dot{\mathbf{X}}(t), \dots, \mathbf{X}^{(n)}(t))$ is a to-be-specified cost functional. Payloads do not need to simultaneously arrive at their goals; payloads that arrive early should remain stationary. We assume \mathbf{s}, \mathbf{g} satisfy the separation constraints:

$$\forall i, j \neq i \in \mathcal{I}_N, \quad \|(\mathbf{s}^i - \mathbf{s}^j) \cdot [1 \ 1 \ 0]\|_2 > 2\sqrt{2}r_Q, \quad \|(\mathbf{g}^i - \mathbf{g}^j) \cdot [1 \ 1 \ 0]\|_2 > 2\sqrt{2}r_Q. \quad (6)$$

In obstacle-free space, any problem can be transformed into one satisfying Eq. 6 by expanding the start and goal positions radially outwards in the $\mathbf{e}_x - \mathbf{e}_y$ plane from the workspace center. Robots can sequentially move outwards to these altered start positions, apply the proposed algorithm to navigate to altered goal positions, then contract back to their originally designated goal positions.

We can now concisely state the labeled MPP as follows: given \mathbf{s}, \mathbf{g} satisfying Eq. 6, find trajectories $\mathbf{X}(t)$ that solve Eq. 2.

4 Technical Approach

In general, Eq. 2 cannot be solved exactly. Eqs. 3 and 5 impose nonlinear, non-convex constraints. Furthermore, Eq. 5 couples robots' trajectories, forcing Eq. 2 to be solved as a joint optimization problem in an exponentially growing search space. To identify a locally optimal solution, we extend the HOOP algorithm [29], which decomposes the MPP into path planning and trajectory generation steps, to the current system. We use *motion plan*, denoted $\mathcal{M}^i(t)$, to refer to a piecewise-constant-velocity parameterization along a piecewise linear path and *trajectory*, denoted $\gamma^i(t)$, to refer to a dynamically feasible parameterization along any path. We will use *path* to refer to the set of positions traversed by the payload.

4.1 Motion planning for kinematic collision avoidance

We define a motion plan $\mathcal{M}^i(t)$ as:

$$\mathcal{M}^i(t) = \begin{cases} \mathbf{x}_{d,0}^i + \frac{t-t_0^i}{t_1^i-t_0^i}(\mathbf{x}_{d,1}^i - \mathbf{x}_{d,0}^i) & t_0^i \leq t \leq t_1^i \\ \mathbf{x}_{d,1}^i + \frac{t-t_1^i}{t_2^i-t_1^i}(\mathbf{x}_{d,2}^i - \mathbf{x}_{d,1}^i) & t_1^i \leq t \leq t_2^i \\ \dots & \dots \\ \mathbf{x}_{d,m^i-1}^i + \frac{t-t_{m^i-1}^i}{t_{m^i}^i-t_{m^i-1}^i}(\mathbf{x}_{d,m^i}^i - \mathbf{x}_{d,m^i-1}^i) & t_{m^i-1}^i \leq t \leq t_{m^i}^i \end{cases}$$

$\mathcal{M}^i(t)$ can be represented concisely with a set of *breaktimes*, $\mathcal{T}^i = \{t_0^i, t_1^i, \dots, t_{m^i}^i\}$ and *waypoints*, $\mathcal{X}^i = \{\mathbf{x}_{d,0}^i, \mathbf{x}_{d,1}^i, \dots, \mathbf{x}_{d,m^i}^i\}$, where m^i represents the number of segments in $\mathcal{M}^i(t)$. Let \mathcal{M} denote the set of motion plans for the team.

We temporarily model each quadrotor-with-payload as a kinematic cylinder of height l and radius r_Q , centered at \mathbf{x}_L^i . Assuming Eq. 6, the motion planning problem reduces to navigating disks in the $\mathbf{e}_x - \mathbf{e}_y$ plane. We solve this using the iterative, geometric motion planning algorithm of HOOP, OMP_CHOP [28].

Fig. 1 provides an algorithm overview. Payloads must navigate from circles to stars of the same color. The algorithm begins by allowing each payload to move directly to its goal at $v_{max} = \delta_{1,max}$. The first collision between any subset of loads is identified and resolved by inserting a Circular Holding Pattern, a roundabout-like collision-avoidance maneuver (Panel 2). Each load will enter and navigate around the holding pattern until it exits at a coordinated time and position to safely reach its goal. The algorithm iteratively adds (Panel 3) and modifies (Panel 4) holding patterns until a safe set of trajectories is found. Payloads that do not collide with neighbors (ie. the orange payload) continue to take straight-line paths to their goals.

This algorithm is guaranteed to be safe and complete and has polynomial-time complexity with respect to the number of robots. Furthermore, it guarantees that the motion plan will be *disjoint*, namely: 1. All payloads share a common set of breaktimes \mathcal{T} (with different waypoint sets). 2. The distance between all pairs of paths at any given interval $s \in [t_{s-1}, t_s]$ is at least $2r_Q$. Let m to denote the number of

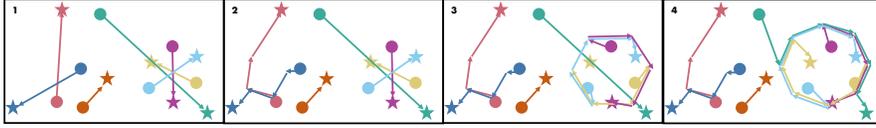


Fig. 1: Summary of OMP_CHOP. 1. Payloads begin by moving to their goals at v_{max} . 2. The first collision between the red and blue loads is resolved using a Circular Holding Pattern. 3. The next collision between the purple, light blue, and yellow loads is resolved using a separate Circular Holding Pattern. 4. The existing Circular Holding Pattern is refined to accommodate the teal payload. The algorithm terminates, as a collision-free motion plan is found.

trajectory segments. We note that not all payloads arrive at their goal simultaneously, and the motion plan guarantees stationary loads will not be subject to any collisions.

OMP_CHOP defines the motion plan in the $\mathbf{e}_x - \mathbf{e}_y$ plane. Each payload's motion plan in \mathbf{e}_z is a constant-velocity trajectory from $\mathbf{s}^i \cdot \mathbf{e}_z$ to $\mathbf{g}^i \cdot \mathbf{e}_z$.

4.2 Differential flatness for dynamic feasibility

The motion plan is clearly safe for a kinematic system with a swing-free payload. From this, we wish to derive a dynamically feasible, safe trajectory for the full dynamical system. To this end, we note that the quadrotor-with-load system is *differentially flat* [18] — there exists *flat outputs*, \mathbf{x}_f , such that \mathbf{x}, \mathbf{u} can be expressed with \mathbf{x}_f and its higher derivatives. For the quadrotor-with-load, the flat outputs are [25]:

$$\mathbf{x}_f = [\mathbf{x}_L \ \boldsymbol{\psi}]^\top.$$

As a result, there is a diffeomorphism mapping all sufficiently smooth trajectories $\mathbf{x}_f(t)$ to dynamically feasible trajectories $\mathbf{x}(t)$. In particular:

$$\mathbf{x}_Q = \mathbf{x}_L - l \frac{\ddot{\mathbf{x}}_L + g\mathbf{e}_z}{\|\ddot{\mathbf{x}}_L + g\mathbf{e}_z\|_2} = \mathbf{x}_L - l\mathbf{p}, \quad (7)$$

that is, the orientation of the payload is determined by its acceleration. Furthermore, the highest flat derivative required to recover \mathbf{x}, \mathbf{u} is the 6th-derivative, appearing in the expression for \mathbf{M} . Thus, the quadrotor-with-payload is a 6th-order system.

Assume the quadrotor yaw remains constant at 0, and let $\boldsymbol{\gamma}^i(t) : [0, t_f^i] \rightarrow \mathbb{R}^3$ be a payload trajectory in \mathbb{R}^3 . Differential flatness suggests $\boldsymbol{\gamma}^i(t)$ must be six times differentiable for dynamic feasibility. This motivates the reformulation:

$$\boldsymbol{\gamma}^*(t) = \operatorname{argmin}_{\boldsymbol{\gamma}(t)} \sum_i^N \int_0^{T_f} \left\| \frac{d^6 \boldsymbol{\gamma}^i(t)}{dt^6} \right\|_2^2 dt \quad (8)$$

subject to:

1. Dynamic constraints — $\forall i \in \mathcal{I}_N, k \in \mathcal{I}_K, \left\| \frac{d^k}{dt^k} \boldsymbol{\gamma}^i(t) \right\|_2 \leq \delta_{k,max}$.

2. Continuity constraints — $\gamma^i(t)$ is six times differentiable.
3. Waypoint constraints — $\forall i \in \mathcal{I}_N, s \in \mathcal{I}_{m-1}, \gamma^i(t_{s-1}) = \mathbf{x}_{d,s-1}^i$.
4. Task constraints —

$$\forall i \in \mathcal{I}_N, k \in \mathcal{I}_5, \gamma^i(0) = \mathbf{s}^i, \gamma^i(T_f) = \mathbf{g}^i, \frac{d^k \gamma^i}{dt^k}(0) = \mathbf{0}, \frac{d^k \gamma^i}{dt^k}(T_f) = \mathbf{0}. \quad (9)$$

5. Safety constraints —

$$\forall t \in [0, T_f], i \in \mathcal{I}_N, j \in \mathcal{I}_N \neq i, \mathcal{Q}(\gamma^i(t)) \cap \mathcal{Q}(\gamma^j(t)) = \emptyset.$$

We indirectly minimize input by minimizing the 6th derivative of $\gamma^i(t)$. Applying the Euler-Lagrange equation to the cost functional yields the condition:

$$\frac{d^{12} \gamma^i(t)}{dt^{12}} = 0,$$

motivating the choice of an 11th-order piecewise-polynomial as robots' trajectories:

$$\gamma^i(t) = \begin{cases} \gamma_1^i(t) = \sum_{j=0}^{11} c_{0,j}^i P_j & t_0 \leq t \leq t_1 \\ \gamma_2^i(t) = \sum_{j=0}^{11} c_{1,j}^i P_j & t_1 \leq t \leq t_2 \\ \dots \\ \gamma_m^i(t) = \sum_{j=0}^{11} c_{m-1,j}^i P_j & t_{m-1} \leq t \leq t_m. \end{cases}$$

Consider a decision vector containing all trajectory coefficients:

$$\mathbf{d} = [(\mathbf{d}^1)^\top (\mathbf{d}^2)^\top \dots (\mathbf{d}^N)^\top]^\top,$$

where \mathbf{d}^i contains the coefficients $c_{0,0}^i, c_{0,1}^i, \dots$ of $\gamma^i(t)$. It is well-established that the cost functional in Eq. 8 is quadratic with respect to \mathbf{d} and the continuity, waypoint, and task constraints are linear. In the absence of safety constraints, we can formulate the trajectory generation problem as a Quadratic Program (QP) [17].

4.3 Trajectory generation for dynamic collision avoidance

To incorporate dynamic and safety constraints into the QP, we leverage the motion plan. Consider a time interval $[t_{s-1}, t_s], s \in \mathcal{I}_m$. OMP_CHOP guarantees payloads' paths are separated by at least $2r_Q$. Let l_s^i, l_s^j denote the paths of loads i and j , respectively. These line segments are nonempty convex sets in \mathbb{R}^3 , thus, by the Separating Hyperplane Theorem [4], there exists a hyperplane separating them. If each quadrotor-and-payload remained on its respective side of this hyperplane, r_Q away from the boundary, it is guaranteed to be safe. The intersection of hyperplanes separating l_s^i from neighbors l_s^j defines a convex region, $\mathcal{P}_s^i = \{\mathbf{A}_s^{ij} \mathbf{x}_L \leq \mathbf{b}_s^{ij} - r_Q \mid \forall j \in \mathcal{I}_N \neq i\}$, within which the system is free to move during $[t_{s-1}, t_s]$.

Fig. 2a illustrates the construction of \mathcal{P}_0 for the blue robot using the motion plan in Fig. 1. Fig. 2b illustrates $\mathcal{P}_1^{\text{blue}}$. Note the convex regions overlap, and a

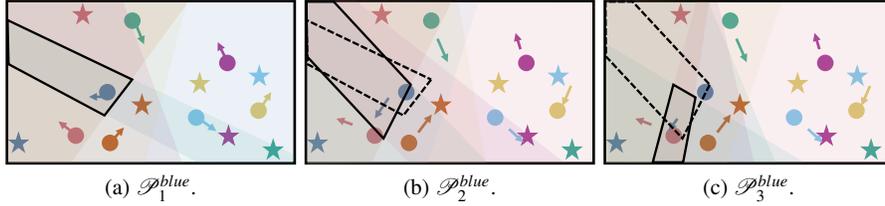


Fig. 2: Convex corridor in the first three time intervals for the blue robot.

continuous piecewise-polynomial trajectory can simultaneously be within regions \mathcal{P}_s^i and \mathcal{P}_{s+1}^i at breaktime t_s . We refer to the series of overlapping convex regions as a *safe corridor*. Note that because the motion plan solved a two-dimensional problem, all convex regions are aligned with \mathbf{e}_z .

To impose safety constraints, we constrain each robot to remain within its safe corridor. We can restate the safety constraints as:

$$\forall i \in \mathcal{I}_N, s \in \mathcal{I}_m, t \in [t_{s-1}, t_s], \mathcal{Q}(\gamma_s^i(t)) \in \mathcal{P}_s^i. \quad (10)$$

Note that we no longer require robots to pass exactly through their motion plan waypoints, allowing trajectories to be significantly shorter than the motion plan paths. Algorithm 1 presents the full trajectory generation algorithm.

Algorithm 1 $\gamma = \text{MPP Solve}(\mathbf{s}, \mathbf{g}, r_L, r_Q, l, \delta_{\max})$

- 1: $\mathcal{M}(t) := \text{Find Motion Plan}(\mathbf{s}, \mathbf{g}, r_Q, l, \delta_{1, \max})$
 - 2: $(\gamma_{\text{nom}}(t), \mathcal{T}_{\text{nom}}) := \text{Find Nominal Trajectories}(\mathcal{M}(t), \delta_{\max})$
 - 3: $\gamma_{\text{smooth}}(t) := \text{Find Initial Smooth Trajectories}(\mathcal{M}(t), \gamma_{\text{nom}}(t), \mathcal{T}_{\text{nom}})$
 - 4: $\alpha := \text{Bisection Search for Min Safe Time}(\gamma_{\text{smooth}}(t), \delta_{\max})$
 - 5: **if** α found **then**
 - 6: $\gamma(t) = \gamma_{\text{smooth}}(t)$ with $\mathcal{T} = \alpha \mathcal{T}_{\text{nom}}$
 - 7: **else**
 - 8: $\gamma(t) = \gamma_{\text{nom}}(t)$
 - 9: **end if**
-

Line 1, detailed in Algorithm 2, identifies a set of *nominal trajectories*, $\gamma_{\text{nom}}(t)$, that indirectly satisfy these convex region constraints by following the motion plan paths exactly. This trajectory can be found analytically as:

$$\gamma_s^i(t) = \mathbf{x}_{d,s-1}^i \left(1 - \beta \left(\frac{t}{t_s - t_{s-1}} \right) \right) + \mathbf{x}_{d,s}^i \beta \left(\frac{t}{t_s - t_{s-1}} \right), \quad (11)$$

where $\beta(\tau) : [0, 1] \rightarrow [0, 1]$ is the 11th-order *non-dimensionalized* polynomial satisfying the boundary conditions:

$$\beta(0) = 0, \quad \beta(1) = 1, \quad \forall k \in \mathcal{I}_5, \beta^{(k)}(0) = 0, \beta^{(k)}(1) = 0. \quad (12)$$

It is clear from the construction of Eq. 11 that each trajectory segment is a straight-line path with a non-constant velocity parameterization.

Algorithm 2 ($\gamma_{nom}(t), \mathcal{T}_{nom}$) = Find Nominal Trajectories($\mathcal{M}(t), v_{max}$)

```

1:  $\beta(t) :=$  solution to Eq. 12
2: for  $s \in \mathcal{S}_m$  do
3:    $\Delta t := \max_{k \in \mathcal{S}_K} \left( \frac{\max_{i \in \mathcal{S}_N} \|\mathbf{x}_{d,s}^i - \mathbf{x}_{d,s-1}^i\|_2}{\delta_{k,max}} \beta_{max}^{(k)}(\tau) \right)^{\frac{1}{k}}$ 
4:   for  $i \in \mathcal{S}_N$  do
5:      $\gamma_{nom,s}^i(t) := (1 - \beta(\frac{t}{\Delta t}))\mathbf{x}_{d,s-1}^i + \beta(\frac{t}{\Delta t})\mathbf{x}_{d,s}^i$ 
6:      $t_{nom,s} := t_{nom,s-1} + \Delta t$ 
7:   end for
8: end for
9:  $\mathcal{T}_{nom} := \{t_{nom,0}, t_{nom,1}, \dots, t_{nom,m}\}$ 

```

The waypoint and breaktime values chosen by the motion plan can result in a nominal trajectory that violates derivative constraints. To address this, we express each trajectory segment's higher derivatives as:

$$\frac{d^k}{dt^k} \gamma^i(t) = \frac{\mathbf{x}_{d,s}^i - \mathbf{x}_{d,s-1}^i}{(t_s - t_{s-1})^k} \beta^{(k)}(\tau).$$

Since $\beta(\tau)$ is known exactly:

$$\frac{d^k}{dt^k} \gamma_{max}^i(t) = \frac{\mathbf{x}_{d,s}^i - \mathbf{x}_{d,s-1}^i}{(t_s - t_{s-1})^k} \beta_{max}^{(k)}(\tau).$$

We can analytically find the minimum time, $t_s - t_{s-1}$, required for all robots to traverse their paths while respecting derivative limitations (Line 3 of Algorithm 2).



Fig. 3: System motion along nominal trajectory segment.

We construct a nominal trajectory for each robot using Eq. 11. Fig. 3 illustrates the resulting system motion along a single segment. Note that while the payload travels along a straight-line path, unlike a kinematic approximation, the system still exhibits dynamic motions. This nominal trajectory is a (sub-optimal) solution to Eqs. 8 and 10 and can always be found.

However, stopping at each trajectory waypoint is inefficient. Thus, using the new breaktimes determined by the nominal trajectory, we construct a QP from Eqs. 8 and 10 in terms of the decision vector \mathbf{d} . This is shown in Line 3 of Algorithm 1 and detailed in Algorithm 3.

Algorithm 3 $\gamma_{smooth}(t) = \text{Find Initial Smooth Trajectories}(\mathcal{M}, \gamma_{nom}(t), \mathcal{T}_{nom})$

```

1:  $Q := \text{Get Cost Function}$ 
2: for  $i \in \mathcal{I}_N$  do
3:   // QP construction.
4:    $\mathbf{A}_{eq}^i, \mathbf{b}_{eq}^i := \text{Get Equality Constraints}$ 
5:    $\mathbf{A}_{ineq}^i, \mathbf{b}_{ineq}^i := \emptyset$ 
6:   for all  $s \in \mathcal{I}_m$  do
7:     for all  $j \in \mathcal{I}_N, i \neq j$  do
8:        $\mathbf{A}_{ineq}^i, \mathbf{b}_{ineq}^i \leftarrow \text{hyperplane constraint, robot } i \text{ to } j, \text{ time } [t_{s-1}, t_s]$ 
9:     end for
10:  end for
11:  // Constraint adjustment for payload safety.
12:   $p_{unsafe} := \mathcal{T}_{nom}$ 
13:  while  $p_{unsafe} \neq \emptyset$  or first iteration do
14:     $\gamma_{smooth}^i(t) := \text{Solve QP}(Q, \mathbf{A}_{eq}, \mathbf{b}_{eq}, \mathbf{A}_{ineq}, \mathbf{b}_{ineq}, \mathcal{T}_{nom})$ 
15:    if QP fails then
16:       $\gamma_{smooth}^i(t) = \gamma_{nom}^i(t)$ 
17:    end if
18:     $p_{unsafe} \leftarrow \text{Find Payload Violations}(\gamma_{smooth}^i, \mathbf{A}_{ineq}, \mathbf{b}_{ineq})$ 
19:  end while
20: end for

```

Lines 1 and 4 construct the quadratic cost and linear continuity and task constraints [21]. Constraining a polynomial to a convex region has been studied extensively in quadrotor planning [5, 9, 15, 20]. We constrain $\gamma_s^i(t)$ by constraining sample points along the segment. Lines 5–10 construct the convex region constraints, which are imposed by sample points at p_{unsafe} . Initially, only segment endpoints are constrained. The QP is solved and the maxima of each trajectory segment is checked. If a maxima is in violation of a constraint, the corresponding time is added as a sample in the next QP formulation (Line 18). This process is repeated until $\gamma_{smooth}^i(t)$ is completely within its safe corridor. [5] show that a finite number of sample points are needed before the $\gamma_{smooth}^i(t)$ is safe, or the QP fails. In the latter case, the nominal trajectory is returned.

At Line 3 of Algorithm 1, the quadrotor can still move outside the designated convex region during highly dynamic maneuvers. Eq. 7 cannot be directly constrained in the form $\mathbf{A}_{ineq}^{ij} \mathbf{x}_Q^i \leq \mathbf{b}_{ineq}^{ij} - r_Q$, as this results in a nonlinear equation with respect to the decision variables. Furthermore, $\gamma_{smooth}(t)$ can contain violations of velocity or higher-derivative constraints.

Note that if $\gamma(t)$ is the optimal solution to a QP with breaktimes \mathcal{T} , the solution to the identical with QP $\tilde{\mathcal{T}} = \alpha \mathcal{T}$ results in a trajectory with derivatives $\tilde{\gamma}^{(k)}(t) = \frac{\gamma^{(k)}(t)}{\alpha^k}$ [17]. By scaling with $\alpha > 1$, we decrease the trajectory’s acceleration, and therefore decrease the relative angle of the quadrotor with respect to the payload, while preserving the trajectory’s path. As safe corridors are axis-aligned, safety constraints will be satisfied for the quadrotor as it approaches the vertical configuration. For $\alpha < 1$, we increase the dynamic motions along the trajectory. Line 4

of Algorithm 1 finds the minimum α to guarantee quadrotor collision avoidance and conformation to derivative constraints using a bisection search.

Algorithm 1 is safe and complete. OMP_CHOP is a safe and complete algorithm. Because the motion plan is disjoint, a safe corridor can be constructed for each robot, and any trajectory in this corridor is safe. By construction, the nominal motion plan will always be within the safe corridor and can always be found.

An important property of Algorithm 1 is that smooth trajectories can be found by solving independent QPs for each robot, eliminating the need to solve a joint optimization problem over all robots' trajectory coefficients. Within each designated safe corridor, each QP will find a locally optimal trajectory, which we see performs well in practice.

5 Experimental Results

We validate our algorithm on an experimental testbed with nine quadrotors, pictured in Fig. 4b. We use the Hummingbird quadrotor from Ascending Technologies¹. Each robot has a diameter of 54 cm and mass of 600 g (with battery), the payloads each have a mass of 80 g, and the cable connections are 70 cm.

Fig. 5 illustrates our experimental architecture. The proposed MPP algorithm (“Multi-robot Trajectory Generator”) and the related control modules are implemented in C++/ROS and run on a 2.5 GHz Intel Core i7 Macbook Pro. We use Gurobi² to solve the trajectory optimization. As our primary motivation is validation of dynamic feasibility, we do not perform explicit feedback control on the payload position. Instead, for each vehicle, we derive the corresponding desired quadrotor states using Eq. 7 and its derivatives (“Quadrotor Trajectory Tracker”). If the planned trajectories are subsequently tracked by the payloads, then they reflect the system's natural dynamics. We control the quadrotor using a hierarchical geometric controller [13]. An outer loop (“Quadrotor Position Controller”) calculates

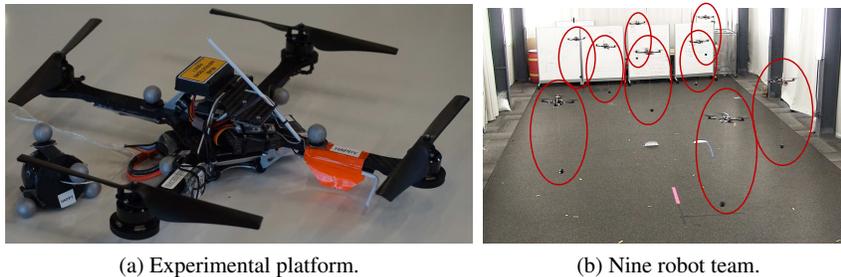


Fig. 4: Asctec Hummingbird quadrotors with cable-suspended payloads.

¹ <http://www.asctec.de/en/>

² www.gurobi.com/

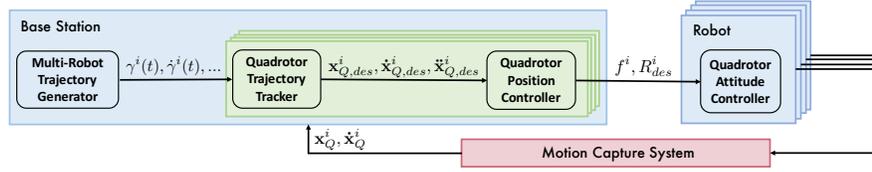


Fig. 5: Experimental system architecture.

a desired thrust and orientation for each robot, which is then sent via Xbee³ to the proper vehicle. An onboard attitude controller tracks this desired orientation. We use a Vicon⁴ motion capture system to close the control loop at 100 Hz. Video footage can be found at <https://youtu.be/EGgRPYbTRig>.

Fig. 6 displays tracking results from a two-robot problem. Fig. 6a displays each quadrotor and payload's trajectory, where planned payload trajectories are pictured in black, actual payload trajectories are pictured in solid, and actual quadrotor trajectories are pictured in dashed lines. We see that the payload is allowed to swing outwards. Fig. 6b plots the payload angle with respect to the vertical:

$$\phi_x = \tan^{-1} \left(\frac{\mathbf{p}_y}{-\mathbf{p}_z} \right) \quad \phi_y = -\tan^{-1} \left(\frac{\mathbf{p}_x}{-\mathbf{p}_z} \right).$$

While load angles are generally tracked adequately, there is an increase in error as payloads swing higher. Figs. 6c–6d plot the planar load and quadrotor velocities.

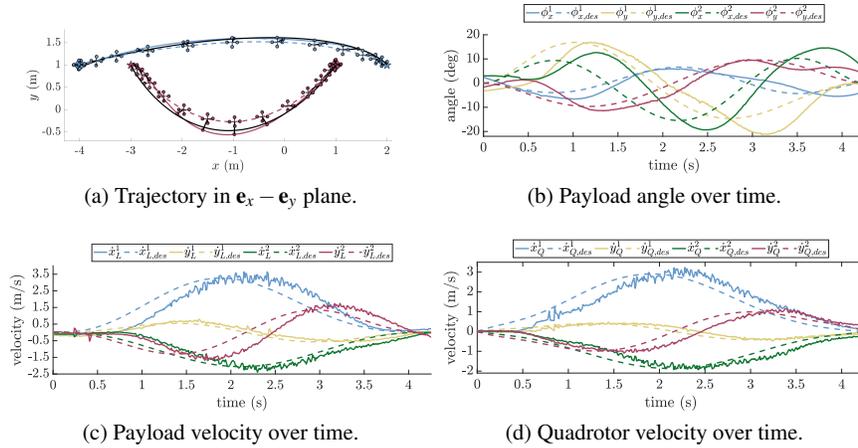


Fig. 6: Trajectory execution data by a two-robot team. Each payload begins at positions denoted with circles and must navigate to stars of the same color.

³ <https://www.digi.com/xbee>

⁴ <https://www.vicon.com/>

Table 2: Performance statistics.

N	Straight-line (m)	$ \mathbf{x}_Q(t) $ (m)	$ \mathbf{x}_L(t) $ (m)	Max. $\ \dot{\mathbf{x}}_Q\ _2$ (m/s)	Max. $\ \dot{\mathbf{x}}_L\ _2$ (m/s)	Max. ϕ_L (deg)
2	6.00	6.10	6.36	3.24	3.65	21.31
2	4.00	4.82	5.37	2.03	2.54	19.35
6	3.00	5.10	6.09	2.53	2.67	36.86
6	3.00	4.93	5.68	1.83	3.37	37.18
6	3.00	4.96	5.52	2.03	2.53	25.06
9	3.00	5.29	6.00	1.64	2.08	32.07
9	3.00	6.91	7.84	2.30	2.87	39.14
9	3.00	6.89	8.07	2.24	3.12	36.00

Table 2 lists relevant performance statistics. Here, N represents the number of robots. “Straight-line” refers to the Euclidean distance between the robot’s start and goal. $|\mathbf{x}_Q(t)|$ refers to the distance traveled by the quadrotor while $|\mathbf{x}_L(t)|$ refers to the distance traveled by the payload. The maximum velocities of the payload and quadrotor, as well as the maximum payload angle, are also reported.

For the two-robot maneuver, the first robot’s payload, corresponding to the blue robot in Fig. 6a, is able to safely reach its goal with the payload and the quadrotor traveling only slightly further than the straight-line distance. Because of the time-parameterized convex-region constraints, the second robot, corresponding to the red robot in Fig. 6a, experiences a larger increase in distance traveled from the straight-line distance. However, in both cases, the dynamic maneuver decreases the distance traveled by the quadrotor relative to that by the payload. The payload reaches a maximum velocity of 3.65 m/s and a maximum angle of 21.31 deg from the vertical.

Fig. 7 illustrates solution trajectories for larger teams, displaying trajectories from three representative robots. Fig. 7a shows a six-robot maneuver where payloads begin in a circle and are tasked to move to antipodal points. All robots enter a single circular holding pattern. Figs. 7c and 7d plot position and velocity tracking for the load over time. The quadrotor tracking data is similar. The nine-robot problem displayed in Fig. 7b illustrates more localized collision-avoidance, where the solution splits robots into two separate holding patterns. Robots’ payloads are still able to swing outwards while maintaining safety between robots in different holding patterns. Figs. 7e and 7f plots the payload tracking data over time. In particular, note that the yellow and red robots are allowed to reach their goals before the navy robot.

Table 2 reports the statistics for the three representative robots from each problem. As expected, as the size of the team and holding patterns increase, the identified safe trajectories become increasingly longer than the straight-line distances. However, the payload is still able to similarly swing, reaching angles of almost 40 deg.

By decomposing the problem in a motion planning, trajectory generation, and time refinement steps, we can find solutions to a complex planning problem. We note that this general paradigm could be useful to other complex mechanical systems as well, as any method for discrete motion planning or generating single-robot trajectories through convex regions can be easily incorporated.

There are a number of algorithm improvements to be explored. The identified solution is sub-optimal, and in dense scenarios, the smooth QP optimization often fails, forcing robots to take nominal trajectories along a potentially large holding pattern. A related issue is the QPs' numerical stability. The existence of many neighbors increases the number of trajectory segments, and therefore the decision vector size, and the number of inequality constraints. Furthermore, the higher-order robot dynamics calls for a higher-order trajectory polynomial, leading to extremely large values in the cost and equality constraint matrices. To mitigate this, we nondimensionalize the trajectory [17] and find that in practice, minimum-jerk (minimizing the third derivative in the cost function), 5th-order polynomials are often sufficient for good performance. We hope to further study these challenges in future work.

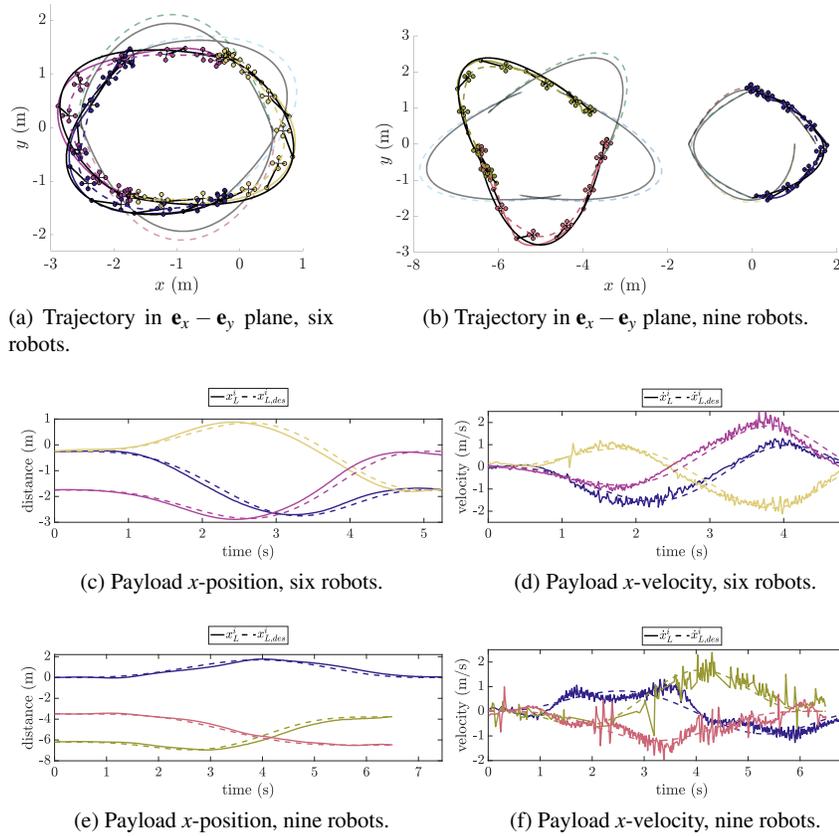


Fig. 7: Trajectory execution data by three representative robots in larger robot teams.

6 Conclusions

We present a safe and complete algorithm to solve the labeled MPP for a team of aerial robots carrying cable-suspended payloads. We synthesize trajectories by decoupling the motion planning and trajectory generation steps, a general approach that can also be used for other complex dynamical systems. We demonstrate our algorithm's usability with dynamic maneuvers on a nine-robot testbed. In future work, we hope to improve the algorithm's computational performance and solution quality, particularly on dense problem geometries, as well as numerical stability for scalability to larger teams. We hope that this work will help set the stage for future research towards dynamic multi-robot teams.

7 Acknowledgements

We gratefully acknowledge the support of ONR grants N00014-09-1-1051 and N00014-09-1-103, NSF grant IIS-1426840, ARL grant W911NF-08-2-0004. Sarah Tang is supported by NSF Research Fellowship Grant No. DGE-1321851. The authors thank Jeremy Wang for the fabrication of the payload suspension attachments.

References

1. Augugliaro F., Lupashin S., Hamer M., Male C., Hehn M., Mueller M. W., Willmann J. S., Gramazio F., Kohler M., and D'Andrea R., "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems*, vol. 34, no. 4, pp. 46–64, Aug 2014.
2. Bennewitz M., Burgard W., and Thrun S., "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," *Robotics and Autonomous Systems*, vol. 41, no. 2, pp. 89–99, 2002.
3. Bernard M. and Kondak K., "Generic slung load transportation system using small size helicopters," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.
4. Boyd S. and Vandenberghe L., *Convex Optimization*. Cambridge, United Kingdom: Cambridge University Press, 2004, ch. 2, pp. 21–66.
5. Chen J., Liu T., and Shen S., "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. New York: Institute of Electrical and Electronics Engineers (IEEE), May 16–21 2016, pp. 1476–83.
6. Faust A., Palunko I., Cruz P., Fierro R., and Tapia L., "Automated aerial suspended cargo delivery through reinforcement learning," *Artificial Intelligence*, vol. 247, pp. 381–398, 2017, special Issue on AI and Robotics.
7. Flores M. E., "Real-time trajectory generation for constrained nonlinear dynamical systems using non-uniform rational b-spline basis functions," Ph.D. dissertation, California Institute of Technology, 2008, ph.D.thesis.
8. Foehn P., Falanga D., Kuppuswamy N., Tedrake R., and Scaramuzza D., "Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload," in *Robotics: Science and Systems (RSS)*, Boston, July 12–16 2017.
9. Gao F. and Shen S., "Online quadrotor trajectory generation and autonomous navigation on point clouds," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, New York, Oct. 23–27 2016, pp. 139–46.
10. Goldenberg M., Felner A., Stern R., Sharon G., Sturtevant N., Holte R. C., and Schaeffer J., "Enhanced partial expansion A*," *Journal of Artificial Intelligence Research*, vol. 50, no. 1, pp. 141–187, 2014.

11. Han S. D., Rodriguez E. J., and Yu J., "Sear: A polynomial-time expected constant-factor optimal algorithmic framework for multi-robot path planning," *arXiv*, 2017, <https://arxiv.org/abs/1709.08215>.
12. J.Alonso-Mora , Naegeli T., Siegwart R., and Beardsley P., "Collision avoidance for aerial vehicles in multi-agent scenarios," *Autonomous Robots*, vol. 39, no. 1, pp. 101–121, Jun 2015.
13. Lee T., Leok M., and McClamroch N. H., "Control of complex maneuvers for a quadrotor UAV using geometric methods on SE(3)," *Asian Journal of Control*, vol. 15, no. 2, pp. 391–408, 2011.
14. Lindsey Q., Mellinger D., and Kumar V., "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, no. 3, pp. 323–336, 2012.
15. Liu S., Watterson M., Mohta K., Sun K., Bhattacharya S., Taylor C. J., and Kumar V., "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 3, 2017.
16. Luna R. and Bekris K. E., "Push and swap: Fast cooperative path-finding with completeness guarantees," in *International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain, July 2011.
17. Mellinger D. and Kumar V., "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, New York, May 9–13 2011, pp. 2520–25.
18. Murray R. M., Rathinam M., and Sluis W., "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME International Congress and Exposition*, 1995.
19. Peng J. and Akella S., "Coordinating multiple robots with kinodynamic constraints along specified paths," *The International Journal of Robotics Research (IJRR)*, vol. 24, no. 4, pp. 295–310, 2005.
20. Preiss J. A., Hönig W., Ayanian N., and Sukhatme G. S., "Downwash-aware trajectory planning for large quadrotor teams," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
21. Richter C., Bry A., and Roy N., "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research: The 16th International Symposium ISRR*, Inaba M. and Corke P., Eds. Cham: Springer International Publishing, 2016, vol. 1, pp. 649–66.
22. Solovey K., Salzman O., and Halperin D., "Finding a needle in an exponential haystack: Discrete rrt for exploration of implicit roadmaps in multi-robot motion planning," *The International Journal of Robotics Research (IJRR)*, vol. 35, no. 5, pp. 501–513, 2016.
23. Spirakis P. and Yap C. K., "Strong NP-hardness of moving many discs," *Information Processing Letters*, vol. 19, no. 1, pp. 55–59, 1984.
24. Sreenath K. and Kumar V., "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems (RSS)*, 2013.
25. Sreenath K., Lee T., and Kumar V., "Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load," in *IEEE Conference on Decision and Control (CDC)*, 2013.
26. Standley T., "Finding optimal solutions to cooperative pathfinding problems," in *AAAI Conference on Artificial Intelligence*, Atlanta, GA USA, July 2010.
27. Tang S. and Kumar V., "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *IEEE International Conference on Robotics and Automation (ICRA)*, New York, May 26–30 2015, pp. 2215 – 22.
28. —, "A complete algorithm for generating safe trajectories for multi-robot teams," in *Robotics Research: Volume 2*, Bicchi A. and Burgard W., Eds. Cham: Springer International Publishing, 2018, pp. 599–616.
29. Tang S., Thomas J., and Kumar V., "Hold or take optimal plan (hoop): A quadratic programming approach to multi-robot trajectory generation," *The International Journal of Robotics Research (IJRR)*, 2018, in press.
30. Yu J. and Rus D., *An Effective Algorithmic Framework for Near Optimal Multi-robot Path Planning*. Cham: Springer International Publishing, 2018, pp. 495–511.