Engineering Notes

Minimum-Time Trajectory Planning for Multi-Unmanned-Aerial-Vehicle Cooperation Using Sequential Convex Programming

Zhu Wang,* Li Liu,† and Teng Long‡
Beijing Institute of Technology, 100081 Beijing, People's Republic of China

DOI: 10.2514/1.G002349

I. Introduction

THE cooperation of unmanned aerial vehicles (UAVs) has been investigated in many applications, such as wide area search, persistent surveillance, and forest firefighting [1,2]. Trajectory planning is an important technique to achieve UAVs' cooperation. Typical trajectory-planning methods include mixed-integer programming [3], rapidly exploring random tree [4], nonlinear programming [5], and heuristic-based intelligent algorithms [6]. However, these methods scale poorly to the number of UAVs due to their computational complexity. Recently, convex programming has gained increasing applications in trajectory planning, because convex problems can be efficiently solved even for high-dimensional problems [7]. Application examples include planetary landing [8], spacecraft coordination [9], and formation reconfiguration [10]. However, simple convex programming may be trapped in infeasible solutions for feasible problems due to over conservative approximation of nonconvex constraints.

To alleviate the conservativeness by single iteration of convex programming [11], sequential convex programming (SCP) is investigated. Through successive convexification, SCP iteratively solves a sequence of convex-programming subproblems to obtain a local optimal solution. For single-vehicle trajectory planning, SCP was successfully applied to fixed-time trajectory planning with highly nonlinear dynamics, such as spacecraft rendezvous and proximity operations [12], and entry flight [13]. Also, SCP was extended to free-final-time trajectory planning of fuel-optimal powered landing [14].

The first application of SCP to cooperative trajectory planning was conducted by Morgan et al. [11]. They provided a centralized SCP and two partially decentralized SCP methods for swarm reconfiguration of spacecraft with nonlinear dynamics. Then, model predictive control (MPC) was introduced into SCP [15] to reduce the computational cost. This MPC/SCP-based trajectory-generation method was further integrated with distributed auction assignment [16] to realize the reconfiguration of a variable swarm. For SCP-based UAV trajectory planning, collision-free trajectories were first generated in [17] for a quadrocopter formation with linear dynamics constraints. Based on this work, Chen et al. [18] presented an incremental SCP (iSCP), which tightened constraints incrementally to improve the feasibility of intermediate subproblems. Meanwhile, they developed a decoupled iSCP, which computes each UAV's trajectory sequentially to achieve better computational tractability.

Nevertheless, the flight time was assumed to be predefined in the aforementioned SCP-based cooperative trajectory planning. And few studies on parallel implementation of free-final-time cooperative planning were reported. To address these limitations, this work extends the coupled SCP to solve the minimum-time cooperative trajectory planning of UAVs with nonlinear dynamics. And a decoupled SCP supporting parallel planning and ensuring the time consensus of different UAVs is proposed. Finally, numerical simulations demonstrate that the SCP approaches can provide superior efficiency for cooperative trajectory planning, and the decoupled SCP scales well to the number of UAVs.

II. Problem Formulation

In this section, the minimum-time cooperative trajectory-planning problem formulation for multiple UAVs is presented. The problem is cast into a nonconvex optimal-control problem, which is then parameterized to be a nonconvex optimization problem through discretization of the states and controls.

It is assumed that N homogeneous UAVs cooperatively perform tasks in a three-dimensional space. And the dynamics of UAVs [19,20] are described by the following simplified point-mass equations of motion:

$$\begin{aligned}\dot{x}_i &= V_i \cos\gamma_i \cos\chi_i \\ \dot{y}_i &= V_i \cos\gamma_i \sin\chi_i \\ \dot{h}_i &= V_i \sin\gamma_i \\ \dot{V}_i &= g\cdot(n_{x,i} - \sin\gamma_i) \\ \dot{\chi}_i &= g\cdot n_{y,i}/(V_i\cos\gamma_i) \\ \dot{\gamma}_i &= g\cdot(n_{z,i} - \cos\gamma_i)/V_i\end{aligned} \tag{1}$$

in which $i = 1, 2, \ldots, N$ is the index of the UAV; x is the downrange; y is the crossrange; h is the altitude; V is the velocity; χ is the heading angle; γ is the flight-path angle; n_x is the tangential load factor; n_y is the horizontal load factor; n_z is the vertical load factor; and g is the constant acceleration of gravity. The states and controls of UAV i are denoted as $\boldsymbol{s}_i = (x_i, y_i, h_i, V_i, \chi_i, \gamma_i)^T$ and $\boldsymbol{u}_i = (n_{x,i}, n_{y,i}, n_{z,i})^T$, respectively.

Other constraints for multi-UAV trajectory planning considered in this work include initial state and final state conditions in Eq. (2), bounds on states and controls in Eq. (3), obstacle avoidance in Eq. (4), and inter-UAV collision avoidance in Eq. (5). Note that the obstacles are considered as cylinders, as given in Eq. (4).

$$\boldsymbol{s}_i(t_0) = \boldsymbol{s}_{i,0}, \qquad \boldsymbol{s}_i(t_f) = \boldsymbol{s}_{i,f} \tag{2}$$

$$\boldsymbol{s}_{\min} \le \boldsymbol{s}_i(t) \le \boldsymbol{s}_{\max}, \qquad \boldsymbol{u}_{\min} \le \boldsymbol{u}_i(t) \le \boldsymbol{u}_{\max} \tag{3}$$

$$\|\boldsymbol{C}\cdot[\boldsymbol{s}_i(t) - \boldsymbol{p}_{\text{obs},m}]\|_2 \ge r_m, \quad i = 1,2,\ldots,N, \quad m = 1,2,\ldots,M \tag{4}$$

Received 15 July 2016; revision received 10 March 2017; accepted for publication 17 June 2017; published online 10 August 2017.

*Postdoctor, School of Aerospace Engineering; also Key Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education; wangzhubit@163.com.

†Professor, School of Aerospace Engineering; also Key Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education; liuli@bit.edu.cn.

‡Professor, School of Aerospace Engineering; also Key Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education; tenglong@bit.edu.cn (Corresponding Author).

$$\|\boldsymbol{C}\cdot[\boldsymbol{s}_i(t)-\boldsymbol{s}_j(t)]\|_2 \geq R,\quad i=1,2,\ldots,N,\quad j=1,2,\ldots,N, i\neq j \tag{5}$$

in which t_0 is the initial time, t_f is the final time, M is the number of obstacles, $\boldsymbol{p}_{\text{obs},m}$ is the center of the obstacle, r_m is the radius of the obstacle, R is the minimum safe distance between UAVs, and $\boldsymbol{C} = [\boldsymbol{I}_{2\times 2}, 0_{4\times 4}]$.

Then, the minimum-time trajectory-planning problem for multi-UAV cooperation can be defined as problem 1. This problem is a free-final-time optimal-control problem with nonlinear dynamics and nonconvex path constraints.

Problem 1 (nonconvex optimal control):

$$\min_{\boldsymbol{s}_i(t),\boldsymbol{u}_i(t),t_f,i=1,2,\ldots,N} \int_{t_0}^{t_f} 1\,\mathrm{d}t$$
$$\text{subject to Eqs. (1–5)} \tag{6}$$

To solve the optimal-control problem using SCP, the collocation method [21] is employed to reformulate problem 1 as a parameter-optimization problem. Based on the trapezoidal method, the equations of dynamics in Eq. (1) can be numerically integrated as Eq. (7), in which $\boldsymbol{s}_i[k]$ and $\boldsymbol{u}_i[k]$ are states and controls at the discretized time t_k, Δt is the constant step size, K is the predefined number of intervals, $t_k = t_0 + k\cdot\Delta t$, and $\Delta t = (t_f - t_0)/K$.

$$\begin{aligned}\boldsymbol{s}_i[k+1] = \boldsymbol{s}_i[k] + \Delta t/2\cdot[f(\boldsymbol{s}_i[k],\boldsymbol{u}_i[k])\\ + f(\boldsymbol{s}_i[k+1],\boldsymbol{u}_i[k+1])],\qquad k=0,1,\ldots,K-1\end{aligned} \tag{7}$$

Meanwhile, the initial and final state conditions, bounds on states and controls, obstacle avoidance, and collision-avoidance constraints can be discretized as Eqs. (8–11) by imposing the constraints at the discretized points.

$$\boldsymbol{s}_i[0] = \boldsymbol{s}_{i,0},\qquad \boldsymbol{s}_i[K] = \boldsymbol{s}_{i,f} \tag{8}$$

$$\boldsymbol{s}_{\min}\leq\boldsymbol{s}_i[k]\leq\boldsymbol{s}_{\max},\quad \boldsymbol{u}_{\min}\leq\boldsymbol{u}_i[k]\leq\boldsymbol{u}_{\max},\quad k=0,1,\ldots,K \tag{9}$$

$$\begin{aligned}&\|\boldsymbol{C}\cdot\boldsymbol{s}_i[k]-\boldsymbol{p}_{\text{obs},m}\|_2\geq r_m,\quad i=1,2,\ldots,N,\quad m=1,2,\ldots,M,\\ &k=0,1,\ldots,K\end{aligned} \tag{10}$$

$$\begin{aligned}&\|\boldsymbol{C}\cdot[\boldsymbol{s}_i[k]-\boldsymbol{s}_j[k]]\|_2\geq R,\quad i=1,2,\ldots,N,\quad j=1,2,\ldots,N,\quad i\neq j,\\ &k=0,1,\ldots,K\end{aligned} \tag{11}$$

Because the value of K is specified and the discrete points are uniformly distributed, minimizing the flight time is equal to minimizing the step size of time. Thus, problem 1 can be transcribed as the following optimization problem, in which the optimization variables include the UAVs' states and controls at discrete times and the step size of time.

Problem 2 (nonconvex optimization problem):

$$\min_{\boldsymbol{s}_i(k),\boldsymbol{u}_i(k),\Delta t,i=1,2,\ldots,N,k=0,1,\ldots,K} \Delta t$$
$$\text{subject to Eqs. (7–11)} \tag{12}$$

III. Coupled SCP

SCP is an iterative method by solving a series of convex-programming subproblems. To apply SCP to cooperative trajectory planning, convex subproblems are formulated by successive convexification, and the iterative framework of the coupled SCP is presented in this section.

A. Convex-Programming-Subproblem Formulation

From problem 2, the objective function, initial and final conditions, and bounds on states and controls are already convex. Thus, the constraints of dynamics, obstacle avoidance, and collision avoidance are convexified. The discretized dynamics in Eq. (7) can be linearized about the nominal trajectory ($\boldsymbol{s}_i[k]$, $\boldsymbol{u}_i[k]$, Δt) of each UAV as

$$\begin{aligned}&\boldsymbol{A}_{k+1}\cdot\boldsymbol{s}_i[k+1]+\boldsymbol{A}_k\cdot\boldsymbol{s}_i[k]+\boldsymbol{B}_{k+1}\cdot\boldsymbol{u}_i[k+1]+\boldsymbol{B}_k\cdot\boldsymbol{u}_i[k]\\ &+(\boldsymbol{C}_k+\boldsymbol{C}_{k+1})\cdot\Delta t+\boldsymbol{D}_k+\boldsymbol{D}_{k+1}=0,\ k=0,1,\ldots,K-1\end{aligned} \tag{13}$$

in which the matrices of coefficients $\boldsymbol{A}_k$, $\boldsymbol{A}_{k+1}$, $\boldsymbol{B}_k$, $\boldsymbol{B}_{k+1}$, $\boldsymbol{C}_k$, $\boldsymbol{C}_{k+1}$, $\boldsymbol{D}_k$, and $\boldsymbol{D}_{k+1}$ are given as Eq. (14). The partial derivatives $\partial f(\boldsymbol{s},\boldsymbol{u})/\partial\boldsymbol{s}$ and $\partial f(\boldsymbol{s},\boldsymbol{u})/\partial\boldsymbol{u}$ in Eq. (14) can be derived from Eq. (1).

$$\begin{aligned}&\boldsymbol{A}_k=\frac{\Delta\bar{t}}{2}\cdot\frac{\partial f(\boldsymbol{s}_i,\boldsymbol{u}_i)}{\partial\boldsymbol{s}_i}\bigg|_{(\bar{s}_i[k],\bar{u}_i[k])}+\boldsymbol{I},\quad \boldsymbol{B}_k=\frac{\Delta\bar{t}}{2}\cdot\frac{\partial f(\boldsymbol{s}_i,\boldsymbol{u}_i)}{\partial\boldsymbol{u}_i}\bigg|_{(\bar{s}_i[k],\bar{u}_i[k])},\\ &\boldsymbol{C}_k=\frac{f(\bar{\boldsymbol{s}}_i[k],\bar{\boldsymbol{u}}_i[k])}{2},\boldsymbol{A}_{k+1}=\frac{\Delta\bar{t}}{2}\cdot\frac{\partial f(\boldsymbol{s}_i,\boldsymbol{u}_i)}{\partial\boldsymbol{s}_i}\bigg|_{(\bar{s}_i[k+1],\bar{u}_i[k+1])}-\boldsymbol{I},\\ &\boldsymbol{B}_{k+1}=\frac{\Delta\bar{t}}{2}\cdot\frac{\partial f(\boldsymbol{s}_i,\boldsymbol{u}_i)}{\partial\boldsymbol{u}_i}\bigg|_{(\bar{s}_i[k+1],\bar{u}_i[k+1])},\quad \boldsymbol{C}_{k+1}=\frac{f(\bar{\boldsymbol{s}}_i[k+1],\bar{\boldsymbol{u}}_i[k+1])}{2},\\ &\boldsymbol{D}_k=-\frac{\Delta\bar{t}}{2}\cdot\left(\frac{\partial f(\boldsymbol{s}_i,\boldsymbol{u}_i)}{\partial\boldsymbol{s}_i}\bigg|_{(\bar{s}_i[k],\bar{u}_i[k])}\cdot\bar{\boldsymbol{s}}_i[k]+\frac{\partial f(\boldsymbol{s}_i,\boldsymbol{u}_i)}{\partial\boldsymbol{u}_i}\bigg|_{(\bar{s}_i[k],\bar{u}_i[k])}\cdot\bar{\boldsymbol{u}}_i[k]\right),\\ &\boldsymbol{D}_{k+1}=-\frac{\Delta\bar{t}}{2}\cdot\left(\frac{\partial f(\boldsymbol{s}_i,\boldsymbol{u}_i)}{\partial\boldsymbol{s}_i}\bigg|_{(\bar{s}_i[k+1],\bar{u}_i[k+1])}\cdot\bar{\boldsymbol{s}}_i[k+1]\right.\\ &\left.+\frac{\partial f(\boldsymbol{s}_i,\boldsymbol{u}_i)}{\partial\boldsymbol{u}_i}\bigg|_{(\bar{s}_i[k+1],\bar{u}_i[k+1])}\cdot\bar{\boldsymbol{u}}_i[k+1]\right)\end{aligned} \tag{14}$$

According to [15,18], the constraints of cylinder obstacle avoidance and inter-UAV collision avoidance can be convexified as Eqs. (15) and (16), respectively. The original concave feasible region of obstacle-avoidance constraints is convexified to be a half-plane, as defined in Eq. (15). And the original inter-UAV collision-avoidance constraints are linearized to ensure that any two UAVs are not simultaneously located inside a safe band [18].

$$\begin{aligned}&\|\boldsymbol{C}\cdot\bar{\boldsymbol{s}}_i[k]-\boldsymbol{p}_{\text{obs},m}\|+\frac{(\boldsymbol{C}\cdot\bar{\boldsymbol{s}}_i[k]-\boldsymbol{p}_{\text{obs},m})^T}{\|\boldsymbol{C}\cdot\bar{\boldsymbol{s}}_i[k]-\boldsymbol{p}_{\text{obs},m}\|}\cdot(\boldsymbol{C}\cdot\boldsymbol{s}_i[k]-\boldsymbol{C}\cdot\bar{\boldsymbol{s}}_i[k])\geq r_m,\\ &i=1,2,\ldots,N,\quad m=1,2,\ldots,M,\quad k=0,1,\ldots,K\end{aligned} \tag{15}$$

$$\begin{aligned}&(\bar{\boldsymbol{s}}_i[k]-\bar{\boldsymbol{s}}_j[k])^T\cdot\boldsymbol{C}^T\boldsymbol{C}\cdot(\boldsymbol{s}_i[k]-\boldsymbol{s}_j[k])\geq R\cdot\|\boldsymbol{C}(\bar{\boldsymbol{s}}_i[k]-\bar{\boldsymbol{s}}_j[k])\|_2,\quad i\neq j,\\ &i=1,2,\ldots,N,\quad j=1,2,\ldots,N,\quad k=0,1,\ldots,K\end{aligned} \tag{16}$$

With the convex approximations of all the nonconvex constraints, the convex subproblem can be constructed. The $L-1$ penalty method [13] is used to handle the constraints, and the trust region [15] is defined to ensure the acceptable accuracy of the convexified terms and the good convergence property of SCP. Thus, the coupled convex-programming subproblem is constructed as follows.

Problem 3 (coupled convex subproblem):

$$\begin{aligned}&\min_{\boldsymbol{s}_i[k],\boldsymbol{u}_i[k],\Delta t}\ \Delta t+\mu\cdot\left(\sum_{l=1}^{n_{\text{eq}}}|h_l(\boldsymbol{Z})|+\sum_{l=1}^{n_{\text{ineq}}}\max(g_l(\boldsymbol{Z}),0)\right)\\ &\text{subject to}\ |\boldsymbol{s}_i[k]-\bar{\boldsymbol{s}}_i[k]|\leq\boldsymbol{\rho},\ i=1,2,\ldots,N,\ k=0,1,\ldots,K\end{aligned} \tag{17}$$

in which μ is the penalty parameter; $\boldsymbol{\rho}\in\boldsymbol{R}^6$ is the size of the trust region; equality constraints h_l include linear constraints in Eq. (13); inequality constraints g_l include convex constraints in Eqs. (8), (9), (15), and (16); and optimization variables $\boldsymbol{Z}$ include states and controls $\boldsymbol{z}_i$ of each UAV and time-step size Δt, as shown in the following:

$$\begin{aligned}&\boldsymbol{Z}=(\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_N,\Delta t)^T,\quad \boldsymbol{z}_i=(\boldsymbol{s}_i[0],\boldsymbol{s}_i[1],\ldots,\boldsymbol{s}_i[K],\boldsymbol{u}_i[0],\\ &\ldots,\boldsymbol{u}_i[1],\ldots,\boldsymbol{u}_i[K]),\quad i=1,2,\ldots,N\end{aligned} \tag{18}$$

B. Coupled SCP-Based Cooperative Trajectory Planning

The procedures of the coupled SCP-based cooperative trajectory planning are shown in Algorithm 1. Coupled method here means only one coupled subproblem is solved at each iteration. On the contrary, the decoupled SCP needs to solve N decoupled subproblems at each iteration, as presented in Sec. IV.

In Algorithm 1, a two-stage optimization strategy is applied to alleviate the sensitivity of SCP to initial guesses. In the first stage (lines 1–4), the optimization is solved without considering obstacle-avoidance and inter-UAV collision-avoidance constraints. An initial guess $\boldsymbol{Z}^0$ is generated by linear interpolating between the initial conditions and the final conditions of the UAVs. The iterations of the first stage continue until the solution satisfies the initial and final conditions in Eq. (8), bounds on states and controls in Eq. (9), and relaxed dynamics constraints in Eq. (19).

$$|\boldsymbol{s}_i^q[k+1]-\boldsymbol{s}_i^q[k]-\Delta t^q/2\cdot(f(\boldsymbol{s}_i^q[k],\boldsymbol{u}_i^q[k]) + f(\boldsymbol{s}_i^q[k+1],\boldsymbol{u}_i^q[k+1]))|\le\tau,\quad k=0,1,\ldots,K-1 \tag{19}$$

in which q denotes the index of iteration, and τ is a convergence tolerance.

In the second stage (lines 5–9), the complete subproblem (problem 3) is iteratively solved. The iterations of the second stage are stopped when all of the trajectory constraints and the convergence conditions, Eq. (20), are satisfied.

$$|\boldsymbol{s}_i^q[k]-\boldsymbol{s}_i^{q-1}[k]|\le\boldsymbol{\varepsilon},\quad i=1,\ldots,N,\quad k=0,1,\ldots,K \tag{20}$$

in which $\boldsymbol{\varepsilon}$ is a tolerance vector, and the inequality is applied component wise.

IV. Decoupled SCP

In this section, a decoupled SCP method for cooperative trajectory planning is proposed. By decomposition of collision avoidance and consideration of the time consensus, the decoupled subproblems for UAVs are formulated and are solved in parallel to obtain cooperative trajectories efficiently.

A. Decoupled Convex-Programming Subproblem

In the decoupled SCP, the coupled convex subproblem is decomposed into N smaller problems to obtain each UAV's trajectory in parallel. From problem 3, only collision-avoidance constraints involve two UAVs' trajectories. According to the freezing idea [15], the inter-UAV collision constraints can be decoupled as Eq. (21), in which each UAV avoids other UAVs' nominal trajectories. Because the optimized trajectories are identical to the nominal trajectories when SCP iterations converge, the feasible solutions under the decoupled collision constraints also satisfy the coupled constraints.

$$(\bar{\boldsymbol{s}}_i[k]-\bar{\boldsymbol{s}}_j[k])^T\cdot\boldsymbol{C}^T\boldsymbol{C}\cdot(\boldsymbol{s}_i[k]-\boldsymbol{s}_j[k])\ge R\cdot\|\boldsymbol{C}(\bar{\boldsymbol{s}}_i[k]-\bar{\boldsymbol{s}}_j[k])\|_2,\quad j\in\Pi_i,\qquad k=0,1,\ldots,K \tag{21}$$

in which Π_i is the set of UAVs avoided by UAV i. To balance the computational complexity for different UAVs, the collision constraints are distributed to UAVs in this work. The assignment rule is determined as follows:

$$\Pi_i=(i,i+1,\ldots,i+n_{c,i}-1)\%N+1,\quad i=1,2,\ldots,N,\qquad n_{c,i}=\begin{cases}(N-1)/2, & N\text{is odd}\\ N/2, & N\text{is even},i\le N/2\\ N/2-1, & N\text{is even},i>N/2\end{cases} \tag{22}$$

in which $n_{c,i}$ is the number of UAVs to be avoided by the i-th UAV, and % denotes the remainder operator.

Because the trajectory of each UAV is optimized independently at each iteration of the decoupled SCP, the step size of different UAVs may not be identical. To ensure the consensus of the flight time of different UAVs, the following condition must be satisfied when the decoupled SCP converges:

$$\max_{i=1,2,\ldots,N}\Delta t_i^q-\min_{i=1,2,\ldots,N}\Delta t_i^q\le\varepsilon_t \tag{23}$$

in which ε_t is a specified tolerance of the flight time. To acquire an identical flight time of different UAVs, the following lower bound on step size is introduced into the decoupled subproblems. By adding these constraints, the flight time of early-arriving UAVs is increased to coordinate with slower UAVs:

$$\Delta t_i^q\ge\Delta t_{\min}^q,\quad i=1,2,\ldots,N,\qquad \Delta t_{\min}^q=(\max_{j\in\Omega}(\Delta\bar{t}_j^q)+\min_{j\in\Omega}(\Delta\bar{t}_j^q))/2,\qquad \Omega=\left\{j\middle||\Delta\bar{t}_j^q-\Delta\bar{t}_j^{q-1}|/\Delta\bar{t}_j^{q-1}\le\varepsilon_{\Delta t}\right\} \tag{24}$$

in which $\Delta t_{\min}^q$ is the lower bound on step size, and $\Delta\bar{t}_j^q$ is the step size of the nominal trajectory. When calculating the lower bound on step size, only the UAVs whose relative variation of the nominal step size is less than a specified error $\varepsilon_{\Delta t}$ are considered, as shown in Eq. (24).

Based on the decomposition of optimization variables and the decentralization of coupled constraints, the decoupled subproblem of each UAV can be formulated as follows:

Problem 4 (decoupled convex subproblem for UAV i):

$$\min_{\boldsymbol{Z}_i}\ \Delta t_i+\mu\cdot\left(\sum_{l=1}^{n'_{\text{eq}}}|h'_l(\boldsymbol{Z}_i)|+\sum_{l=1}^{n'_{\text{ineq}}}\max(g'_l(\boldsymbol{Z}_i),0)\right) \tag{25}$$

$$\text{subject to }|\boldsymbol{s}_i[k]-\bar{\boldsymbol{s}}_i[k]|\le\boldsymbol{\rho},\qquad k=0,1,\ldots,K$$

in which equality constraints h'_l include linear constraints in Eq. (13); inequality constraints g'_l include convex constraints in Eqs. (8), (9), (15), (21), and (24); optimization variables $\boldsymbol{Z}_i$ for UAV i include its own states and controls $\boldsymbol{z}_i$ and its own time-step size Δt_i, as shown in the following:

$$\boldsymbol{Z}_i=(\boldsymbol{z}_i,\Delta t_i)^T,\qquad \boldsymbol{z}_i=(\boldsymbol{s}_i[0],\ldots,\boldsymbol{s}_i[K],\boldsymbol{u}_i[0],\ldots,\boldsymbol{u}_i[K])^T,\quad i=1,2,\ldots,N \tag{26}$$

Algorithm 1 Coupled SCP-based cooperative trajectory planning

Initialization: $\bar{\boldsymbol{Z}}=\boldsymbol{Z}^0,\boldsymbol{\rho}=\boldsymbol{\rho}^0,q=0$
1 **while** $\boldsymbol{Z}^q$ does not satisfy Eqs. (8), (9), and (19)
2 $q=q+1,\bar{\boldsymbol{Z}}=\boldsymbol{Z}^{q-1},\boldsymbol{\rho}=0.5\boldsymbol{\rho}$
3 $\boldsymbol{Z}^q\leftarrow$ solution to problem 3 without Eqs. (15) and (16)
4 **end while**
5 $\boldsymbol{\rho}=\boldsymbol{\rho}^0$
6 **while** $\boldsymbol{Z}^q$ does not satisfy Eqs. (7–9), (15), (16), and (20)
7 $q=q+1,\bar{\boldsymbol{Z}}=\boldsymbol{Z}^{q-1},\boldsymbol{\rho}=0.5\boldsymbol{\rho}$
8 $\boldsymbol{Z}^q\leftarrow$ solution to problem 3
9 **end while**
10 **return** $\boldsymbol{Z}^q$

C. Decoupled SCP-Based Cooperative Trajectory Planning

Now that the coupled subproblem has been decomposed into a group of decoupled subproblems, a decoupled SCP method can be derived in Algorithm 2. The decoupled SCP adopts the same two-stage optimization as the coupled SCP.

As can be seen in lines 4–8 and 13–17 of Algorithm 2, the decoupled subproblems of UAVs are solved in parallel at each iteration of the decoupled SCP. Meanwhile, the parallel computing should be conducted synchronously, because the computation of the lower bound on step size (lines 6 and 15) relies on nominal trajectories of all the UAVs. Because the decoupled SCP is implemented in parallel and a decoupled subproblem has less computational complexity than the coupled subproblem, the decoupled SCP is expected to possess higher

Algorithm 2 Decoupled SCP-based cooperative trajectory planning

Initialization: $\bar{z}_i = z_i^0$, $\Delta\bar{t}_i = \Delta t_i^0$, $\boldsymbol{\rho} = \boldsymbol{\rho}^0$, $i = 1, 2, \ldots, N$, $q = 0$
$\Delta t^q \leftarrow \max(\Delta t_i^q)$, $i = 1, 2, \ldots, N$, $\boldsymbol{Z}^0 \leftarrow (z_1^0, z_2^0, \ldots, z_N^0, \Delta t^0)$
while $(z_i^q, \Delta t_i^q)$ does not satisfy Eqs. (8), (9), (19), and (23)
  $q \leftarrow q + 1$, $\boldsymbol{\rho} = 0.5\boldsymbol{\rho}$
  parfor $i = 1, 2, \ldots, N$
    $\bar{z}_i \leftarrow z_i^{q-1}$, $\Delta\bar{t}_i \leftarrow \Delta t^{q-1}$
    $\Delta t_{\min}^q \leftarrow (\max \Delta\bar{t}_j^q + \min \Delta\bar{t}_j^q)/2$, $j \in \Omega$
    $(z_i^q, \Delta t_i^q) \leftarrow$ solution to problem 4 without Eqs. (15) and (21)
  end parfor
end while
$\boldsymbol{\rho} = \boldsymbol{\rho}^0$
while $(z_i^q, \Delta t_i^q)$ does not satisfy Eqs. (7–9), (15), (16), (20), and (23)
  $q \leftarrow q + 1$, $\boldsymbol{\rho} = 0.5\boldsymbol{\rho}$
  parfor $i = 1, 2, \ldots, N$
    $\bar{z}_i \leftarrow z_i^{q-1}$, $\Delta\bar{t}_i \leftarrow \Delta t^{q-1}$
    $\Delta t_{\min}^q \leftarrow (\max \Delta\bar{t}_j^q + \min \Delta\bar{t}_j^q)/2$, $j \in \Omega$
    $(z_i^q, \Delta t_i^q) \leftarrow$ solution to problem 4
  end parfor
end while
return $(z_i^q, \Delta t_i^q)$, $i = 1, 2, \ldots, N$

computational efficiency than the coupled SCP for cooperative trajectory planning. Additionally, the rigorous convergence proof of SCP is still an open problem, but SCP has been validated in many applications as stated in Sec. I, and some proofs have been reported separately for several specific problems considering various sources of nonconvexity (e.g., inter-UAV collision constraints [16], concave state constraints [22], and nonlinear dynamics [23]).

V. Numerical Simulations

In this section, the developed coupled SCP and decoupled SCP methods are demonstrated on numerical simulations of cooperative trajectory planning. The simulations are implemented in MATLAB R2013b on a desktop computer equipped with Intel Core i5-2310 2.90 GHz and 4 GB RAM. Convex-programming subproblems are solved using SeDuMi [24] in CVX [25]. The numerical results by SCP are compared with those by GPOPS-II [26], which combines Radau pseudospectral method and nonlinear programming.

A. Simulation Scenarios and Parameter Setting

Two typical scenarios for UAV cooperation are considered (i.e., formation rendezvous and formation reconfiguration). In the rendezvous case, a formation of UAVs is required to rendezvous from their initial positions to a specified formation. In the reconfiguration case, trajectories of the UAVs are designed to change the formation shape from the line formation to a V formation.

The initial and final positions of UAVs for the formation-rendezvous case and formation-reconfiguration case are given in Tables 1 and 2, respectively. And the velocity, heading angle, and flight-path angle of all the UAVs at the initial time and final time are specified as 25 m/s, 0, and 0, respectively. Note that those states can change within the given bounds during the flight. The bounds on states and controls are defined in Eq. (27). The information of the obstacles is given in Table 3. And the minimum safe distance between UAVs is set as 100 m.

$$\begin{aligned} \boldsymbol{s}_{i,\min} &= (-\infty, -\infty, 200\ \text{m}, 20\ \text{m/s}, -\infty, -5\pi/180) \\ \boldsymbol{s}_{i,\min} &= (+\infty, +\infty, 500\ \text{m}, 30\ \text{m/s}, +\infty, +5\pi/180) \\ \boldsymbol{u}_{i,\min} &= (-0.2, -0.2, 0.8)^T, \qquad \boldsymbol{u}_{i,\max} = (0.2, 0.2, 1.2)^T \end{aligned} \tag{27}$$

For both cases, the coupled SCP, decoupled SCP, GPOPS, and decoupled GPOPS are employed to plan the cooperative trajectories. A similar parallel strategy to the decoupled SCP is used in decoupled GPOPS, which employs GPOPS as the trajectory solver. The parameters of GPOPS are set as the default values. The parameters of the coupled SCP and decoupled SCP are set as Eq. (28). For the initial

Table 1 Initial and final positions of UAVs for the formation-rendezvous case

UAV number	Initial position, m	Final position, m
1	0, 0, 350	1900, 2200, 400
2	0, 2500, 350	2100, 2400, 400
3	0, 5000, 350	2300, 2600, 400
4	2500, 0, 350	2500, 2800, 400
5	5000, 0, 350	3100, 2200, 400
6	5000, 2500, 350	2900, 2400, 400
7	5000, 5000, 350	2700, 2600, 400

Table 2 Initial and final positions of UAVs for the formation-reconfiguration case

UAV number	Initial position, m	Final position, m
1	0, 2000, 350	5000, 2000, 400
2	0, 1700, 350	4800, 1800, 400
3	0, 2300, 350	4800, 2200, 400
4	0, 1400, 350	4600, 1600, 400
5	0, 2600, 350	4600, 2400, 400
6	0, 1100, 350	4400, 1400, 400
7	0, 2900, 350	4400, 2600, 400

Table 3 Obstacle information

Obstacle number	Position, m	Radius, m
1	1500, 1400	250
2	1500, 2200	300
3	1500, 3200	350
4	3500, 900	350
5	3500, 1800	300
6	3500, 2700	250

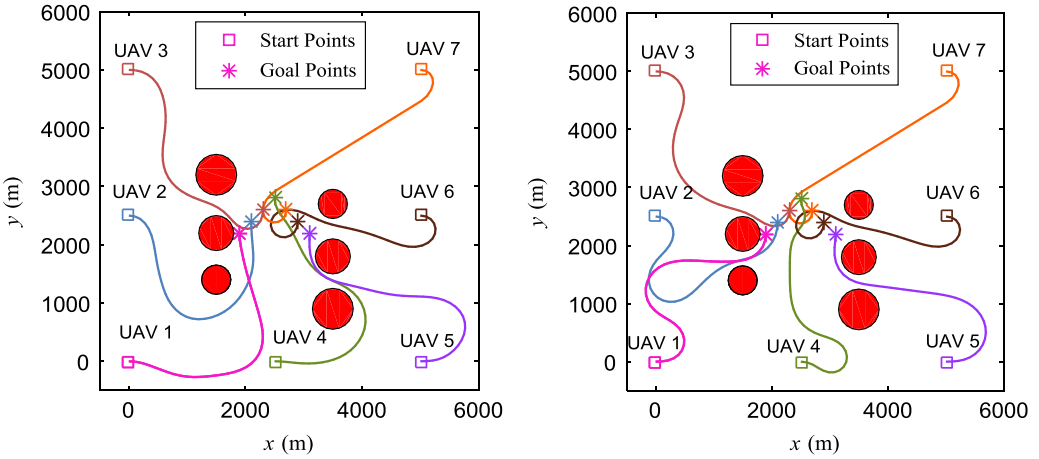
guesses of all the algorithms, the states are generated according to the linear interpolation between initial states and final states, the flight time is determined according to the maximum speed and minimum distance between starts and goals, and the controls are computed using Eq. (7).

$$\begin{aligned} &K = 40, \qquad \tau = 0.1, \qquad \varepsilon_{\Delta t} = 0.1, \\ &\rho^0 = (4000\ \text{m}, 4000\ \text{m}, 100\ \text{m}, 20\ \text{m/s}, \pi, \pi/10)^T, \\ &\varepsilon = (0.1, 0.1, 0.05, 0.01, 0.05\pi/180, 0.01\pi/180)^T \end{aligned} \tag{28}$$

B. Results and Discussions

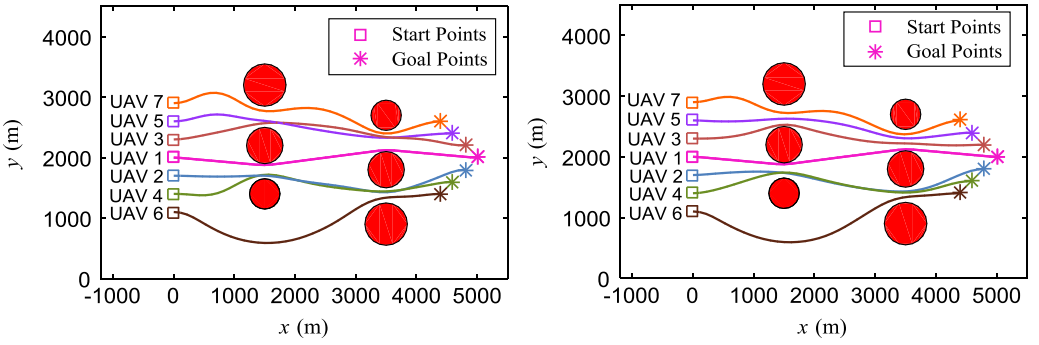
The cooperative trajectory-planning results of the formation-rendezvous and formation-reconfiguration cases are shown in Figs. 1–4. From Figs. 1 and 2, all the generated trajectories by the SCP methods can achieve transition from the initial conditions to the final conditions, and can avoid the obstacles in the environment. In the rendezvous case, the UAVs close to their goals make detours to achieve simultaneous arrival with other UAVs. In the reconfiguration case, the UAVs realize the formation shape changing from the line formation to the V formation. From Figs. 3 and 4, the minimum distance between UAVs is always larger than the safe threshold during the entire flight, that is, the optimized cooperative trajectories satisfy the collision-avoidance constraints. Thus, the coupled SCP and decoupled SCP methods are validated to be effective for multi-UAV cooperative trajectory-planning problems.

To demonstrate the optimality and efficiency of the SCP methods, comparative studies of the coupled SCP, decoupled SCP, coupled GPOPS, and decoupled GPOPS are conducted on scenarios with different numbers of UAVs. The optimized objective values of different formations for rendezvous and reconfiguration are shown in Figs. 5 and 6, respectively. From these figures, the optimized



a) Planned trajectories by coupled SCP b) Planned trajectories by decoupled SCP

Fig. 1 Trajectory-planning results for formation rendezvous.



a) Planned trajectories by coupled SCP b) Planned trajectories by decoupled SCP

Fig. 2 Trajectory-planning results for formation reconfiguration.

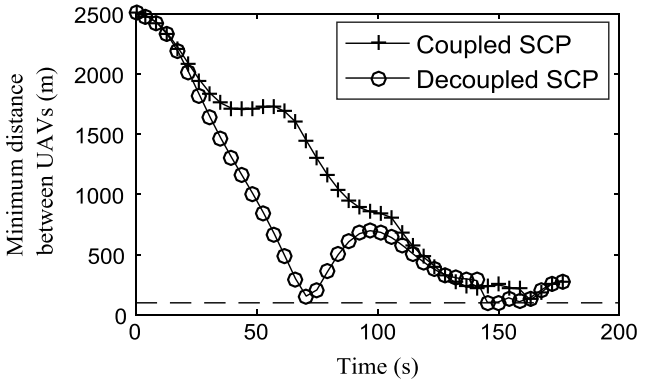


Fig. 3 History of minimum distance between UAVs for formation rendezvous.

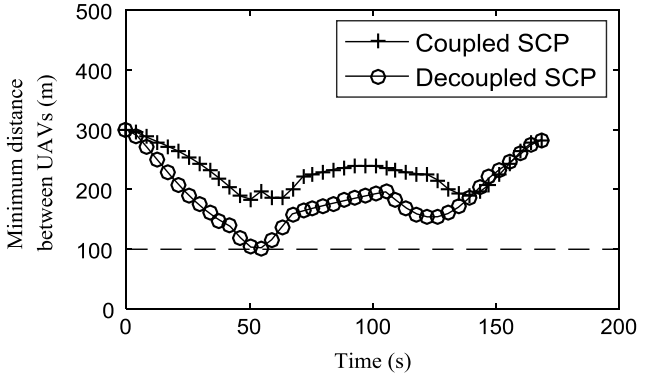


Fig. 4 History of minimum distance between UAVs for formation reconfiguration.

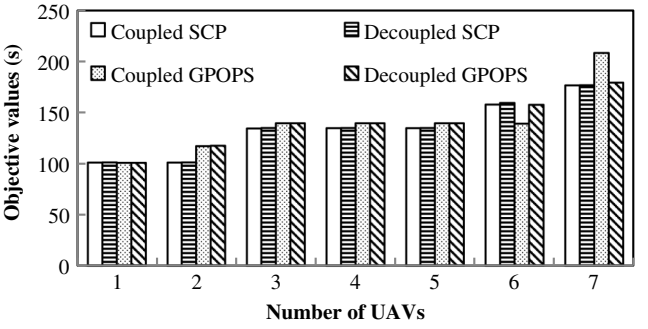


Fig. 5 Optimized objectives for formation rendezvous.

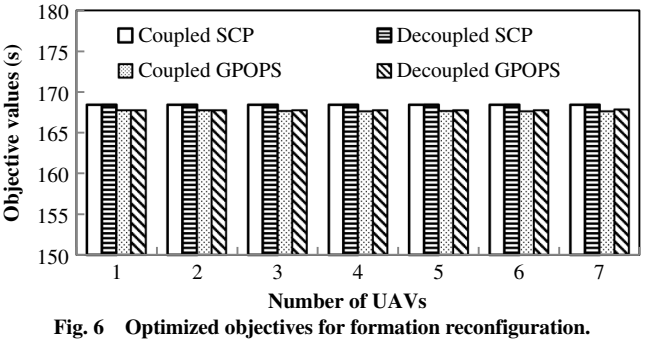
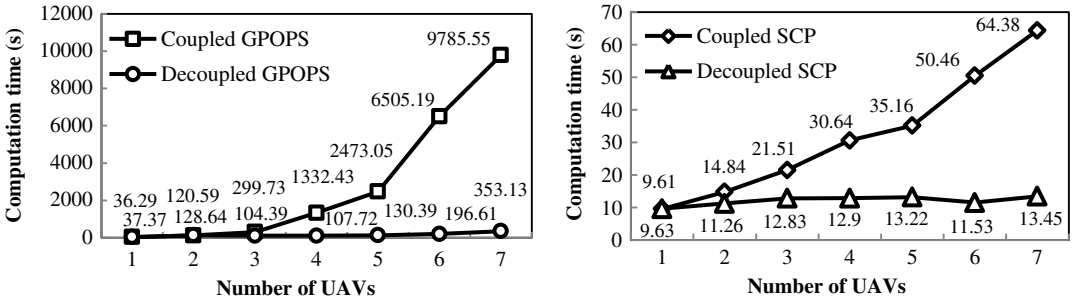


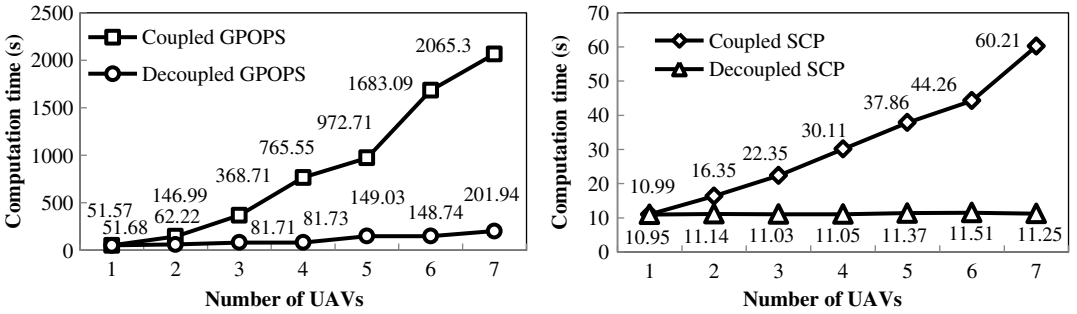
Fig. 6 Optimized objectives for formation reconfiguration.

objective values of the decoupled SCP are almost the same with those of the coupled SCP in all the scenarios. Thus, it is proved that the decoupled SCP can obtain similar optimal solutions to the coupled SCP. On the other hand, there is no significant difference between



a) Computation time of GPOPS b) Computation time of SCP

Fig. 7 Computation time of different methods for formation rendezvous.



a) Computation time of GPOPS b) Computation time of SCP

Fig. 8 Computation time of different methods for formation reconfiguration.

solutions of SCP and GPOPS. From the simulation results, the largest deviation between the SCP and GPOPS solutions (in rendezvous case with seven UAVs) is about 15%, and for most scenarios, the deviation is less than 5%. Thus, the SCP methods are demonstrated to provide comparable optimality to GPOPS for cooperative trajectory planning.

The computation times of the four methods are given in Figs. 7 and 8. The SCP methods show higher efficiency than GPOPS, and this efficiency advantage becomes more evident as the formation scale grows. The computation time of the coupled GPOPS increases exponentially as the number of UAVs grows, whereas the time of the coupled SCP is nearly linear to the number of UAVs. Through problem decoupling and parallel computing, the efficiency of GPOPS can be largely improved, but the decoupled GPOPS still consumes much more computation time than the decoupled SCP. The computation time of the decoupled SCP is insensitive to the number of UAVs. Thus, the decoupled SCP can scale well to the number of UAVs and shows an appealing computation efficiency for cooperative trajectory planning.

VI. Conclusions

In this work, sequential convex programming (SCP) is employed to solve minimum-time cooperative trajectory-planning problems for multiple unmanned aerial vehicles (UAVs). The problem is formulated as a nonconvex optimal-control problem and parameterized to be a nonconvex-programming problem. Then, the SCP method is extended to address the nonconvex problem by solving a series of convex-programming subproblems. And a decoupled SCP method is proposed to decompose the coupled subproblem into multiple independent subproblems, which can be solved in parallel. Numerical simulations on multi-UAV formation-rendezvous and -reconfiguration cases are conducted to demonstrate the effectiveness of the SCP-based cooperative trajectory-planning methods. The simulation results show that the SCP methods outperform GPOPS in terms of efficiency, whereas they show comparable optimality with GPOPS. And the decoupled SCP exhibits an excellent scalability to the number of UAVs, and provides a potential application to trajectory planning of large-scale UAV formation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant numbers 51675047, 51105040, and 11372036), and the Aeronautical Science Foundation of China (grant number 2015ZA72004).

References

[1] Shima, T., and Rasmussen, S., *UAV Cooperative Decision and Control: Challenges and Practical Approaches*, Soc. for Industrial and Applied Mathematics, Philadelphia, PA, 2009, pp. 6–10.

[2] Ollero, A., and Maza, I., *Multiple Heterogeneous Unmanned Aerial Vehicles*, Springer–Verlag, Berlin, 2007, pp. 207–228.

[3] Kuwata, Y., and How, J., "Cooperative Distributed Robust Trajectory Optimization Using Receding Horizon MILP," *IEEE Transactions on System Technology*, Vol. 19, No. 2, 2011, pp. 423–431. doi:10.1109/TCST.2010.2045501

[4] Neto, A., Macharet, D., and Campos, M., "On the Generation of Trajectories for Multiple UAVs in Environments with Obstacles," *Journal of Intelligent and Robotic Systems*, Vol. 57, No. 1, 2010, pp. 123–141. doi:10.1007/s10846-009-9365-3

[5] McCracken, B. D., Burzota, K. L., Ramos, M. M., and Jorris, T. R., "Cooperative Unmanned Air Vehicle Mission Planning and Collision Avoidance," *Infotech@Aerospace Conferences*, AIAA Paper 2011-1584, March 2011.

[6] Duan, H., Luo, Q., Shi, Y., and Ma, G., "Hybrid Particle Swarm Optimization and Genetic Algorithm for Multi-UAV Formation Reconfiguration," *IEEE Computational Intelligence Magazine*, Vol. 8, No. 3, 2013, pp. 16–27. doi:10.1109/MCI.2013.2264577

[7] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge Univ. Press, Cambridge, England, U.K., 2004, pp. 7–8.

[8] Açıkmeşe, B., and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. doi:10.2514/1.27553

[9] Tillerson, M., Inalhan, G., and How, J., "Co-Ordination and Control of Distributed Spacecraft Systems Using Convex Optimization Techniques," *International Journal of Robust and Nonlinear Control*, Vol. 12, Nos. 2–3, 2002, pp. 207–242. doi:10.1002/(ISSN)1099-1239

[10] Açıkmeşe, B., Scharf, D. P., Murray, E. A., and Hadaegh, F. Y., "A Convex Guidance Algorithm for Formation Reconfiguration," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2006-6070, Aug. 2006.

[11] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Spacecraft Swarm Guidance Using a Sequence of Decentralized Convex Optimizations," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2012-4583, Aug. 2012.

[12] Lu, P., and Liu, X., "Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389. doi:10.2514/1.58436

[13] Liu, X., Shen, Z., and Lu, P., "Entry Trajectory Optimization by Second-Order Cone Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2016, pp. 227–241. doi:10.2514/1.G001210

[14] Szmuk, M., Açıkmeşe, B., and Berning, A., "Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2016-0378, Jan. 2016.

[15] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Model Predictive Control of Swarm of Spacecraft Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740. doi:10.2514/1.G000218

[16] Morgan, D., Subramanian, G. P., Chung, S.-J., and Hadaegh, F. Y., "Swarm Assignment and Trajectory Optimization Using Variable-Swarm, Distributed Auction Assignment and Sequential Convex Programming," *International Journal of Robotics Research*, Vol. 35, No. 10, 2016, pp. 1261–1285. doi:10.1177/0278364916632065

[17] Augugliaro, F., Schoellig, A. P., and D'Andrea, R., "Generation of Collision-Free Trajectories for a Quadrocopter Fleet: A Sequential Convex Programming Approach," *25th IEEE/RSJ International Conference on Robotics and Intelligent Systems*, IEEE Publ., Piscataway, NJ, 2012, pp. 1917–1922.

[18] Chen, Y., Cutler, M., and How, J., "Decoupled Multiagent Path Planning via Incremental Sequential Convex Programming," *IEEE International Conference on Robotics and Automation*, IEEE Publ., Piscataway, NJ, 2015, pp. 5954–5961.

[19] Yakimenko, O. A., "Direct Method for Rapid Prototyping of Near-Optimal Aircraft Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, 2000, pp. 865–875. doi:10.2514/2.4616

[20] Dai, R., and Cochran, J. E., "Three-Dimensional Trajectory Optimization in Constrained Airspace," *Journal of Aircraft*, Vol. 46, No. 2, 2009, pp. 627–634. doi:10.2514/1.39327

[21] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Soc. for Industrial and Applied Mathematics, Philadelphia, PA, 2010, pp. 132–134.

[22] Liu, X., and Lu, P., "Solving Nonconvex Optimization Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765. doi:10.2514/1.62110

[23] Mao, Y., Szmuk, M., and Açıkmeşe, B., "Successive Convexification of Non-Convex Optimal Control Problems and Its Convergence Properties," *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE Publ., Piscataway, NJ, 2016, pp. 3636–3641.

[24] Sturm, J. F., "Using SeDuMi 1.02, A Matlab Toolbox for Optimization over Symmetric Cones," *Optimization Methods and Software*, Vol. 11, Nos. 1–4, 1999, pp. 625–653. doi:10.1080/10556789908805766

[25] Grant, M. C., and Boyd, S. P., "CVX: Matlab Software for Disciplined Convex Programming, Ver. 2.1," June 2015, http://cvxr.com/cvx [retrieved 14 Oct. 2015].

[26] Patterson, M. A., and Rao, A. V., "GPOPS-II: A Matlab Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, 2014, pp. 1–37. doi:10.1145/2684421