A Convex Guidance Algorithm for Formation Reconfiguration

A. Behçet Açıkmeşe,* Daniel P. Scharf,† Emmanuell A. Murray,‡ and Fred Y. Hadaegh§

Jet Propulsion Laboratory, California Institute of Technology,
Mail-Stop:198-326, Pasadena, CA 91109-8099,USA.

In this paper, a reconfiguration guidance algorithm for formation flying spacecraft is presented. The formation reconfiguration guidance problem is first formulated as a continuous-time minimum-fuel or minimum-energy optimal control problem with collision avoidance and control constraints. The optimal control problem is then discretized to obtain a finite dimensional parameter optimization problem. In this formulation, the collision avoidance constraints are imposed via separating planes between each pair of spacecraft. A heuristic is introduced to choose these separating planes that leads to the convexification of the collision avoidance constraints. Additionally, convex constraints are imposed to guarantee that no collisions occur between discrete time samples. The resulting finite dimensional optimization problem is a second order cone program, for which standard algorithms can compute the global optimum with deterministic convergence and a prescribed level of accuracy. Consequently, the formation reconfiguration algorithm can be implemented onboard a spacecraft for real-time operations.

I. Introduction

Formation reconfiguration guidance is the planning of optimal translational trajectories to transfer spacecraft from their current states to a set of desired final states in a given time without violating the collision avoidance and control constraints [1]. Several emerging formation flying missions [2–5] will require reconfigurations both to establish their science configurations after deployment and to re-target or change baselines during observations [6]. Similarly, Earth-orbiting formations, such as interferometric synthetic aperture radar or sparse antenna synthesis, will use reconfigurations to tailor radar baselines and gain patterns to specific targets. Reconfigurations can also be used as part of a formation fault response to establish a sub-formation of healthy spacecraft.

There are variety of formation reconfiguration guidance algorithms proposed in the literature (see [1] for a survey). The reconfiguration algorithm developed in [7] is based on a mixed-integer linear programming problem (MILP) formulation. Since MILPs are inherently non-deterministic polynomial time (NP)-complete, this algorithm scales exponentially with the number of spacecraft. In [8] reconfiguration trajectories are parameterized as polynomials and a heuristically-modified gradient descent algorithm is used to satisfy the collision avoidance constraints. In [9, 10] the reconfiguration trajectories are also polynomials, but are constrained to pass through numerically generated waypoints to avoid collisions. In [11] a convex programming approach with randomized schemes is used to solve the reconfiguration problem with collision avoidance constraints. Potential-based path-planning techniques are common in the robotic path-planning literature (see [1,8] for references). These algorithms are not directly applicable since they cannot guarantee collision avoidance with constrained control magnitudes. Additional results on formation reconfiguration can also be found in [12–15].

In this paper, each spacecraft is assumed to be a point mass in deep space or low-earth-orbit (LEO), and only translational path planning is considered. The first step in the formulation is to linearize and time-discretize the relative spacecraft dynamics, thereby converting the infinite-dimensional, continuous-time optimization problem into a finite-dimensional parameter optimization problem. The collision avoidance and control magnitude constraints are enforced at the discrete time samples as second-order cone (SOC) [16] or linear inequality constraints. The collision avoidance constraint is imposed by using separating planes between each pair of spacecraft, where a heuristic is used

*Senior Engineer, AIAA Member, Guidance and Control Analysis Group, Jet Propulsion Laboratory
†Senior Engineer, AIAA Member, Guidance and Control Analysis Group, Jet Propulsion Laboratory
‡Associate Engineer, Guidance and Control Analysis Group, Jet Propulsion Laboratory
§Senior Research Scientist, AIAA Fellow, Guidance and Control Analysis Group, Jet Propulsion Laboratory

to generate these separating planes. To ensure no collisions occur between discrete time samples, we impose additional constraints to guarantee that a relative trajectory does not pass across the separating plane between two discrete time samples. This additional constraint also allows larger time steps to be used, thereby shortening algorithm computation time. As a result, the non-convex collision avoidance constraints are replaced by linear inequality or SOC constraints, thereby converting the reconfiguration guidance problem into a second order cone program (SOCP). Since there are interior-point algorithms that can compute the global optimum of SOCPs with a deterministic stopping criteria and a prescribed level of accuracy, the resulting formation reconfiguration algorithm can be implemented for real-time, onboard operations.

In summary, the principal challenge in translational reconfiguration guidance is that the collision avoidance constraint describes a non-convex feasible region in the state-space. In fact, the trajectory optimization problem with collision avoidance constraints is NP-complete [17, 18]. Regarding the previous work in formation reconfiguration, the algorithm developed in this paper makes two specific contributions. First, the collision avoidance constraint is convexified via a heuristic, resulting in a scalable, efficient algorithm. Second, we extend the collision avoidance constraint to apply not only at the discrete times, but between the time samples as well.

The paper is organized as follows: Section II introduces the formation reconfiguration problem formulation with control and state constraint. Section III introduces the heuristic used to convexify the problem and a solution algorithm. Section IV presents illustrative simulations of the algorithm. Section V summarizes our conclusions and describe potential future research problems. The notation used is as follows: $\mathbb{R}$ is the set of real numbers, $\mathbb{R}^n$ is the space of n dimensional vectors with real components, I denotes the identity matrix of appropriate dimensions, $\|x\|$ denotes the standard 2-norm of a vector x, and, for any $f : \mathbb{R}_+ \to \mathbb{R}^n$, $f(\cdot)$ represents the time profile defined by f on a time interval $[0, T]$.

II. Formation Reconfiguration Trajectory Planning

In this section, we first describe the reconfiguration trajectory planning problem as a continuous finite horizon optimal control problem. Then, the optimal control problem is discretized to obtain a finite dimensional parameter optimization problem. This approach describes a direct method [19] to solve an optimal control. We also utilize separating planes to impose the collision avoidance constraints in the discrete optimal control problem.

A. Formulation of an Optimal Control Problem

In this paper, the spacecraft are modeled as point masses in deep space or a circular LEO with a single thrust vector, and we assume the following linear dynamics for each spacecraft,

$$\dot{x}_j = A_c x_j + B_c u_i, \qquad j = 1, \ldots, N \tag{1}$$

where N is the number of spacecraft, $x_j \in \mathbb{R}^6$ is the state vector and $u_j \in \mathbb{R}^3$ is the control acceleration vector of the ith spacecraft,

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix}, \qquad B_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

and ω is a constant determined by the orbit ($\omega = 0$ for deep space). The state vector is composed of position and velocity relative to the orbit in LEO and to an inertially fixed point in deep space, i.e.,

$$x_j = \begin{bmatrix} r_j \\ v_j \end{bmatrix},$$

where $r_j \in \mathbb{R}^3$ is the position vector and $v_j \in \mathbb{R}^3$ is the velocity vector. The objective of a reconfiguration maneuver is to bring the formation to a desired configuration at time $t = T > 0$ from an existing configuration at time $t = 0$, which

implies the following state constraints

$$\begin{aligned} x_1(0) &= x_{1,0} \\ x_{1j}(0) &= x_{1j,I}, \qquad j = 2,\ldots,N \\ x_{1j}(T) &= x_{1j,F}, \qquad j = 2,\ldots,N \end{aligned} \tag{3}$$

where, for $j = 2,\ldots,N$, $x_{1j,I}$ are the initial states of all spacecraft relative to the first one, $x_{1j,F}$ are the desired states relative to the first spacecraft, $x_{1,0}$ is the initial state of the first spacecraft, and

$$x_{ij} = x_j - x_i, \qquad j > i, \;\; i = 1,\ldots,N-1.$$

Note that the initial and final state are assumed to satisfy all other state constraints that will be listed from this point on. There can be an additional constraint of one of the following forms for the first spacecraft's end position

$$\begin{aligned} &x_1(T) = x_{1,F} \qquad \text{or} \\ &x_1(T) \in \mathrm{Co}\{\mathrm{a}_1,\ldots,\mathrm{a}_{\mathrm{m}_1}\} \end{aligned} \tag{4}$$

where $m_1 > 1$ is a positive integer, $\mathrm{Co}\{\mathrm{a}_1,\ldots,\mathrm{a}_{\mathrm{m}_1}\}$ indicates the convex hull of the vectors $a_1,\ldots,a_{m_1}$. The equality in (4) constraints the final state of the first spacecraft to a prescribed value and the inequality in (4) bounds it to a region.

Remark 1 A set of the following form

$$X = \{x \,:\, x \in \mathrm{Co}\{\mathrm{a}_1,\ldots,\mathrm{a}_{\mathrm{m}}\}\},$$

where $a_1,\ldots,a_m$ are given vectors, can equivalently be expressed by a finite number of linear inequalities. We use the convex hull notation in some of the constraint descriptions for its notational compactness. ◇

Relative state constraints are imposed on $t \in (0,T)$ in the following general form

$$Fx_{ij}(t) \in \mathrm{Co}\{\mathrm{b}_1,\ldots,\mathrm{b}_{\mathrm{m}}\}, \mathrm{j} > \mathrm{i}, \;\; \mathrm{i} = 1,\ldots,\mathrm{N}-1, \quad \mathrm{t} \in (0,\mathrm{T}). \tag{5}$$

For example, by choosing $F = C_v$ where

$$C_v = \begin{bmatrix} 0 & I \end{bmatrix}, \tag{6}$$

we can bound the relative velocity between each pair of spacecraft with (5).

The only control constraint considered considered in this paper is a bound on available control acceleration

$$\|u_j(t)\| \le U_j, \qquad j = 1,\ldots,N, \quad t \in [0,T]. \tag{7}$$

All of the constraints mentioned up to this point are SOC constraints and they define a convex set of feasible solutions. The last constraint is the collision avoidance constraint that makes the problem non-convex and NP-complete [17],

$$\|C_p x_{ij}(t)\| \ge R_{ij} > 0, \qquad j > i, \;\; i = 1,\ldots,N-1, \quad t \in [0,T], \tag{8}$$

where R_{ij} is the minimum allowable distance between ith and jth spacecraft,

$$C_p = \begin{bmatrix} I & 0 \end{bmatrix}. \tag{9}$$

Given all the constraints above, we can describe the formation reconfiguration trajectory planning problem as follows:

Problem 1

$$\min_{u_j(\cdot),\, j=1,\ldots,N} \|\phi^c\|_\lambda \qquad \text{subject to} \qquad \left\{ \ (1), (3), (4), (5), (7), (8) \ \right\}$$

where $\phi^c \in \mathbb{R}^N$ *is defined by*

$$\phi_j^c = \begin{cases} \int_0^T \|u_j(t)\|\, dt, & l = 1; \\ \int_0^T \|u_j(t)\|^2 dt, & l = 2. \end{cases} \qquad j = 1,\ldots,N, \tag{10}$$

and $\|\phi\|_\lambda$ *defines the norm of the vector* $\phi \in \mathbb{R}^N$ *as follows*

$$\|\phi\|_\lambda = \begin{cases} \sum_{j=1}^N |\phi_j|, & \lambda = 1; \\ \sum_{j=1}^N \phi_j^2, & \lambda = 2; \\ \max_{j=1,\ldots,N} |\phi_j|, & \lambda = \infty. \end{cases} \tag{11}$$

Note that the cost in the Problem 1 is also convex, and we have the following table that gives a physical sense for the cost function:

$l \setminus \lambda$	1	2	∞
1	Total fuel	-	Maximum fuel in a spacecraft
2	-	Total energy	Maximum energy in a spacecraft

B. Discretization of the Optimal Control Problem

The dynamics of each spacecraft (1) is discretized with a zero-order-hold approach, i.e.,

$$u_j(t) = u_j[k], \qquad t \in [t_k,\, t_{k+1}), \quad j = 1,\ldots,N, \ \ k = 1,\ldots,n,$$

where n is a positive integer such that $T = (n+1)\Delta t$, $k = 1$ refers to $t = 0$, and

$$\Delta t = t_{k+1} - t_k, \qquad k = 1,\ldots,n.$$

This discretization of the control input leads to the following set of equalities

$$x_j[k+1] = Ax_j[k] + Bu_j[k], \qquad j = 1,\ldots,N, \ \ k = 1,\ldots,n, \tag{12}$$

where

$$A = e^{A_c \Delta t}, \qquad B = \int_0^{\Delta t} e^{A_c(t-\tau)} B_c\, d\tau,$$

$$x_j[k] = x_j(t_k), \qquad u_j[k] = u_j(t_k), \quad t_k = k\Delta t.$$

The state and control constraints for the discrete problem are expressed at as follows,

$$\begin{aligned} x_1[1] &= x_{1,0} \\ x_{1j}[1] &= x_{1j,I}, \qquad j = 2,\ldots,N \\ x_{1j}[n+1] &= x_{1j,F}, \qquad j = 2,\ldots,N \\ x_1[n+1] &= x_{1,F} \quad \text{or} \quad x_1[n+1] \in \mathrm{Co}\{\mathrm{a}_1,\ldots,\mathrm{a}_{\mathrm{m_a}}\} \\ Fx_{ij}[k] &\in \mathrm{Co}\{\mathrm{b}_1,\ldots,\mathrm{b}_{\mathrm{m_b}}\}, \quad \mathrm{j} > \mathrm{i}, \ \mathrm{i} = 1,\ldots,\mathrm{N}-1, \quad \mathrm{k} = 1,\ldots,\mathrm{n}, \\ \|u_j[k]\| &\le U_j, \quad j = 1,\ldots,N, \qquad k = 1,\ldots,n, \end{aligned} \tag{13}$$

$$\|C_p x_{ij}[k]\| \ge R_{ij} > 0, \qquad j > i, \ \ i = 1,\ldots,N-1, \quad k = 1,\ldots,n, \tag{14}$$

and ϕ^c describing the cost is given by

$$\phi_j^d = \begin{cases} \sum_1^n \|u_j[k]\|, & l=1; \\ \sum_1^n \|u_j[k]\|^2, & l=2. \end{cases} \qquad j=1,\ldots,N, \tag{15}$$

where

$$\|c_{ij}[k]\| = 1.$$

Note that the discrete version of the collision avoidance constraint (14) is only satisfied at the times t_k, $k = 1,\ldots,n+1$ as typical for a direct numerical method [7]. However, this constraint alone does not guarantee the collision avoidance on (t_k, t_{k+1}) for any $k = 1,\ldots,n$. This observation together with a desire to convexify the collision avoidance constraint motivate the utilization of the separating planes to impose the collision avoidance constraint, i.e., two spacecraft is separated by a prescribed plane

$$c_{ij}[k]^T C_p x_{ij}[k] \geq R_{ij}, \quad j > i, \ i = 1,\ldots,N-1, \ k = 1,\ldots,n. \tag{16}$$

The next question is how to generate the separating planes that is how to generate the unit vectors $c_{ij}[k]$, $j > i$, $i = 1,\ldots,N-1$, $k = 1,\ldots,n$. A heuristic to generate the separating planes will be discussed in detail in the next section, and, in this section, it is only assumed that two consecutive half-spaces for a pair of spacecraft has a nonempty intersection, that is

$$c_{ij}[k]^T c_{ij}[k+1] \geq \alpha > 0, \quad j > i, \ i = 1,\ldots,N-1, \ k = 1,\ldots,n-1, \tag{17}$$

where α is a positive constant.

The following lemma establishes two additional constraints together with (16) that guarantees the collision avoidance between discrete time samples.

Lemma 1 *Consider the position component $r = C_p x$, where C_p is as in (9), of the state x with dynamics (12) and a vector $c \in \mathbb{R}^3$ with $\|c\| = 1$. Then the inequality below is satisfied*

$$c^T r(\tau) \geq R, \qquad \forall \tau \in [t,\ t+\Delta t], \tag{18}$$

if the following conditions hold: The input u in (1) is constant over $[t,\ t+\Delta t)$ that is

$$u(\tau) = \eta, \qquad \tau \in [t,\ t+\Delta t), \tag{19}$$

and

$$c^T\left[-0.5\Delta t^2\eta + r(t)\right] \geq R, \quad c^T r(t) \geq R, \quad c^T r(t+\Delta t) \geq R \qquad \textit{when} \qquad \omega = 0 \tag{20}$$

or

$$\begin{aligned} &-0.5\alpha(c,x(t),\eta)\Delta t^2 + \gamma(c,x(t),\eta) - R \geq \left\| \begin{bmatrix} \beta_1(c,x(t),\eta) \\ \beta_2(c,x(t),\eta) \end{bmatrix} \right\|, \\ &c^T r(t) - R \geq 2\left\| \begin{bmatrix} \beta_1(c,x(t),\eta) \\ \beta_2(c,x(t),\eta) \end{bmatrix} \right\|, \quad c^T r(t+\Delta t) - R \geq 2\left\| \begin{bmatrix} \beta_1(c,x(t),\eta) \\ \beta_2(c,x(t),\eta) \end{bmatrix} \right\| \end{aligned} \qquad \textit{when} \qquad \omega > 0, \tag{21}$$

where α, γ, β_1, *and* β_2 *are* linear *functions of* c, $x(t)$, *and* η *defined as*

$$\begin{aligned}
\alpha(c,x(t),\eta) &= c^T \begin{bmatrix} 0 \\ -3\eta_2/2 \\ 0 \end{bmatrix} \\
\gamma(c,x(t),\eta) &= c^T \begin{bmatrix} \left(4\omega^2 x_1(t) + 2\omega x_2(t) + \eta_1\right)/\omega^2 \\ 2\left(4\omega^2 x_1(t) + 2\omega x_2(t) + \eta_1\right)/\omega^3 - 2x_4(t)/\omega \\ \eta_3/\omega^2 \end{bmatrix} \\
\beta_1(c,x(t),\eta) &= c^T \begin{bmatrix} x_1(t) - \left(4\omega^2 x_1(t) + 2\omega x_2(t) + \eta_1\right)/\omega^2 \\ -2\left(4\omega^2 x_1(t) + 2\omega x_2(t) + \eta_1\right)/\omega^3 + 2x_4(t)/\omega \\ x_3(t) - \eta_3/\omega^2 \end{bmatrix} \\
\beta_1(c,x(t),\eta) &= c^T \begin{bmatrix} \left(4\omega^2 x_1(t) + 2\omega x_2(t) + \eta_1\right)/\omega^3 - x_4(t)/\omega \\ -2x_1(t) + 2\left(4\omega^2 x_1(t) + 2\omega x_2(t) + \eta_1\right)/\omega^2 \\ x_6(t)/\omega \end{bmatrix},
\end{aligned} \tag{22}$$

$x(t) = [x_1(t),\ x_2(t),\ x_3(t),\ x_4(t),\ x_5(t),\ x_6(t)]^T,\ \eta = [\eta_1,\ \eta_2,\ \eta_3]^T$. ◇

Remark 2 Since α, γ, β_1, and β_2 are **linear** functions of $x(t)$ and η, constraints in (20) and (21) define convex feasible regions (second order cones) for $x(t)$ and η. ◇

Proof: • $\underline{\omega = 0:}$
In this case, we have

$$r(\tau) = C_p x(\tau) = r(t) + \dot{r}(t)(\tau - t) + \eta(\tau - t)^2/2 \qquad t \geq t.$$

Let,

$$g(\tau) = c^T r(\tau) - R = a_0 + a_1(\tau - t) + a_2(\tau - t)^2/2, \tag{23}$$

where $a_0 = c^T r(t) - R$, $a_1 = c^T \dot{r}(t)$, and $a_2 = c^T \eta$. The second and third inequality in (20) imply that

$$g(t) \geq 0 \qquad \text{and} \qquad g(t + \Delta t) \geq 0. \tag{24}$$

Consequently, if $a_2 \leq 0$, $g(\tau) \geq 0$ for all $\tau \in [t,\ t + \Delta t]$. When, $a_2 > 0$, $g(\tau)$ attains a minimum at $\tau^* = t - a_1/a_2$ and

$$g(\tau^*) = a_0 - a_2(\tau^* - t)^2/2.$$

If $\tau^* \notin [t,\ t + \Delta t]$ then $g(\tau) \geq 0$ for all $[t,\ t + \Delta t]$. Otherwise, since $a_2 > 0$,

$$g(\tau^*) \geq a_0 - a_2 \Delta t^2/2.$$

Now using the first inequality in (20) the following follows

$$g(\tau) \geq 0 \qquad \tau \in [t,\ t + \Delta t].$$

- $\underline{\omega > 0:}$

  In this case, we can solve the linear differential equations and show that

$$g(\tau) = c^T r(\tau) - R = \gamma - R + \beta(\tau - t) + \alpha(\tau - t)^2/2 + \beta_1 \cos(\omega(\tau - t)) + \beta_2 \sin(\tau - t) cos(\omega(\tau - t)),$$

where the argument list $(c, x(t), \eta)$ is dropped for simplicity, and β is a linear function of c, $x(t)$, and η (its explicit form is given since it is not needed for the proof). Let

$$g(\tau) = g_1(\tau) + g_2(\tau),$$

where

$$g_1(\tau) = \gamma - R + \beta(\tau - t) + \alpha(\tau - t)^2/2$$

and

$$g_2(\tau) = \beta_1 \cos(\omega(\tau - t)) + \beta_2 \sin(\tau - t) cos(\omega(\tau - t)).$$

Note that

$$g_2(\tau) \leq \left\| \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \right\| \qquad \forall \tau. \tag{25}$$

Consequently, the second and third inequalities in (21) imply that

$$g_1(t) \geq \left\| \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \right\| \qquad \text{and} \qquad g_1(t+\Delta t) \geq \left\| \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \right\|.$$

This implies that

$$g_1(t) \geq \left\| \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \right\| \qquad \forall \tau \in [t,\, t+\Delta t]$$

when $\alpha \leq 0$. Now, defining

$$h(\tau) = g_1(\tau) - \left\| \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \right\|,$$

we note that the function h has the same for as g defined in (23) and it satisfies $h(t) \geq$ and $h(t+\Delta t) \geq 0$ as in (24). Therefore, progressing as in the first part of this proof (when $\omega = 0$) and using the first inequality in (21) as the first inequality in (20), we can conclude that

$$h(\tau) \geq 0 \quad \Rightarrow \quad g_1(\tau) \geq \left\| \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \right\| \qquad \forall \tau \in [t,\, t+\Delta t].$$

Now, by using(25), we obtain that

$$g(t) \geq 0 \qquad \forall \tau \in [t,\, t+\Delta t].$$

This concludes the proof of the lemma. ■

Remark 3 The inequalities (20) give conditions under which there exists no collision at any time during a reconfiguration maneuver, and the inequalities (21) give the corresponding condition at LEO. Both set of inequalities define convex sets, however, the inequalities for deep space define only linear cones [16] where as the ones for LEO contain SOCs. In a SOCP existence of more SOCs imply more computation time [20]. Therefore, when ω is small, we use the inequalities in (20) instead of (21). ◇

The following parameter optimization problem is an approximation discrete approximation of Problem 1 where collision avoidance constraints are imposed via the separating planes by utilizing Lemma 1.

Problem 2

$$\min_{u_j[k],\ j=1,\ldots,N,\ k=1,\ldots,N} \|\phi^d\|_\lambda \qquad \text{subject to} \quad \left\{ \ (12), (13) \ \right\} \quad \text{and}$$

Collision avoidance constraints for $\omega = 0$:

$$\begin{aligned} c_{ij}[k]^T \left(-0.5\Delta t^2 \left(u_j[k] - u_i[k]\right) + r_{ij}[k]\right) &\geq R_{ij}, \\ c_{ij}[k]^T r_{ij}[k] \geq R_{ij}, \quad c_{ij}[k]^T r_{ij}[k+1] &\geq R_{ij}, \end{aligned} \qquad k = 1,\ldots,n,\ j > i,\ i = 1,\ldots,N-1; \tag{26}$$

Collision avoidance constraints for $\omega > 0$:

$$\begin{aligned} -0.5\alpha(z_{ij}[k])\Delta t^2 + \gamma(z_{ij}[k]) - R_{ij} &\geq \left\| \begin{bmatrix} \beta_1(z_{ij}[k]) \\ \beta_2(z_{ij}[k]) \end{bmatrix} \right\|, \\ c_{ij}[k]^T r_{ij}[k] - R_{ij} &\geq 2 \left\| \begin{bmatrix} \beta_1(z_{ij}[k]) \\ \beta_2(z_{ij}[k]) \end{bmatrix} \right\|, \qquad k = 1,\ldots,n,\ j > i,\ i = 1,\ldots,N-1, \\ c_{ij}[k]^T r_{ij}[k+1] - R_{ij} &\geq 2 \left\| \begin{bmatrix} \beta_1(z_{ij}[k]) \\ \beta_2(z_{ij}[k]) \end{bmatrix} \right\|, \end{aligned} \tag{27}$$

where

$$r_{ij}[k] = C_p x_{ij}[k], \qquad z_{ij}[k] = \left[\ c_{ij}[k],\ x_{ij}[k],\ u_j[k] - u_i[k]\ \right]^T,$$

the linear functions α, γ, β_1, and β_2 are defined in (22), and ϕ_d and $\|\cdot\|_\lambda$ are defined by (15) and (11).

III. Convex Reconfiguration Algorithm

In this section, a convex solution algorithm for collision free formation reconfiguration based on the solution of Problem 2 is described. First, we present the heuristic used to generate the separating planes defined by $c_{ij}[k]$ for $k = 1,\ldots,n,\ j > i,\ i = 1,\ldots,N-1$. The main objective in generating the separating planes is to have a nonempty feasible set for the inequalities described by them at any time step. This is a necessary condition for the second and third inequalities in (26) and (27) to have a nonempty feasible set of solutions. Geometrically this makes the planes separating each pair of spacecraft consistent. More precisely, once a set of separating planes are chosen, the following set of inequalities define the separation between each pair of spacecraft (time index is dropped for simplicity)

$$c_{ij}^T r_{ij} \geq R_{ij}, \qquad j > i,\ i = 1,\ldots,N-1, \tag{28}$$

which can be shown in a more compact form as follows

$$Ar \succeq \rho, \tag{29}$$

where $\succeq$ indicates element-wise inequalities,

$$A^T = \left[\begin{array}{ccccc|ccc|ccc|c} c_{12} & 0 & 0 & \ldots & 0 & -c_{23} & -c_{24} & \ldots & 0 & \ldots & \ldots & 0 \\ 0 & c_{13} & 0 & \ldots & 0 & c_{23} & 0 & \ldots & -c_{34} & \ldots & \ldots & 0 \\ 0 & 0 & \ddots & . & . & 0 & c_{24} & \ldots & c_{34} & \ldots & \ldots & 0 \\ \vdots & \vdots & . & \ddots & 0 & \vdots & \vdots & . & 0 & \ldots & \ldots & -c_{N-1,N} \\ 0 & 0 & \ldots & 0 & c_{1N} & 0 & \ldots & \ldots & \vdots & \ldots & \ldots & c_{N-1,N} \end{array}\right] \tag{30}$$

$$r = \begin{bmatrix} r_{12} \\ r_{13} \\ \vdots \\ r_{1N} \end{bmatrix}, \qquad \rho = \left[\begin{array}{c} R_{12} \\ R_{13} \\ \vdots \\ \hline R_{23} \\ \vdots \\ \hline \vdots \\ \hline R_{N-1,N} \end{array}\right]. \tag{31}$$

The following lemma establishes a sufficient condition for the existence of a nonempty feasible set for (29), that is, it establishes a sufficient condition for consistent generation of the separating planes.

Lemma 2 *Given a set of unit vectors* $c_{12},\ldots,c_{1N}$ *in* $\mathbb{R}^3$. *Then, inequalities defined by (29) have a nonempty set of feasible solutions if* $c_{ij},\ j>i, i=2,\ldots,N-1$, *are chosen to satisfy*

$$c_{1i}^T c_{ij} < 0. \tag{32}$$

◇

Proof: According to a form of Farkas Lemma [21] one and only one of the following has a solution:

$$Ax \succeq \rho, \tag{33}$$

$$A^T y = 0, \quad \rho^T y = 1, \quad y \succeq 0. \tag{34}$$

Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be the set of feasible solutions for (33) and (34). Farkas Lemma indicates that $\mathcal{F}_1$ is nonempty if and only if $\mathcal{F}_2$ is empty. Therefore, we will prove the lemma via contradiction by showing that $\mathcal{F}_2$ is empty. First noting that

$$\rho \succ 0,$$

the system of equalities and inequalities described by (34) is equivalent to

$$A^T y = 0, \quad y \succeq 0, \quad y \neq 0. \tag{35}$$

Suppose $\mathcal{F}_2$ is nonempty and $y = [y_{12},\ y_{13},\ \ldots, y_{N-1,N}] \in \mathcal{F}_2$ is a solution. Since

$$y_{12}c_{12} - \sum_{j=3}^{N} y_{2j}c_{2j} = 0,$$

pre-multiplying the equation above by c_{12} we obtain

$$y_{12} - \sum_{j=3}^{N} y_{2j}c_{12}^T c_{2j} = 0. \tag{36}$$

Assume that one of the components $y_{12}, y_{23},\ldots,y_{2N}$ is nonzero, i.e. it is positive. Since $c_{12}^T c_{2j} < 0,\ j=3,\ldots,N$, this implies

$$y_{12} - \sum_{j=3}^{N} y_{2j}c_{12}^T c_{2j} > 0,$$

which is a contradiction with equality (36). Consequently,

$$y_{12} = 0, \quad y_{2j} = 0 \qquad j = 3,\ldots,N. \tag{37}$$

Actually we can write $A^T y = 0$ as follows,

$$\sum_{i=1}^{m-1} y_{im}c_{im} - \sum_{j=m+1}^{N} y_{mj}c_{mj} = 0, \qquad m = 2,\ldots,N. \tag{38}$$

This implies that

$$\sum_{i=1}^{m-1} y_{im} c_{1m}^T c_{im} - \sum_{j=m+1}^{N} y_{mj} c_{1m}^T c_{mj} = 0, \qquad m = 2,\ldots,N. \tag{39}$$

To use induction with (37), we suppose that

$$y_{2m} = \ldots = y_{m-1,m} = 0,$$

which is true when $m = 3$ from (37). This together with (39) imply that

$$y_{1m} - \sum_{j=m+1}^{N} y_{mj} c_{1m}^T c_{mj} = 0.$$

Since $c_{1m}^T c_{mj} < 0$ and $y \succeq 0$, this implies that

$$y_{1m} = 0, \qquad y_{mj} = 0 \qquad j = m+1,\ldots,N.$$

Consequently, by induction we prove that

$$y = 0.$$

This shows that $y = 0$ that implies that $\mathcal{F}_1$ is a nonempty set. ■

Remark 4 Satisfaction of inequality (29) is a necessary condition for Problem 2 to have a feasible solution. Consequently, Lemma 2 guides us to a consistent generation of the separating planes. Yet another requirement that must be satisfied during the generation of these planes is that two consecutive separating planes for a pair of spacecraft must not be antiparallel, i.e.,

$$c_{ij}[k+1]^T c_{ij}[k] \neq -1 \qquad j > i,\ i = 1,\ldots,N-1,\ k = 1,\ldots,n. \tag{40}$$

Therefore, the following condition is imposed, which requires a maximum angle $\hat{\theta}$ between two consecutive separating planes,

$$c_{ij}[k+1]^T c_{ij}[k] \geq \cos\hat{\theta} \qquad j > i,\ i = 1,\ldots,N-1,\ k = 1,\ldots,n, \tag{41}$$

where we typically choose $\hat{\theta} \in [30^\circ,\ 90^\circ]$. ◇

The following is the heuristic procedure used in this paper to generate the separating planes such that the inequalities (32) and (41) are satisfied. Here we assume that the initial and final desired states of the spacecraft do not violate the collision avoidance constraints.

Procedure 1 1. Obtain a solution, $\tilde{x}_j[k]$, $k = 1,\ldots,n$, $j = 1,\ldots,N$, and corresponding position vectors $\tilde{r}_j$ for Problem 2 without imposing the collision avoidance constraints (14).

2. Let

$$c_{1j}[1] = \frac{r_j[1] - r_1[1]}{\|r_j[1] - r_1[1]\|} \qquad j = 2,\ldots,N.$$

3. For $k > 2$, given $c_{12}[k-1],\ldots,c_{1N}[k-1]$, generate the separating planes $c_{12}[k],,\ldots,c_{1N}[k]$ as follows:

   a) Let

$$\tilde{c}_{1j} = \begin{cases} \dfrac{r_j[k] - r_1[k]}{\|r_j[k] - r_1[k]\|} & j = 2,\ldots,N., \quad \text{if } r_j[k] - r_1[k] \neq 0; \\ c_{1j}[k-1], & \text{otherwise.} \end{cases}$$

   b) Let $\theta_{1j} = \cos^{-1}\left(c_{1j}[k-1]^T \tilde{c}_{1j}[k]\right)$, then

$$c_{1j}[k] = \begin{cases} \tilde{c}_{1j}[k], & \text{if } \theta_{1j} \geq \hat{\theta}; \\ \text{rotate}\,(\tilde{c}_{1j}, \hat{u}_{1j}, \theta_{1j} - \hat{\theta}), & \text{otherwise.} \end{cases}$$

where $\hat{\theta}$ is the maximum allowed angle separation between two consecutive separating planes,

$$\hat{u}_{1j} = \frac{\tilde{c}_{1j}[k] \times c_{1j}[k-1]}{\left\| \tilde{c}_{1j}[k] \times c_{1j}[k-1] \right\|}$$

and *rotate* is the rotation operator defined for any $v, u \in \mathbb{R}^3$ and angle α as

$$\text{rotate}(v, u, \alpha) = \cos\alpha\, v + (1 - \cos\alpha)\,(v^T u)u + \sin\alpha\,(u \times v).$$

4. Given $c_{1j}[k]$ $k = 1, \ldots n$, $j = 2, \ldots, N$, generate the rest of the separating planes for $k = 1, \ldots, n$ $j > i$, $i = 2, \ldots, N-1$. Let $v = c_{1j}[k] - c_{1i}[k]$, $\alpha = \cos^{-1}\left(c_{1i}[k]^T c_{ij}[k-1]\right)$, and

$$u_r = \frac{c_{1i}[k] \times c_{ij}[k-1]}{\left\| c_{1i}[k] \times c_{ij}[k-1] \right\|}, \qquad \theta_r = \min\left\{\pi - \hat{\theta} - \alpha, \frac{\pi}{2}\right\},$$

$$c_{ij}[k] = \begin{cases} \dfrac{v}{\|v\|}, & \text{if } v^T c_{ij}[k-1] \geq \|v\| \cos\hat{\theta}; \\ \begin{cases} -c_{1i}[k], & \text{if } \pi - \alpha \leq \hat{\theta}; \\ \text{rotate}(-c_{1i}[k], \theta_r, u_r), & \text{otherwise.} \end{cases}, & \text{otherwise.} \end{cases}$$

□

Remark 5 In step 4 of Procedure 1, the primary objective in generating the separating planes is to guarantee $c_{ij}[k]^T c_{1i}[k] < 0$. If $c_{ij}[k]^T c_{1i}[k] < 0$ and $c_{ij}[k]^T c_{ij}[k-1]^T \geq \cos\hat{\theta}$ can not be simultaneously satisfied, the angle separation between two consecutive planes $c_{ij}[k-1]$ and $c_{ij}[k]$ is minimized. ◇

Now, we describe the solution algorithm.

Algorithm 1 *Given T, Δt, $\hat{\theta}$, $x_j(0)$, $x_j(T)$, and the parameters describing the constraints in Problem 2.*

1. *Generate the separating planes by using Procedure 1.*
2. *Solve Problem 2 by using a SOCP solver [22, 23].*

IV. Simulations

In this section, we present several illustrative simulations of three and seven spacecraft formation reconfiguration. In each simulation, we assume that $\Delta t = 60$ seconds, $n = 30$ samples, and $\theta = \frac{\pi}{3}$ radians. Simulation results presented include plots of the relative trajectory, the inter-spacecraft distance, and the control input.

When three spacecraft are used, each starts on the x-axis, with spacecraft (SC) 1 at the origin and SC 2 and SC 3 at a distance of $10m$ away on either side. The objective of the computed maneuver is for SC 2 to switch places with SC 3. In the final example of seven spacecraft, SC 1 again starts at the origin and the remaining spacecraft are located $10m$ away on the corresponding sides of each axis. Again, the spacecraft switch places: SC 2 with SC 3, SC 4 with SC 5, and SC 6 with SC 7. The maximum available control acceleration for each spacecraft is $1\,cm/s^2$, and the spacecraft must stay at least $4m$ away from each other, i.e., the collision avoidance radius, R, is 4 meters.

A. Simulation 1: 3 Spacecraft in Deep Space, Minimum Fuel

First we consider a reconfiguration of three-spacecraft formation in deep space, and obtain guidance trajectories via minimizing the total fuel. In Figure 1, the relative trajectory graph shows the two switching spacecraft accomplishing the required maneuver by moving symmetrically in the X-Z plane around SC 1. Here X, Y, Z axes correspond to the first, the second, and the third entries in the position vector. Figure 2 shows the four burns required for the maneuver. Only the plot for SC 2 is shown, but the plot for SC 3 is the same. Figure 3 shows that the inter-spacecraft distance never reaches below 4 m.

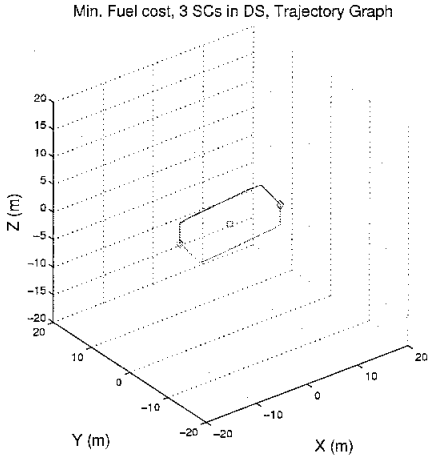

Figure 1. Relative trajectory of SC 2 and SC 3 around SC 1 for simulation 1

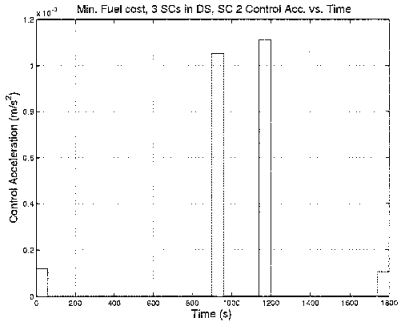

Figure 2. Control Accelerations of SC 2 and SC 3 for simulation 1

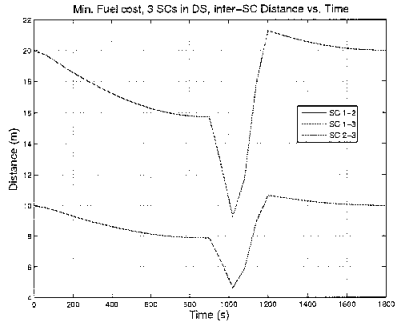

Figure 3. Inter-spacecraft distances among formation for simulation 1

B. Simulation 2: 3 Spacecraft in Deep Space, Minimum Energy

In this section, we resolve the problem described in Section B with a cost of total energy rather than fuel. Figures 4, 5, and 6 show that the trajectories and the control input are smoother functions of time when compared to minimum fuel solution, which are typical characteristics of minimum energy versus minimum fuel trajectories.

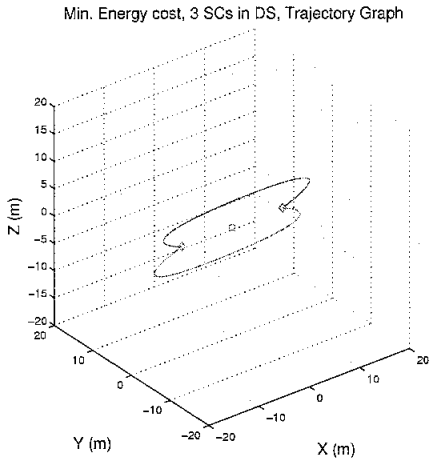

Figure 4. Relative trajectory of SC 2 and SC 3 around SC 1 for simulation 2

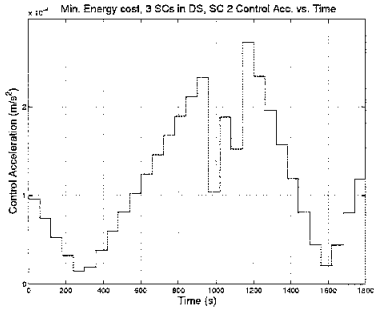

Figure 5. Control Accelerations of SC 2 and SC 3 for simulation 2

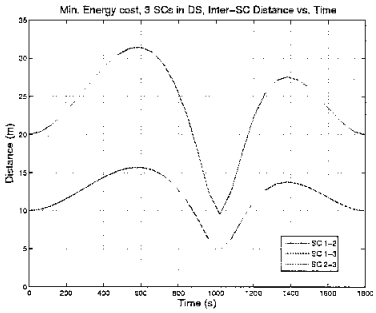

Figure 6. Inter-spacecraft distances among formation for simulation 2

C. Simulation 3: 3 Spacecraft in LEO, Minimum Fuel

In this section, we resolve the problem described in Section B in LEO rather than in deep space with $\omega = 1.11 \times 10^{-3}\, 1/s^2$. Figures 7, 8, and 9 give the simulation result that show more circular trajectories with out of plane components due to the coupled dynamics between the x and y axes in LEO when compared to the deep space results.

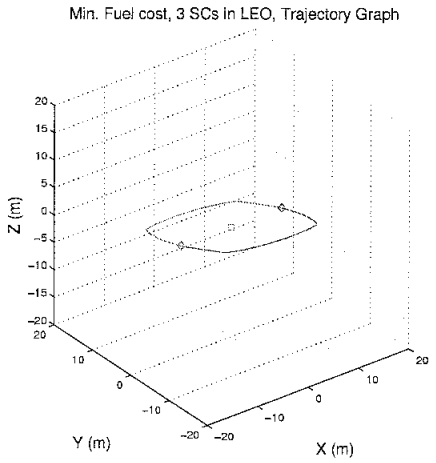

Figure 7. Relative trajectory of SC 2 and SC 3 around SC 1 for simulation 3

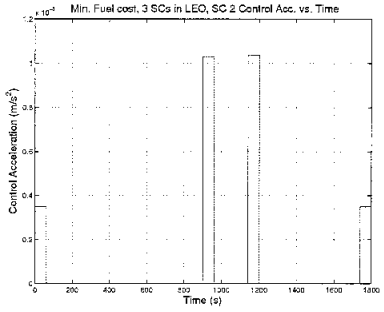

Figure 8. Control Accelerations of SC 2 and SC 3 for simulation 3

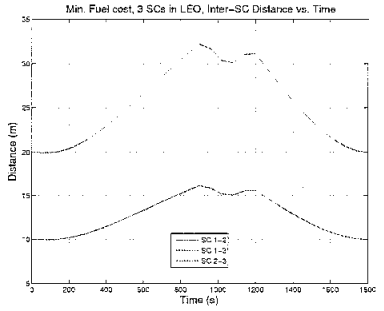

Figure 9. Inter-spacecraft distances among formation for simulation 3

D. Simulation 4: 7 Spacecraft in Deep Space, Minimum Fuel

This section presents simulation results for a seven spacecraft reconfiguration in deep space. The trajectories are obtained by minimizing the total fuel usage. Figures 10, 11, and 12 present the simulation results, and show that all the state and control constraints are satisfied. Again, the control acceleration plots for SC 2 - SC 7 are the same, so only that of SC 2 is shown.

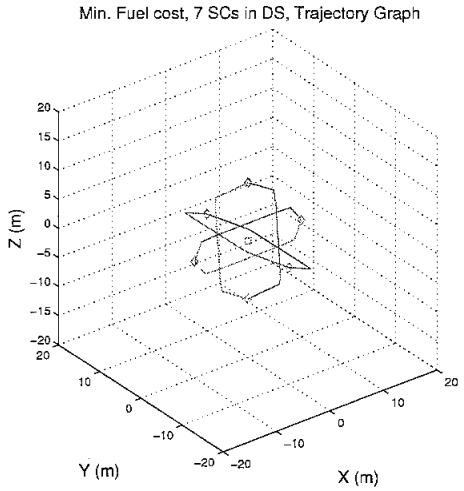

Figure 10. Relative trajectory of SC 2 - SC 7 around SC 1 for simulation 4

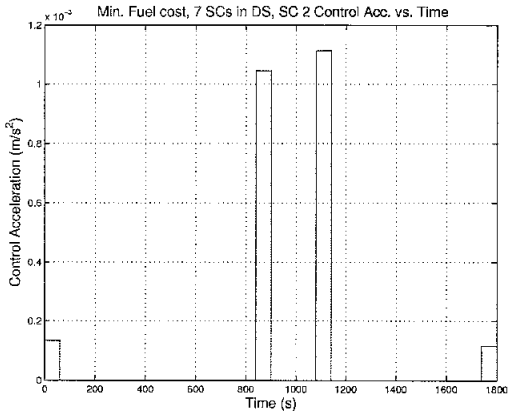

Figure 11. Control Accelerations of SC 2 and SC 3 for simulation 4

V. Conclusions

In this paper, we presented a formation reconfiguration algorithm for spacecraft in both deep space and low-earth-orbit. A heuristic is introduced to convexify the formation reconfiguration problem with collision avoidance constraints. Then, a discrete version of the problem, which is a second order cone program, is solved via readily available algorithms [22, 23] that compute the global optimum with a deterministic stopping criteria and prescribed level of accuracy. The main convex relaxation is introduced via the separating planes that impose the collision avoidance constraint. Analysis results are presented to justify the heuristic used to generate the separating planes. We also

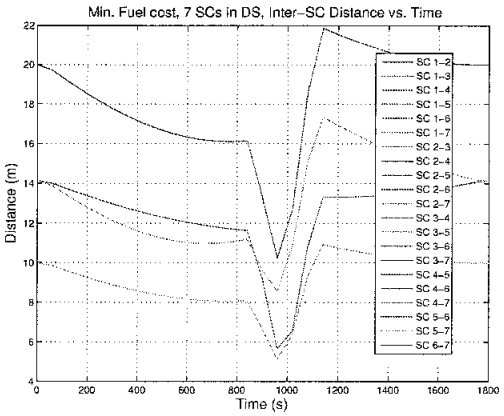

Figure 12. Inter-spacecraft distances among formation for simulation 4

provide additional constraints at the discrete time instances that guarantee the satisfaction of the collision avoidance constraints on the time intervals between two consecutive discrete time samples. Simulation results are presented to demonstrate the algorithm.

The maneuver time is currently treated as an input to the problem but this must be also computed as a part of the trajectory planning problem in an efficient way. The heuristic used in the generation of the separating planes can further be improved by establishing a measure on the size of the feasible domain described by the choice of these planes. Since we are only considering a subset of all feasible solutions, this can lead to considering larger subsets of the feasible set. Our current approach to the translational formation reconfiguration can be complemented by the recent results on the constrained attitude planning based on convex optimization [24, 25]. Furthermore, we can extend this formulation to formations in elliptical low-earth-orbit.

Acknowledgements

This research was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

[1]Scharf, D., Ploen, S., and Hadaegh, F., "A Survey of Spacecraft Formation Flying Guidance and Control (Part I): Guidance," 2003.

[2]et al., K. C., "The Stellar Imager Mission Concept," *Proceedings SPIE in Waikoloa*, Vol. 4854, August, 2002, pp. 293–302.

[3]Lawson, P. R., "The Terrestrial Planet Finder," *Proceedings of IEEE Aerospace Conference*, Vol. 101, 2001, pp. 2005–2011.

[4]Scharf, D. P., Hadaegh, F. Y., and Ploen, S. R., "A Survey of Spacecraft Formation Flying Guidance and Control (Part II): Control," *Preceedings of the 2004 American Control Conference*, Vol. 4, June 2004, pp. 2976–2985.

[5]K. C. Gendreau, W. C. Cash, A. F. S. N. W., "The MAXIM pathfinder X-ray interferometry mission," *Proceedings SPIE - The International Society for Optical Engineering*, Vol. 4851, No. 1, 2002, pp. 353–364.

[6]Scharf, D., Hadaegh, F., Rahman, Z., Shields, J., Singh, G., and Wette, M., "An Overview of the Formation and Attitude Control System for the Terrestrial Planet Finder Interferometer," *2nd Int. Symp. on Formation Flying Missions & Technologies*, 2004.

[7]Richards, A., Schouwenaars, T., How, J., and Feron, E., "Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming," *AIAA Journal of Guidance and Control*, Vol. 25, No. 4, 2002, pp. 755–764.

[8]Singh, G. and Hadaegh, F., "Collision Avoidance Guidance for Formation-Flying Applications," *AIAA Guid., Nav.,& Contr. Conf.*, 2001.

[9]Sultan, C., Seereeram, S., Mehra, R. K., and Hadaegh, F. Y., "Energy Optimal Reconfiguration for Large Scale Formation Flying," *Proceedings of the American Control Conference*, June, 2004, pp. 2986–2991.

[10]Sultan, C., Seereeram, S., and Mehra, R. K., "Energy Optimal Multi-Spacecraft Relative Reconfiguration of Deep SPace Formation Flying," *Proceedings of the 43rd IEEE Conference on Decision and Control*, December, 2004, pp. 284–289.

[11]Frazzoli, E., Mao, Z., Oh, J., , and Feron, E., "Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming," *AIAA Journal of Guidance and Control*, Vol. 24, No. 1, 2001, pp. 79–86.

[12]Wang, P. and Hadaegh, F., "Minimum-Fuel Formation Reconfiguration of Multiple Free-Flying Spacecraft," *J. Astro. Sci.*, Vol. 47, No. 1,2, 1999, pp. 77–102.

[13]Campbell, M., "Planning Algorithm for Multiple Satellite Clusters," *J. Guid., Contr., & Dyn.*, Vol. 26, No. 5, 2003, pp. 770–780.

[14]Li, S., Mehra, R., Smith, R., and Beard, R., "Multi-Spacecraft Trajectory Optimization and Control Using Genetic Algorithm Techniques," *IEEE Aero. Conf.*, 2000.

[15]Kim, Y., Mesbahi, M., and Hadaegh, F., "Dual-Spacecraft Formation Flying in Deep Space: Optimal Collision-Free Reconfigurations," *J. Guid., Contr, & Dyn.*, Vol. 26, No. 2, 2003, pp. 375–379.

[16]Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[17]Canny, J., Reif, J., Donald, B., and Xavier, P., "On the Complexity of Kinodynamic Planning," *29th IEEE Annual Symp. on Foundations of Computer Science*, 1988, pp. 306–316.

[18]Garcia, I. and How, J., "Trajectory Optimization for Satellite Reconfiguration Maneuvers with Position and Attitude Constraints," 2005.

[19]Betts, J. T., "Survey of Numerical Methods fro Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[20]Sturm, J. F., "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, Vol. 17, No. 6, 2002, pp. 1105–1154.

[21]Berkovitz, L. D., *Convexity and Optimization in* $\mathbb{R}^n$, John Wiley & Sons, Inc., 2002.

[22]Toh, K., Todd, M., and Tutuncu, R., "SDPT3–a MATLAB software package for semidefinite programming," *Optimization Methods and Software*, Vol. 11, No. 1, 1999, pp. 545–581.

[23]Sturm, J. F., "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, Vol. 11, No. 1, 1999, pp. 625–653.

[24]Kim, Y. and Mesbahi, M., "Quadratically Constrained Attitude Control via Semidefinite Programming," *IEEE Transactions on Automatic Control*, Vol. 49, No. 5, 2004, pp. 731–735.

[25]Kim, Y., Mesbahi, M., Singh, G., and Hadaegh, F. Y., "On the constrained attitude control problem," *AIAA Guidance, Navigation, and Control Conference, Providence, RI*, August, 2004.