Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming

Arthur Richards,* Tom Schouwenaars,† Jonathan P. How,‡ and Eric Feron§
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

A method for finding fuel-optimal trajectories for spacecraft subjected to avoidance requirements is introduced. These include avoidance of collisions with obstacles or other vehicles and prevention of thruster plumes from one spacecraft impinging on another spacecraft. The necessary logical constraints for avoidance are appended to a fuel-optimizing linear program by including binary variables in the optimization. The resulting problem is a mixed-integer linear program (MILP) that can be solved using available software. The logical constraints can also be used to express the configuration requirements for maneuvers where only the final relative alignment of the vehicles is important and the assignment of spacecraft within the fleet is not specified. The collision avoidance, trajectory optimization, and fleet assignment problems can be combined into a single MILP to obtain the optimal solution for these maneuvers. The MILP problem formulation, including these various avoidance constraints, is presented, and then several examples of their application to spacecraft maneuvers, including reconfiguration of a satellite formation and close inspection of the International Space Station by a microsatellite, are shown. These examples clearly show that the trajectory design methods presented are particularly well suited to proposed formation flying missions that involve multiple vehicles operating in close proximity.

Nomenclature

G	=	number of global configurations available for end states
M	=	large number for logical constraints
N	=	number of dimensions
P	=	plume length
T	=	number of time steps
u	=	control input
V	=	number of vehicles
W	=	plume width
x	=	vehicle state

Subscripts

g	=	global configuration for final states
i	=	time step
l	=	obstacle
n, m	=	axes in some orthogonal coordinate frame
p, q	=	vehicles
r	=	position within final configuration

I. Introduction

AUTONOMOUS formation flying of satellite clusters has been identified as an enabling technology for many future NASA and U.S. Air Force missions.[1-4] The use of fleets of smaller satellites instead of a single monolithic satellite offers improved science return through longer baseline observations, enables faster ground track repeats, and provides a high degree of redundancy and reconfigurability in the event of a single vehicle failure. These bene-fits can be achieved at the expense of more stringent requirements on fleet coordination, high-level mission management, and fault detection.[5,6] This paper is concerned with the trajectory-planning aspects of mission management. A method is described for designing fuel-optimal maneuvers that accounts for the presence of other vehicles. The approach builds on recent research on the use of linear programming (LP) to design fuel-optimal solutions to spacecraft maneuvering problems.[5,7,8] In particular, this paper shows how integer constraints can be added to the linear program to account for three extensions of the basic maneuvering problem: collision avoidance, plume impingement avoidance, and configuration selection.

A common feature of these problems is a combinatorial aspect to the optimization. This results from the nonconvex nature of the feasible solution space. For example, when a vehicle approaches an obstacle in two-dimensional space, it can choose to pass on either the left or right, each option leading to a further path-planning subproblem. Including these choices makes these trajectory design problems intrinsically difficult to solve: Path-planning in the presence of obstacles has been shown to be *NP*-complete.[9]

The approach presented in this paper formulates the problem as a mixed-integer linear program (MILP). This is a special case of a linear program in which some variables are constrained to take only integer values. In particular, we use binary variables, taking only the values 0 or 1. Constraints on such variables enable the inclusion of logical expressions in the optimization,[10,11] encoding the combinatorial part of the problem. MILPs are also NP complete,[12] but can be solved in many instances using a branch-and-bound algorithm. A key advantage of writing the trajectory optimization in the MILP form is the existence of highly optimized, commercially available software that can be used to solve these mixed-integer linear problems. These codes were developed to solve MILPs in the field of operations research, such as airline scheduling.[13] CPLEX optimization software[14] is used to solve the MILPs in this paper, although various other options exist. CPLEX implements the branch-and-bound algorithm in conjunction with many adjustable heuristics, allowing quite large problems to be solved in practical computation times.

The MILP form of the trajectory optimization problems is linear throughout, and so the method is immune to issues of local minima, and global optimal solutions are found. Because all of the constraints in the MILP problem must be linear, it is necessary to represent the system dynamics model in linear form. Spacecraft avoidance problems are well suited to this approximation. Vehicles typically operate in close proximity, performing maneuvers over relatively

*Research Assistant, Space Systems Laboratory; arthurr@mit.edu.
†Research Assistant, Laboratory for Information and Decision Systems; toms@mit.edu.
‡Associate Professor, Space Systems Laboratory; jhow@mit.edu. Senior Member AIAA.
§Associate Professor, Laboratory for Information and Decision Systems; feron@mit.edu. Senior Member AIAA.

short timescales of the order of an orbit. Under these circumstances, the linearized Hill's equations (see Ref. 15) provide a good model of the relative vehicle dynamics.[16]

Depending on the nature of the trajectory problem, various other techniques can be used to simplify the MILP and accelerate the solution process. These methods involve using prior knowledge of the solution to identify redundant or inactive constraints before solving the problem. For example, in many cases it is possible to predict that the optimal solution will only involve firing at the beginning and end of the maneuver with a coasting phase between, which is the familiar bang–off–bang result from optimal control theory. With this prior observation, constraints on thruster firing during the predicted coasting phase can be removed. The resulting trajectory can then be postanalyzed to confirm that all avoidance constraints are satisfied at every time step. Examples of this approach and other useful simplifications are discussed in detail in the paper.

In all cases, a single optimization problem is solved to generate the minimum-fuel trajectory and associated control inputs to complete the maneuver in a specified time. This may be thought of as a reference input for the system to follow. In practice, there will be unmodeled disturbances, measurement uncertainty, and mismatch between the real dynamics and the model used for planning. Ongoing research involves the incorporation of an MILP trajectory planner in a real-time, predictive control scheme to reduce the impact of the unmodeled effects, but this is beyond the scope of this paper.

The following subsections present each of the trajectory design problems of interest, including a discussion of the motivation and previous work in each case. The "Problem Formulation" section presents the equations for the constraints to be added in each case. Examples of the results of these techniques applied to representative problems are shown in the "Application Examples" section. This includes computation time results and a discussion of problem-specific simplifications to the formulation.

A. Collision Avoidance

Many approaches have been investigated for solving the problem of trajectory optimization with collision avoidance for dynamic systems. For example, randomized searches[17] were developed to find feasible paths through fields of obstacles rapidly. In reactive schemes,[18] vehicles fly a nominal trajectory and perform predetermined evasive maneuvers when conflicts are detected. For aircraft, the path-planning problem can be reduced to a single heading change decision,[19] which greatly simplifies the global trajectory optimization. Potential field modeling[20] has also received significant attention. The approach typically includes an additional proximity penalty in the cost function in the optimization to account for the collision constraints. Other approaches use splines[21] and lower-dimensional representations[22] of nonlinear systems to reduce the solution space before performing nonlinear optimization.

In the MILP formulation, the objective is to design trajectories that minimize the fuel for the maneuvers. The MILP approach is simplified by the use of discretized, linear dynamics, which is a reasonable approximation for most formation flying spacecraft problems.[16] Integer constraints are appended directly to the fuel-minimization linear program (LP) to enforce collision avoidance at each time step. Thus, in contrast to reactive techniques, collision avoidance is included in the trajectory planning for all vehicles along the entire maneuver, permitting more efficient operation through cooperative maneuvering. Another advantage is the clear tunability of the MILP formulation: The trajectories will avoid collisions by exactly the margin specified in the constraints. This contrasts with the potential function approach in which the cost penalty for proximity typically has to be adjusted to achieve the desired balance between fuel conservation and the avoidance margin.

B. Plume Impingement

The plumes of high-energy gas particles emitted when a thruster is fired can cause adverse effects if they hit another spacecraft. Concerns include degradation of panels, deposition on instruments, and excess structural loading. In particular, loading due to plume

impingement during docking was a design driver for International Space Station solar panels.[23] Instrument and spacecraft damage is a concern for a number of forthcoming formation flying missions,[24,25] particularly those involving interferometry.[4,26] In contrast to the collision avoidance, plume impingement has not been widely studied in connection with path planning. However, it can be included in an MILP using a similar formulation to collision avoidance, forcing other vehicles to remain outside a certain region when a thruster is fired.

C. Final Configuration Selection

For formation flying missions involving multiple identical spacecraft, the arrangement of the vehicles within the formation is typically not important. For example, in a standard formation flying scenario,[4,7] the vehicles will be arranged around a passive aperture, which is a short-baseline, periodic formation configuration that provides good, distributed Earth imaging while reducing the tendency of the vehicles to drift apart.[16,27−29] Changing the viewing mode of the fleet could require a change in the formation configuration, which involves relatively expensive maneuvers from one aperture to another. The MILP formulation can be extended to include the assignment of vehicle locations on the new aperture. The solution of the MILP then gives the reconfiguration maneuver with the lowest overall fuel cost.

Previous approaches to this reconfiguration problem include separating the reconfiguration into permutation maneuvers within groups of satellites.[30] Another method[7] computes the costs for many maneuvers and then selects the one that gives the lowest overall cost. These approaches attempt to decouple the problems of configuration selection and trajectory planning. Once the trajectory costs have been calculated for each configuration option, the selection is a linear assignment problem, which is readily solved using standard LP tools.[31]

In contrast, the MILP approach in this paper retains the inherent coupling of the fuel minimization and configuration selection by including both in a single optimization. The combinatorial constraints for the configuration selection can readily be included in an MILP as a modification of the terminal constraints in the LP problem. Another advantage is that the optimal MILP solution will satisfy all collision avoidance and impingement constraints, which cannot be directly enforced in the decoupled approaches.

II. Problem Formulation

This section presents the details of the MILP formulations to solve the various trajectory optimization problems.

A. Basic LP

The core of the optimization is to choose discrete state values x_{ip} for each vehicle p and time step $i \in [0, \ldots, T]$ and the corresponding input values u_{ip}. The state at the first time point is constrained to be the specified starting conditions

$$x_{0p} = x_{S_p} \tag{1}$$

where x_{S_p} is the initial state vector for the pth vehicle. Similarly, the state at the final time point is fixed at the specified end conditions

$$x_{Tp} = x_{F_p} \tag{2}$$

where x_{F_p} is the final state vector for the pth vehicle. The states at intermediate points of time must be consistent with the system dynamics

$$x_{(i+1)p} = Ax_{ip} + Bu_{ip} \tag{3}$$

where A and B are a discretized form of the continuous system dynamics. For simplicity, the constraints show the same dynamics for all vehicles, but it is straightforward to modify the formulation to account for heterogenous vehicles. For the problems of interest in this paper, in which the spacecraft are in close proximity on similar

orbits, the most common approximation for spacecraft dynamics are the linearized Hill's equations (see Ref. 15)

$$
\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1/m & 0 & 0 \\ 0 & 1/m & 0 \\ 0 & 0 & 1/m \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{4}
$$

where the x coordinate is in the radial direction, the y coordinate is in the in-track direction, and the z component is in the out-of-plane direction.[15] The spacecraft mass is m and the natural frequency of the reference orbit is ω. The corresponding state and input vectors are $\boldsymbol{x} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T$ and $\boldsymbol{u} = [u_x \ u_y \ u_z]^T$. These equations can be discretized resulting in the form in Eq. (3) by assuming that the thrust $\boldsymbol{u}_{ip}$ is applied continuously throughout the time step. Other discretization methods could be used, such as in impulsive input at the beginning of each time step, to suit the problem at hand. Note that other linearized models of the relative dynamics exist and can also be used in this optimization framework.[32]

The state and input vectors are confined to lie within specified, symmetrical limits

$$
-x_{\max_n} \leq x_{ipn} \leq x_{\max_n} \tag{5}
$$

$$
-u_{\max_n} \leq u_{ipn} \leq u_{\max_n} \tag{6}
$$

where x_{ipn} and u_{ipn} denote the nth component of the state and thrust vectors, respectively, for the pth vehicle at the ith time step. The input limit is typically used to represent the limited force available from each thruster. The state limit has no such significance, but is necessary to bound the problem and ensure the optimization has a unique solution.

The objective is to minimize the total fuel consumption of all vehicles in the problem. For a total of V vehicles moving in N-dimensional space over T time steps, the cost function is

$$
J = \sum_{i=0}^{T-1} \sum_{p=1}^{V} \sum_{n=1}^{N} |u_{ipn}| \tag{7}
$$

This piecewise linear function can be converted to a linear function using slack variables.[33]

In summary, the minimum-fuel path-planning LP, neglecting avoidance so far, is

$$
\min_{u,x} J = \sum_{i=0}^{T-1} \sum_{p=1}^{V} \sum_{n=1}^{N} |u_{ipn}| \tag{8}
$$

such that

$$
\forall p \in [1, \ldots, V], \qquad \boldsymbol{x}_{(i+1)p} = \boldsymbol{A}\boldsymbol{x}_{ip} + \boldsymbol{B}\boldsymbol{u}_{ip}, \qquad \forall i \in [0 \ldots T-1]
$$

$$
\text{and} \qquad \boldsymbol{x}_{0p} = \boldsymbol{x}_{S_p}
$$

$$
\text{and} \qquad \boldsymbol{x}_{Tp} = \boldsymbol{x}_{F_p}
$$

$$
\text{and} \quad -x_{\max_n} \leq x_{ipn} \leq x_{\max_n}
$$
$$
\forall i \in [0 \ldots T-1], \qquad \forall n \in [1 \ldots N]
$$

$$
\text{and} \quad -u_{\max_n} \leq u_{ipn} \leq u_{\max_n}
$$
$$
\forall i \in [0 \ldots T-1], \qquad \forall n \in [1 \ldots N] \tag{9}
$$

All of the problems considered in this paper involve the linear program shown, subject to additional mixed-integer constraints to enforce various forms of avoidance. A second formulation is given later in the paper that replaces the final state constraints with a more flexible formulation that can be used to select the final configuration as part of the optimization.

B. Obstacle Avoidance

This section presents the additional constraints on the LP to avoid static obstacles.[34,35] The obstacles can be modeled in this framework as convex polygons of any number of sides, but, to simplify the presentation, the results in this paper only use rectangles. Collisions are prevented by ensuring that the vehicle trajectories lie outside the obstacles at each of the discrete time points. Note that it is feasible for the trajectory to cut into obstacles in between the discrete time points. It is, therefore, necessary to enlarge the obstacle models and select the time-step length such that these incursions cannot intersect the real obstacles.

For visualization, the constraints are first derived for the two-dimensional case. The location of the rectangular obstacle is defined by its lower left-hand corner $(x_{\min}, y_{\min})$ and its upper right-hand corner $(x_{\max}, y_{\max})$. At all time steps i, the position (x_i, y_i) of the vehicle must lie in the area outside of the obstacle. This requirement can be formulated as the set of conditions

$$
x_i \leq x_{\min}
$$
$$
\text{or} \quad x_i \geq x_{\max}
$$
$$
\text{or} \quad y_i \leq y_{\min}
$$
$$
\text{or} \quad y_i \geq y_{\max} \tag{10}
$$

These constraints can be transformed into a mixed-integer form by introducing binary variables.[10,12] A set of binary variables a_k are added to the problem for each pair of vehicles at each time step. Additional subscripts on a_k for time steps, vehicles, and obstacles are omitted for clarity, but they will be included later. Let M be an arbitrary positive number, larger than any distance in the problem. The constraints in Eq. (10) are represented by the following mixed-integer constraints:

$$
x_i \leq x_{\min} + Ma_1
$$
$$
\text{and} \qquad -x_i \leq -x_{\max} + Ma_2
$$
$$
\text{and} \qquad y_i \leq y_{\min} + Ma_3
$$
$$
\text{and} \qquad -y_i \leq -y_{\max} + Ma_4
$$
$$
\text{and} \qquad \sum_{k=1}^{4} a_k \leq 3 \tag{11}
$$

Note that, if $a_k = 0$, then the kth constraint from Eq. (10) is enforced. However, if $a_k = 1$, then that constraint is relaxed because the M term moves the upper bound beyond the solution space. The last *and* constraint in Eq. (11) ensures that no more than three constraints from Eq. (10) are relaxed, and, hence, at least one of the original *or* constraints is satisfied.

This process can be extended to a general number of dimensions (in practice, $N = 2$ or 3), time steps, vehicles, and obstacles. The position of vehicle p at time step i is the vector $\boldsymbol{x}_{ip} = [x_{ip1}, \ldots, x_{ipn}]^T$. The vertex of obstacle l with the minimum value of each coordinate is at position $\boldsymbol{L}_l$. This is the bottom-left-hand corner in the two-dimensional case. Its vertex with the maximum of each coordinate is at $\boldsymbol{U}_l$. The binary variables a_{iplk} are the switches, with $k \in [1, \ldots, 2N]$, corresponding to being on one of two sides of the obstacle in each of N dimensions. The complete formulation is

$$
\forall p, \forall l, \forall i \in [1, \ldots, T-1]: \qquad x_{ipn} \geq U_{ln} - Ma_{ipln} \qquad \forall n
$$

$$
\text{and} \qquad x_{ipn} \leq L_{ln} + Ma_{ipl(n+N)} \qquad \forall n
$$

$$
\text{and} \qquad \sum_{k=1}^{2N} a_{iplk} \leq 2N - 1 \tag{12}
$$

These become additional constraints on the LP in Eqs. (8) and (9). The binary variables a become extra decision variables in the problem. The new constraints in Eq. (12) are linear in the decision variables, and so the new problem is an MILP.

C. Collision Avoidance

This section derives constraints to avoid collisions between different vehicles.[34,35] Every pair of vehicles must be at least a specified distance apart in each direction at each time step. This corresponds to the enforcement of a rectangular exclusion region around each vehicle.

As in the preceding section, the constraints are first shown in two dimensions for clarity. Let the safety distances in the X and Y directions be denoted by r_x and r_y, respectively. Denote the positions of two different vehicles p and q at time step i as (x_{ip}, y_{ip}) and (x_{iq}, y_{iq}), respectively. The constraints can be written as

$$x_{ip} - x_{iq} \geq r_x,$$

$$\text{or} \quad x_{iq} - x_{ip} \geq r_x$$

$$\text{or} \quad y_{ip} - y_{iq} \geq r_y,$$

$$\text{or} \quad y_{iq} - y_{ip} \geq r_y \qquad (13)$$

As in the preceding section, these can be converted to the more useful *and* constraints by introducing binary variables, giving

$$x_{ip} - x_{iq} \geq r_x - Mb_1$$

$$\text{and} \quad x_{iq} - x_{ip} \geq r_x - Mb_2$$

$$\text{and} \quad y_{ip} - y_{iq} \geq r_y - Mb_3$$

$$\text{and} \quad y_{iq} - y_{ip} \geq r_y - Mb_4$$

$$\text{and} \quad \sum_{k=1}^{4} b_k \leq 3 \qquad (14)$$

where the vehicle and time-step subscripts on the binary variables b have been omitted for clarity. M is the same large number used in the preceding section.

This is extended to the general case using the same notation as before. The safety avoidance distance in direction n is r_n. The condition $q > p$ avoids duplication of the constraints on the positions of pairs of vehicles:

$$\forall p, q | q > p : \forall i \in [1, \ldots, T-1] :$$

$$x_{ipn} - x_{iqn} \geq r_n - Mb_{ipqn}, \qquad \forall n$$

$$\text{and} \quad x_{iqn} - x_{ipn} \geq r_n - Mb_{ipq(n+N)}, \qquad \forall n$$

$$\text{and} \quad \sum_{k=1}^{2N} b_{ipqk} \leq 2N - 1 \qquad (15)$$

These constraints can also be added to the LP of Eqs. (8) and (9). The binary variables b become additional decision variables in the optimization problem.

D. Plume Avoidance for Vehicles

This section develops the formulation to prevent plume impingement by one vehicle on another.[36] The plumes extend in discrete directions from the vehicle, which assumes that the thrusters are aligned with the axes. The inclusion of vehicle attitude within this formulation would complicate it considerably and is the subject of ongoing research. As in the case of obstacle avoidance, the plume region could be represented by any convex polygon, but a rectangular shape is used here for simplicity. All other vehicles are required to remain outside this region while the thruster is firing. Conversely, a spacecraft cannot fire its thrusters if the resulting plume would impinge on another vehicle.
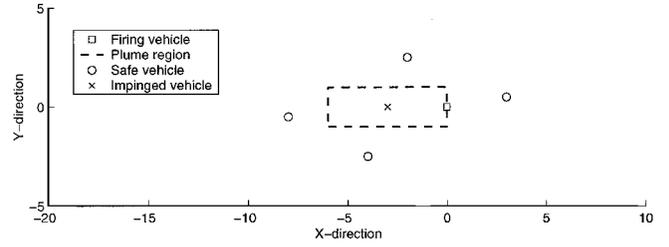
a) Exclusion region for plume impingement on vehicles
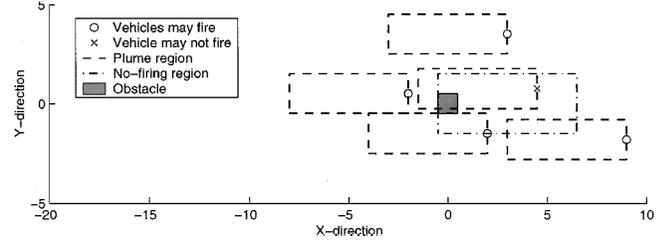
b) Exclusion region for plume impingement on obstacles

Fig. 1 Plume impingement regions in two dimensions.

Once again, the constraints are first developed in two dimensions to simplify the visualization. Figure 1a shows the modeled impingement region extending in the $-X$ direction. The vehicles shown by O are clear of the plume because they are outside the impingement region. The vehicle shown by $\times$ will be impinged if the thruster is firing in the $-X$ direction, generating thrust in the $+X$ direction, but could still escape impingement if the thruster is not firing. Thus, for the plume shown, there are five ways to avoid impingement: Either the thruster is not firing or the target vehicle is clear of the box in any of the four directions. These can be represented by the *or* constraints

$$u_{xip} \leq 0$$

$$\text{or} \quad x_{ip} - x_{iq} \geq P$$

$$\text{or} \quad x_{iq} - x_{ip} \geq 0$$

$$\text{or} \quad y_{ip} - y_{iq} \geq W$$

$$\text{or} \quad y_{iq} - y_{ip} \geq W \qquad (16)$$

where W is the plume half-width, P is the plume length, and u_{xip} is the thrust from vehicle p in the X direction at time step i. Thus, for the situation shown in Fig. 1, vehicle p is marked by the $\square$ and any of the others may be vehicle q. The vehicles marked by O each satisfy one of the last four constraints. The vehicle marked by $\times$ satisfies none of the last four constraints because it is inside the plume region, but it will not be impinged if the first constraint is satisfied.

As shown earlier, these constraints can be converted to the more convenient *and* form using binary variables

$$u_{xpi} \leq Mc_0$$

$$\text{and} \quad x_{ip} - x_{iq} \geq P - Mc_1$$

$$\text{and} \quad x_{iq} - x_{ip} \geq -Mc_2$$

$$\text{and} \quad y_{ip} - y_{iq} \geq W - Mc_3$$

$$\text{and} \quad y_{iq} - y_{ip} \geq W - Mc_4$$

$$\text{and} \quad \sum_{k=0}^{4} c_k \leq 4 \qquad (17)$$

To extend the constraints to the most general case of dimensions and vehicles, denote the thrust vector for vehicle p at time-step i as $\boldsymbol{u}_{ip} = [u_{ip1}, \ldots, u_{ipN}]^T$. The formulation for vehicle q to avoid the plumes from forward (positive thrust) thrusters of vehicle p is

$\forall p, q | q \neq p : \forall n : \forall i \in [0, \ldots, T - 1] :$

$$-u_{ipn} \geq -Mc_{ipqn0}^{+}$$

and $\quad x_{ipn} - x_{iqn} \geq P - Mc_{ipqnn}^{+}$

and $\quad x_{iqn} - x_{ipn} \geq -Mc_{ipqn(n+N)}^{+}$

and $\quad x_{ipm} - x_{iqm} \geq W - Mc_{ipqnm}^{+}, \qquad \forall m | m \neq n$

and $\quad x_{iqm} - x_{ipm} \geq W - Mc_{ipqn(m+N)}^{+}, \qquad \forall m | m \neq n$

and $\quad \sum_{k=0}^{2N} c_{ipqnk}^{+} \leq 2N \qquad (18)$

Similarly, the constraints for avoiding plumes from reverse thrusters are

$\forall p, q | q \neq p : \forall n : \forall i \in [0, \ldots, T - 1] :$

$$u_{ipn} \geq -Mc_{ipqn0}^{-}$$

and $\quad x_{ipn} - x_{iqn} \geq -Mc_{ipqnn}^{-}$

and $\quad x_{iqn} - x_{ipn} \geq P - Mc_{ipqn(n+N)}^{-}$

and $\quad x_{ipm} - x_{iqm} \geq W - Mc_{ipqnm}^{-}, \qquad \forall m | m \neq n$

and $\quad x_{iqm} - x_{ipm} \geq W - Mc_{ipqn(m+N)}^{-}, \qquad \forall m | m \neq n$

and $\quad \sum_{k=0}^{2N} c_{ipqnk}^{-} \leq 2N \qquad (19)$

When these constraints are added to the LP in Eqs. (8) and (9), the variables c become additional decision variables.

E. Plume Avoidance for Obstacles

This section derives the constraints to prevent vehicles from firing their thrusters when they would impinge on stationary obstacles. In spacecraft applications, it is often useful to represent large, complicated structures, such as a space station, as combinations of fixed obstacles. The constraints are similar to the ones in the preceding section.

Figure 1b shows how vehicles may avoid impinging on an obstacle in two dimensions. The vehicles marked by O are all free to fire as shown because their plumes will not contact the obstacle. The vehicle marked by × will impinge on the obstacle if it fires in the direction shown. It is clear from this example that there is a region around the obstacle in which the vehicles cannot emit plumes in the $-X$ direction without impinging. As before, there are five ways to avoid impinging: to be outside the box, or not to fire. These can be written as the following *or* group using the corners of the obstacle, as shown in the obstacle avoidance section:

$$u_{xip} \leq 0$$

or $\quad x_{ip} - x_{\max} \geq P$

or $\quad x_{\min} - x_{ip} \geq 0$

or $\quad y_{ip} - y_{\max} \geq W$

or $\quad y_{\min} - y_{ip} \geq W \qquad (20)$

As before, binary variables are added to convert to *and* form and extend to the general case. The constraints for forward thrust are

$\forall p : \forall l : \forall n : \forall i \in [0, \ldots, T - 1] :$

$$-u_{ipn} \geq -Md_{ipqn0}^{+}$$

and $\quad x_{ipn} - U_{ln} \geq P - Md_{ipqnn}^{+}$

and $\quad L_{ln} - x_{ipn} \geq -Md_{ipqn(n+N)}^{+}$

and $\quad x_{ipm} - U_{lm} \geq W - Md_{ipqnm}^{+}, \qquad \forall m | m \neq n$

and $\quad L_{lm} - x_{ipm} \geq W - Md_{ipqn(m+N)}^{+}, \qquad \forall m | m \neq n$

and $\quad \sum_{k=0}^{2N} d_{ipqnk}^{+} \leq 2N \qquad (21)$

and, similarly, for reverse thrust

$\forall p : \forall l : \forall n : \forall i \in [0, \ldots, T - 1] :$

$$u_{ipn} \geq -Md_{ipqn0}^{-}$$

and $\quad x_{ipn} - U_{ln} \geq -Md_{ipqnn}^{-}$

and $\quad L_{ln} - x_{ipn} \geq P - Md_{ipqn(n+N)}^{-}$

and $\quad x_{ipm} - U_{lm} \geq W - Md_{ipqnm}^{-}, \qquad \forall m | m \neq n$

and $\quad L_{lm} - x_{ipm} \geq W - Md_{ipqn(m+N)}^{-}, \qquad \forall m | m \neq n$

and $\quad \sum_{k=0}^{2N} d_{ipqnk}^{-} \leq 2N \qquad (22)$

Again, these can be appended to the LP equations (8) and (9), the binary variables d becoming extra decision variables in the optimization.

F. Final Configuration Selection

The constraint in Eq. (2) enforces a fixed final state for each vehicle. This section generalizes that constraint to the case where each vehicle is assigned a specific final state from a set of possible alternatives.[35] A subset of final states, known as a global configuration, is selected, and spacecraft are assigned to positions within that subset. The selection and assignment are performed within the trajectory optimization to achieve the lowest overall fuel cost. For example, it might be required that the satellites reconfigure so that they are evenly spaced around a given ellipse, forming a passive aperture for a particular interferometry observation. If the spacecraft are assumed to be identical, their ordering around the ellipse is not important. In addition, the rotation of the whole formation around the ellipse is not important. In the MILP formulation, the ellipse is discretized into a set of possible rotation angles for the formation. Each of these is entered as a global configuration, containing the final state for each spacecraft with the formation at that angle. When the resulting MILP is solved, the formation angle and the assignment of spacecraft around the formation are selected within the optimization to give the minimum fuel use in the reconfiguration maneuver.

The final state constraints for this formulation can be thought of as an extensive *or* expression, in which the final states of the vehicles must be one of the available global configurations g, and the vehicles can be distributed in any one of the possible permutations across the terminal states. In more concise terms

$$\forall p : \boldsymbol{x}_{Tp} = \boldsymbol{x}_{Fr}^{g} \qquad (23a)$$

for some

$$g \in [1, \ldots, G] \qquad (23b)$$

where r is the unique position within the formation assigned to vehicle p. With binary variables, these constraints can be expressed as

$$\forall p : \boldsymbol{x}_{Tp} = \sum_{g=1}^{G} \sum_{r=1}^{V} \boldsymbol{x}_{Fr}^{g} f_{pgr} \qquad (24)$$

where binary variable $f_{pgr} = 1$ if vehicle p takes the rth position within the gth global configuration and 0 otherwise. It is then necessary to place the following logical constraints on these variables:

$$\forall p : \sum_{g} \sum_{r} f_{pgr} = 1$$

$$\forall g, \forall p : \sum_{r} f_{pgr} = \sum_{r} f_{rgp}$$

$$\forall g : \sum_{p} \sum_{r} f_{pgr} = V \sum_{r} f_{1gr} \qquad (25)$$

The first constraint ensures that a satellite p chooses exactly one position. The second constraint ensures that different satellites move to different positions r. The third constraint then ensures that the chosen positions belong to only one configuration corresponding to the global configuration g. The right-hand side of this equation equals 0 or V, the number of satellites.

III.   Computation Considerations

This section discusses issues related to the implementation and solution of the optimization problems posed in the preceding section. The methods presented here do not change the basic constraints already described, but assist greatly in obtaining solutions within practical computation times.

A.   Normalization

The constraints shown so far can involve quantities of very different magnitudes. For example, separation distances can be on the order of hundreds of meters, whereas thrust inputs may be a few micronewtons. To improve the numerical conditioning of the problem, the variables are normalized. New normalized states $\hat{x}$ and inputs $\hat{u}$ are related to the full-scale quantities by

$$x = X\hat{x} \tag{26}$$

$$u = U\hat{u} \tag{27}$$

where X and U are diagonal matrices of scaling factors equal to the original upper bounds in Eqs. (5) and (6). The new state and input bounds are given by

$$-1 \le \hat{x}_{ipn} \le 1 \tag{28}$$

$$-1 \le \hat{u}_{ipn} \le 1 \tag{29}$$

Other position quantities such as the obstacle definitions L and U and the plume scales P and W are scaled by the same factors. As a result, all of the decision variables are of the same order of magnitude because the binary variables lie between 0 and 1, by definition. This modification has been found to make significant improvements in computation time in some cases, but no difference in others. Whereas the reason for the improvement is not exactly clear, it appears to be a result of an improvement in the efficacy of the heuristics employed by the CPLEX software to select branching nodes and directions.

B.   Simplifying Approximations

This section describes additional approximations that can be used to simplify the problems given earlier to reduce the solution time. The most useful approach is to use prior knowledge to identify redundant or inactive constraints. These can then be removed from the problem, which typically leads to a faster solution time. Two strategies are presented for identifying these removable constraints, each well suited to a particular class of problems.

1.   Removal of Plume Constraints During Coast

In constrained-input, minimum-fuel problems of the type in Eq. (9), the optimal solution can be shown to consist of firing at the beginning and end of the trajectory, separated by a coasting phase. This is more commonly known as a bang–off–bang trajectory.[37] In certain problems, where the maneuvering space is large compared to the avoidance regions, it is possible to predict that the optimal solution including avoidance constraints will still be of a bang–off–bang form. Therefore, during the coasting phase, the plume constraints will be inactive.

This prior knowledge can be exploited by omitting some of the plume constraints during the anticipated coasting phase before solving the problem. The reduced number of binary variables and constraints typically leads to a faster solution time. However, it is then necessary to verify that no plume impingement occurred at the steps where the constraints were removed. This can be done quickly, and if no impingement is found, the result is also the optimal solution to the completely constrained problem. Note that this behavior exemplifies an NP-complete problem: The optimality of a candidate

solution can be verified in polynomial time, but the global optimum cannot necessarily be found in polynomial time. If impingement is found to occur, it would then be necessary to include some of the removed constraints and solve again. This leads to an iterative solution process, but this is often still faster than solving the complete, global problem.

2.   Time-Step Grouping

For problems where the avoidance regions are not small compared to the maneuvering space, there is likely to be extensive interaction between vehicles, and it is not likely that the trajectory will be of the bang–off–bang type. However, as vehicles and obstacles move past each other, the interactions typically last for several time steps at least, due to the comparatively large avoidance regions. Consequently, the binary variable settings for those regions will typically be identical over a long sequence of steps. For example, if a vehicle is on one side of an avoidance region at a certain time step, it is likely to be on the same side at adjacent time steps.

This prediction can be used by sharing binary variables across small groups of adjacent time steps. This essentially equates the binary variables across groups. Note that these extra constraints add conservatism to the problem. However, the prediction of equal binary settings implies that the solution to the original problem is likely to satisfy these new constraints, and so the expected fuel penalty is small.

In physical terms, the grouping represents the requirement that all members of the group be on the same side of an avoidance region, or satisfy the same one of the original *or* constraints. Consider an example with plume impingement. A vehicle might move within impinging range of an obstacle on the second of three time adjacent steps. In the full problem, the vehicle can fire on the first step without impinging on the obstacle, but not on the second and third steps. When these three steps are grouped together, the formulation prevents firing on all three steps. To compensate, the solution would involve firing in the group before or after, if necessary. However, because the time-steps are close together compared to the size of the obstacle, shifting the firing by one or two steps should not significantly impact the fuel cost. Time step grouping does, however, significantly reduce the problem size, as shown in the examples.

C.   Implementation

The global optimization problem is solved by a mixed-integer program solver implemented in the CPLEX software package.[14] The AMPL language[38] is used as the interface to CPLEX. Implementing the constraints in AMPL is straightforward, requiring minimal translation from the form shown in this paper. The problem formulation and constraints are defined in a model file, whereas the parameter values are in a separate data file. As a result, changes to the problem can be made without rebuilding the constraint expressions. AMPL combines the model and data files into a suitable format before invoking CPLEX to solve the problem. A combination of MATLAB® and AMPL scripts enables the path-planning problem to be initiated by a single command and then conveniently combined with simulation and plotting utilities.

IV.   Application Examples

This section presents four examples to demonstrate all of the constraint formulations from the preceding section. The results include the designed trajectories for each maneuver and the computation times required. The models used for the simulations were the same as those in the trajectory design: The problem of compensating for unmodeled effects will be addressed in future work. In all cases, the MILP problems were solved on a 1-GHz personal computer with 256 MB of RAM. Table 1 summarizes the features included in each example. The first two examples are simple cases to demonstrate that the constraints work as intended. The second pair of examples involves larger and more realistic problems, illustrating the application of the approximation techniques to reduce solution times.

A.   Multiple Vehicles in Two Dimensions

These examples involve a very simple maneuver that clearly illustrates the effects of adding collision avoidance and then plume

Table 1 Summary of examples

Example	Collision avoidance		Plume avoidance		Configuration selection	Dynamics
	Obstacles	Vehicles	Obstacles	Vehicles		
Multiple vehicles in two dimensions	——	X[a]	——	X	——	Free space
Multiple satellites	——	X	——	X	——	Hill's equations
ISS remote camera	X	——	X	——	——	Hill's equations
Formation reconfiguration	——	X	——	X	X	Hill's equations

[a]X indicates that those constraints were included in that example.

Table 2 Start and end positions for two-dimensional example

Vehicle	Start position	End position
1	(−2,0)	(8,0)
2	(0,0)	(6,0)
3	(2,0)	(4,0)
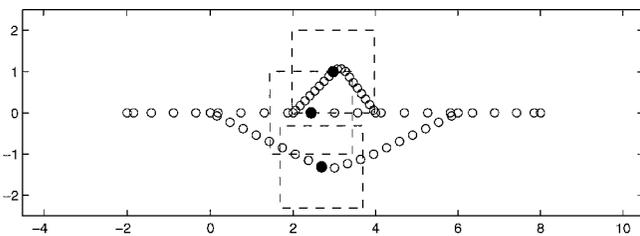
a) Collision avoidance
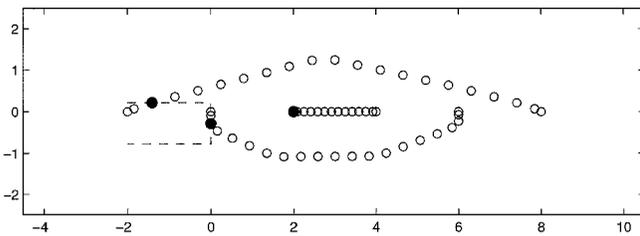
b) Collision and plume avoidance

Fig. 2 Maneuvers in two dimensions.

a)   b)

Fig. 3 Reconfiguration maneuver a) without plume impingement constraints and b) with plume impingement constraints: trajectory of $\bigtriangledown$, vehicle 1; $\square$, vehicle 2; and $\bigcirc$, vehicle 3; lines indicate the direction of the firing plumes, and their lengths are proportional to the thrust demanded.

impingement constraints. Three vehicles, each modeled as a point mass moving in free space with thruster actuation in two dimensions, are required to move between the positions given in Table 2. The start and end positions are all on the X axis, but the order of the vehicles is reversed. Therefore, the LP solution of straight-line trajectories would clearly lead to a collision between all three vehicles at the halfway point.

Figure 2a shows the result of the MILP optimization that includes the collision avoidance constraints from Eq. (15) with a safety distance of one unit. Vehicles 2 (lower) and 3 (upper) move in the Y direction off of the X axis to allow vehicle 1 to follow a straight-line path. The heavy dots indicate the positions of the vehicles at the 10th time step, with the corresponding exclusion boxes shown dashed. Vehicles 1 and 3 are separated by exactly the safety distance in the Y direction. At earlier time steps, vehicles 1 and 2 also move along the edges of the exclusion regions. This shows that collision avoidance is efficiently implemented in this MILP formulation.

In Fig. 2a, vehicle 2 leaves its starting position by moving down and to the right. The velocity component in the X direction indicates that its positive X thruster must have fired at the start of the maneuver, which would impinge on vehicle 1 if the plume extended that far. Vehicle 3 would also have impinged on both vehicles to its left. Given these observations, adding the plume impingement constraints should have a significant effect on the designed trajectories.

Figure 2b shows the results from the optimization with both collision avoidance and plume impingement, using the constraints from Eqs. (18) and (19). The plume region was chosen to be 10 units long by 1 unit across. With the new constraints, it is now favorable for
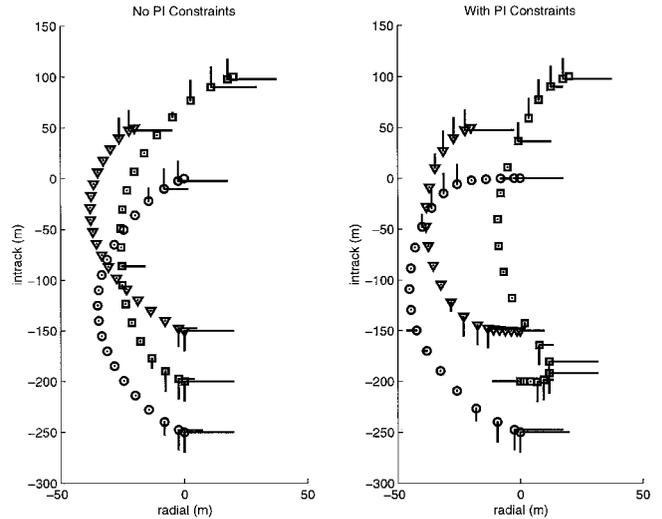
vehicle 3 to remain on the X axis while the other two vehicles move around it. The trajectory of vehicle 2 best demonstrates the plume constraints: Because it starts with a vehicle on each side, it initially cannot fire in the X direction. It can be seen moving away from the axis in the Y direction only. At the third time step, marked by the heavy dots, vehicle 2 fires in the X direction. Its plume region is shown dashed, and it just avoids impinging on vehicle 1. Again, this illustrates the efficiency of the MILP constraints in enforcing avoidance. Note that vehicle 3 is still at its starting position at this time: It cannot fire its thrusters until the other two vehicles are far enough away from the X-axis. This occurs at the fifth time step, at which time vehicle 3 fires to start its maneuver.

Even in this simple example, the optimal solution to the avoidance problems is not obvious, but it can be found very easily using MILP. Also, it can be seen that the designed trajectories use the avoidance margins exactly as specified.

B. Multiple-Satellite Plume Impingement Avoidance

This example demonstrates the use of the constraints from Eqs. (18) and (19) to prevent spacecraft jet plumes impinging on other spacecraft. Three identical spacecraft of mass 30 kg are separated along a line in-track. They are required to reconfigure onto one of the passive apertures, which in this case is a triangle in the plane of the orbit. The maneuver must be done in 9 min, which is $\frac{1}{10}$th of a 90-min orbit. This artificially short timescale makes in-track firing a favorable option where available, and so the effect of the plume impingement constraints is clearly demonstrated.

Each spacecraft has thrusters providing up to 0.2 N in each direction. The plume avoidance regions for the thrusters are 50 m wide and extend 120 m from the spacecraft. Collision avoidance is also enforced with a safety distance of 10 m. The problem was discretized into 20 time steps.

Figure 3a shows the trajectory designed without plume constraints. Note that the designed trajectories remain in the orbital

plane, even though the full three-dimensional model from Eq. (4) was used in the problem. As expected, substantial in-track firing is used to complete the problem in the time available. This leads to considerable plume impingement at the beginning and end of the maneuvers. Figure 3b shows the trajectories redesigned to prevent plume impingement. Considerable deviations from the preceding case are evident. For example, vehicle 1 ($\bigtriangledown$) moves slowly at first, to allow the other vehicles to escape its plume region, before firing in-track to move toward its target. Vehicle 2 ($\square$) actually moves in the opposite direction to avoid the plume from vehicle 1 before turning toward its destination. Vehicle 3 ($\bigcirc$) is the only one free to fire in-track from the beginning. It moves quickly away, but has to approach its target point in the radial direction so that its braking thrust does not impinge on the other vehicles. The fuel required for the maneuver avoiding impingement is equivalent to a total ΔV of 7.097 m/s, compared with 5.033 m/s for the design that does not consider impingement.

C. International Space Station Remote Camera

This problem involves a microsatellite being used for external inspection of the International Space Station (ISS). The satellite is required to move between specified start and endpoints on opposite sides of the station without colliding with the structure or firing its thrusters at the station. The collision avoidance part of this problem was addressed in Ref. 20 using potential functions.

The dynamics are the Hill's equations for a 90-min orbit, as in the preceding example. The camera satellite is modeled as a mass of 5 kg with thrusters giving up to 1 mN in each direction. The ISS is modeled as a collection of boxes as shown in Fig. 4. The maneuver lasts for 4000 s and is discretized into 40 time steps. Plume impingement on the ISS is prevented using the constraints in Eqs. (21) and (22), using a plume length $P = 10$ m and half-width $W = 1$ m.

This problem was first solved with collision avoidance constraints, but ignoring plume impingement. The designed trajectory is shown in Fig. 4. The total fuel use is equivalent to a ΔV of 0.236 m/s. However, note that, during the final few steps, the braking thrust impinges on the station. Figure 5 shows the trajectory redesigned to prevent plume impingement, but with the same start and finish points and duration. The final stages of this maneuver are shown in Fig. 6, seen in a larger scale from a different angle. Because the final position is in a corner formed by two adjacent modules, the camera satellite must approach from the side to prevent its braking thrust from impinging on the station. Figure 6 shows the satellite making the necessary adjustment to its course by firing while still clear of the station, and its final approach leaves it requiring a braking thrust in the only available direction. The total fuel use for this maneuver is equivalent to a ΔV of 0.269 m/s. The additional firing is needed to achieve the final approach direction.

Of the examples presented here, this is the most computationally demanding. Although there is only one vehicle, collision avoidance

Table 3   Results for ISS problem with plume constraints

Time-step grouping size	Computation time, s	Fuel cost as ΔV
None	1800	0.2692
2	190	0.2727
3	54	0.2746
4	67	0.2864
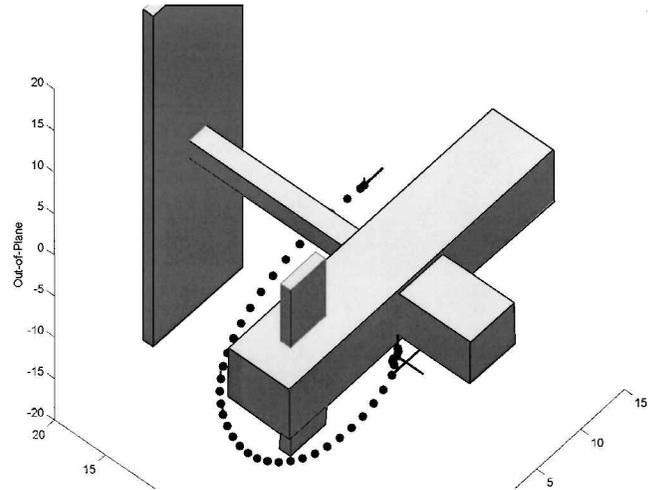
Fig. 5   ISS remote camera maneuver with plume impingement constraints added; start and end positions are the same as in Fig. 4.
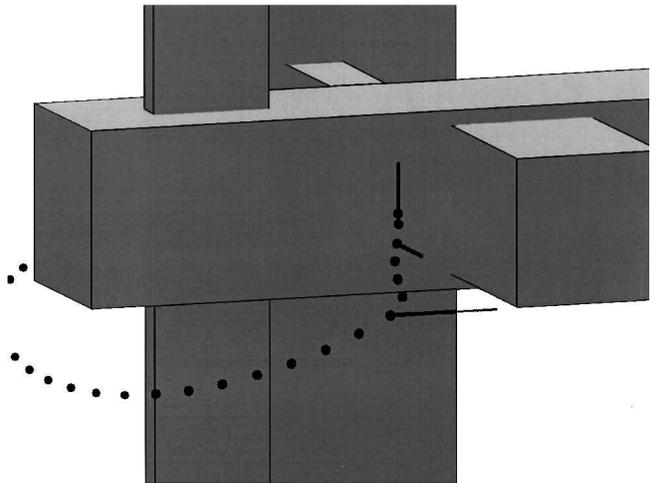
Fig. 6   Final stages of maneuver from Fig. 5, shown in close-up from below.

in three dimensions considering five obstacles requires some 30 binary variables per time step. Because some of the obstacles are thin panels, it is necessary to use a short time step to ensure that the vehicle cannot jump straight through an obstacle between time steps. Therefore at least 40 time steps are needed. To solve for collision avoidance alone involves roughly 1200 binary variables, but this problem can be solved in approximately 8.0 s. Adding plume impingement requires a further 210 binary variable per time step, forming a problem with 9600 binary variables. Because the binary variable search space is so large, it is impractical to compute the global optimal solution to the full problem. The result shown in Fig. 5 is the best feasible solution found in half of an hour of computation, returned using the solution time limit facility in CPLEX.

This problem is well suited to the time-step grouping technique described in the computation section, because the obstacles are large compared to the distance traveled in a time step. Table 3 compares the results from using groups of different sizes. The top row shows the results for the original problem without grouping. It can be seen

Fig. 4   ISS remote camera maneuver with collision avoidance but without plume impingement constraints; all dimensions in meters.

that using groups of three time steps reduces the computation time by a factor of at least 30 at the expense of only a 2% increase in fuel use. This additional conservatism was expected due to the more constrained nature of the problem when grouping is used. Note that increasing the group size beyond three actually causes a slight increase in the solution time.

This problem is particularly sensitive to the numerical conditioning of the MILP. With groups of three time steps, but without the normalization shown in Eqs. (26) and (27), the same problem took 587 s to solve. A further experiment was done to find the variation of the solution time with the particular problem instance. Problems were solved with 52 different, randomly generated start and end positions around the station, using time-step groups of length three. All of these gave feasible solutions within the maneuver time. The mean solution time was 65.0 s, with individual times ranging from 0.28 to 386 s. These results suggest that in general, trajectory optimization problems of this nature can be solved in reasonable computation times.

D. Satellite Formation Reconfiguration

In this example, a group of satellites is reconfigured to be evenly spaced around an ellipse of a given size and inclination. In an interferometry application, this would correspond to a change in aperture, as discussed earlier. The satellites are identical, so that the assignment within the formation and the overall rotation of the ellipse can be selected in the optimization using the terminal constraints from Eqs. (24) and (25). This example will also be used to demonstrate the computation times involved with various numbers of vehicles and different constraint forms. Results are shown for cases involving two, three, and four vehicles, with and without plume impingement.

The vehicles are initially arranged along a line in-track. Each satellite is modeled as a point mass of 50 kg, with Hill's equations as the relative dynamics using a 90-min orbit. The final configuration selection is discretized such that the ellipse is divided into 30 possible global configurations, each containing one position for each spacecraft. The maneuver must be performed in 1000 s, equivalent to just over 15 min or one-sixth of an orbit. The maneuver is discretized into 25 time steps, each of 40 s. The plume exclusion box is 100 m long and is 20×20 m square in cross section ($P = 100$ and $W = 10$).

Figures 7 and 8 show the designed trajectories for the three- and four-vehicle cases considering only collision avoidance. The ellipse
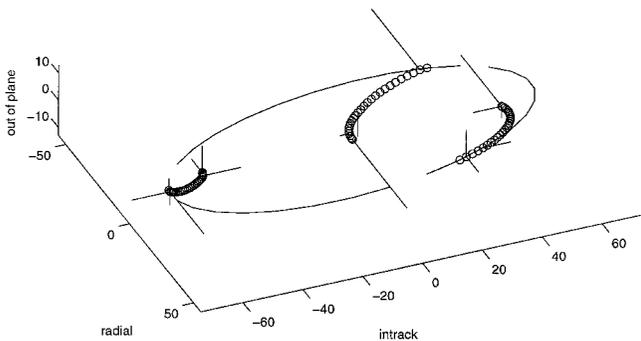
Fig. 7   Reconfiguration maneuver for three spacecraft.
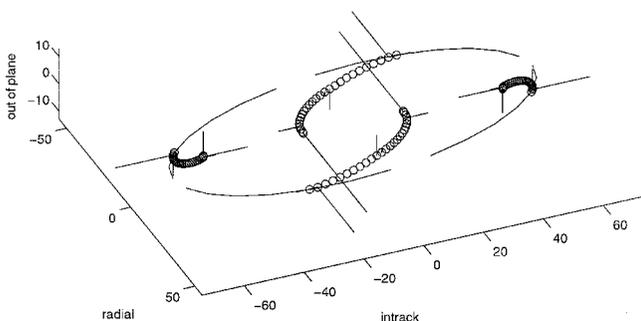
Fig. 8   Reconfiguration maneuver for four spacecraft.

Table 4   Results for reconfiguration maneuver, showing computation time (in seconds) and fuel (as $\triangle V$ in meters per second) and V = number of vehicles

Case	$V = 2$		$V = 3$		$V = 4$	
	Time	Fuel	Time	Fuel	Time	Fuel
Collision avoidance only	1.0	4.67	7.8	9.82	28	13.18
With PI[a]	6.8	4.67	170	10.20	1900	13.84
PI on first two steps	1.1	4.67	7.3	9.99	36	13.24
PI on first ten steps	2.0	4.67	33	10.20	640	13.84
PI on all steps, groups of three	4.8	4.67	100	10.35	2500	14.41

[a]Plume impingement.

associated with the final aperture is also shown. The first row of results in Table 4 shows the computation time and fuel use in each case. Adding plume impingement restrictions on all steps causes a considerable increase in computation time, as shown in the second row of Table 4. Although the computation times for two- and three-vehicle cases are still shorter than the maneuver time, the four-vehicle case now takes over half of an hour to compute, which is approximately twice as long as the maneuver itself. As in the ISS example, this complexity arises from the number of binary variables in the problem. For example, the four-vehicle case involves 900 binaries for collision avoidance, 480 for configuration selection, and 9000 for plume avoidance.

This issue with the computation time can be addressed by recognizing that this problem is likely to result in a bang-off-bang trajectory because the vehicle avoidance regions are small compared to the maneuvering distances. Therefore, the removal of plume constraints in the coast phase, as discussed in the computation section, can be used to reduce solution time. Also, the final positions are much further apart and not aligned with each other, and so impingement is not expected to occur at the end of the maneuver. The third row of results in Table 4 was obtained by preventing plume impingement only on the first two time steps. However, a postanalysis of the trajectories showed that plume impingement occurred where the constraints had been removed. Thus, the fourth row shows results with constraints preventing plume impingement on the first 10 steps. This time, the postanalysis showed that no impingement occurred, which can also be deduced from the matching fuel use values in the second and fourth rows. The significant reduction in solution time with the removal of redundant constraints demonstrates that the iterative approach to constraint removal is often faster than solving the completely constrained problem.

The fifth row shows the results of experimental application of the time-step grouping idea to this problem. The grouping technique is not expected to be very effective here because the avoidance regions are small compared to the maneuvering space, and the interaction between vehicles is short lived. In this case, plume impingement was prevented on all steps, but the binary variables were shared across groups of three adjacent time steps. The results are slightly more conservative than those in the second row, as expected from the grouping method. The computation times are slightly reduced for the two- and three-vehicle cases, but increased in the four-vehicle case. This contrasts with the ISS example, where grouping made the most complicated case solve more quickly. This demonstrates the difference between the two approximation techniques and shows that they are dependent on the problem characteristics. More generally, the inclusion of prior knowledge for simplification is problem specific.

V.   Conclusion

This paper describes and demonstrates the use of MILP to design spacecraft trajectories that avoid collisions and plume impingements. Logical constraints to enforce avoidance are appended to the basic fuel optimization to suit the application. Other logical constraints can be used to enforce a final relative alignment of identical spacecraft while allowing the assignment of the spacecraft within the formation to be chosen as part of the optimization. Examples are presented across a range of applications, involving various combinations of the avoidance constraints. Two important simplifications are presented to enable the solution of very difficult trajectory optimization problems. The first is the conversion to a linear form,

and the second is the use of prior knowledge of the problem to remove redundant constraints and apply efficient approximations. With these approaches, solutions have been obtained in practical computation times, even in cases involving very large numbers of binary variables.

Acknowledgments

References

[1]Bauer, F. H., Bristow, J., Folta, D., Hartman, K., Quinn, D., and How, J. P., "Satellite Formation Flying Using an Innovative Autonomous Control System (AUTOCON) Environment," AIAA Paper 97-3821, 1997.

[2]Bauer, F. H., Hartman, K., and Lightsey, E. G., "Spaceborne GPS: Current Status and Future Visions," ION-GPS Conference, Inst. of Navigation, Alexandria, VA, 1998, pp. 1493–1508.

[3]Bauer, F. H., Hartman, K., How, J. P., Bristow, J., Weidow, D., and Busse, F., "Enabling Spacecraft Formation Flying Through Spaceborne GPS and Enhanced Automation Technologies," ION-GPS Conference, Inst. of Navigation, Alexandria, VA, 1999, pp. 369–384,.

[4]Das, A., and Cobb, R., "TechSat21: Space Missions Using Collaborating Constellations of Satellites," AIAA Paper 98-5255, 1998.

[5]Robertson, A., Inalhan, G., and How, J. P., "Spacecraft Formation Flying Control Design for the Orion Mission," Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA, Reston, VA, 1999, pp. 1562–1575.

[6]Inalhan, G., Busse, F. D., and How, J. P., "Precise Formation Flying Control of Multiple Spacecraft Using Carrier-Phase Differential GPS," AAS/AIAA Spaceflight Mechanics Meeting, AIAA, Reston, VA, 2000, pp. 151–165.

[7]Tillerson, M., Inalhan, G., and How, J., "Coordination and Control of Distributed Spacecraft Systems Using Convex Optimization Techniques," International Journal of Robust and Nonlinear Control, Vol. 12, No. 2, 2002, pp. 207–242.

[8]Tillerson, M., and How, J., "Formation Flying Control in Eccentric Orbits," AIAA Paper 2001-4092, 2001.

[9]Reif, J. H., "Complexity of the Mover's Problem and Generalizations," 20th IEEE Symposium on the Foundations of Computer Science, Inst. of Electrical and Electronics Engineers, New York, 1979, pp. 421–427.

[10]Williams, H. P., and Brailsford, S. C., "Computational Logic and Integer Programming," Advances in Linear and Integer Programming, edited by J. E. Beasley, Clarendon, Oxford, 1996, pp. 249–281.

[11]Bemporad, A., and Morari, M., "Control of Systems Integrating Logic, Dynamics, and Constraints," Automatica, Vol. 35, No. 3, 1999, pp. 407–427.

[12]Floudas, C. A., Nonlinear and Mixed-Integer Programming—Fundamentals and Applications, Oxford Univ. Press, New York, 1995 pp. 95–107.

[13]Subramanian, R., Scheff, R., Jr., Quillinan, J., Wiper, D., and Marsten, R., "Coldstart: Fleet Assignment at Delta Air Lines," Interfaces, Vol. 24, No. 1, 1994, pp. 104–20.

[14]ILOG AMPL CPLEX System Version 7.0 User's Guide, ILOG, Inc., Incline Village, NV, 2000, pp. 17–53,

[15]Kaplan, M. H., Modern Spacecraft Dynamics and Control, Wiley, New York, 1976, pp. 108–115.

[16]Sedwick, R., Miller, D., and Kong, E., "Mitigation of Differential Perturbations in Synthetic Apertures Comprised of Formation Flying Satellites," Advances in Astronautical Sciences: Space Flight Mechanics, Vol. 102, Univelt, San Diego, CA, 1999, pp. 323–342.

[17]Barraquand, J., Kavraki, L., Latombe, J.-C., Motwani, R., Li, T.-Y., and Raghavan, P., "A Random Sampling Scheme for Path Planning," International Journal of Robotics Research, Vol. 16, No. 6, 1997, pp. 759–774.

[18]Krozel, J., and Peters, M., "Strategic Conflict Detection and Resolution for Free Flight," 36th Conference on Decision and Control, Inst. of Electrical and Electronics Engineers, New York, 1997, pp. 1822–1828.

[19]Mao, Z.-H., and Feron, E., "Stability of Intersecting Aircraft Flows Under Decentralized Conflict Avoidance Rules," Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA, Reston, VA, 2000, pp. 1042–1052.

[20]Roger, A. B., and McInnes, C. R., "Safety Constrained Free-Flyer Path Planning at the International Space Station," Journal of Guidance, Control, and Dynamics, Vol. 23, No. 6, 2000, pp. 971–979.

[21]Singh, G., and Hadaegh, F. Y., "Collision Avoidance Guidance for Formation Flying Applications," AIAA Paper 2001-4088, 2001.

[22]Milam, M. B., Mushambi, K., and Murray, R. M., "A New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems," 39th IEEE Conference on Decision and Control, Inst. of Electrical and Electronics Engineers, New York, 2000, pp. 845–851.

[23]Spehar, P. T., and Le, T. Q., "Automating an Orbiter Approach to Space Station Freedom to Minimize Plume Impingement," NASA Automated Rendezvous and Capture Review, NASA N93-22305, 1993.

[24]How, J., Twiggs, R., Weidow, D., Hartman, K., and Bauer, F., "Orion: A Low-Cost Demonstration of Formation Flying in Space Using GPS," Proceedings of the AIAA Astrodynamics Specialists Conference, AIAA, Reston, VA, 1998, pp. 276–286.

[25]Ferguson, P., Busse, F., Engberg, B., How, J., Tillerson, M., Pohlman, N., Richards, A., and Twiggs, R., "Formation Flying Experiments on the Orion-Emerald Mission," AIAA Paper 2001-4688, 2001.

[26]Beichman, C. A., "The Terrestrial Planet Finder: The Search for Life-bearing Planets Around Other Stars," SPIE Conference on Astronomical Interferometry, Vol. 3350, Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, 1998, pp. 719–723.

[27]Yeh, H.-H., and Sparks, A., "Geometry and Control of Satellite Formations," Proceedings of the American Control Conference, Vol. 1, Inst. of Electrical and Electronics Engineers, New York, 2000, pp. 384-388.

[28]Inalhan, G., Tillerson, M., and How, J. P., "Relative Dynamics and Control of Spacecraft Formations in Eccentric Orbits," Journal of Guidance, Control, and Dynamics, Vol. 25, No. 1, 2002, pp. 48–59.

[29]Alfriend, K. T., Schaub, H., and Gim, D.-W., "Gravitational Perturbations, Nonlinearity and Circular Orbit Assumption Effects on Formation Flying Control Strategies," Advances in the Astronautical Sciences, Guidance and Control, Vol. 104, Univelt, San Diego, CA, 2000, pp. 139–158.

[30]Wang, P. K. C., and Hadaegh, F. Y., "Optimal Formation-Reconfiguration for Multiple Spacecraft," Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA, Reston, VA, 1998, pp. 686–696.

[31]Burkard, R. E., and Çela, E., "Linear Assignment Problems and Extensions," Handbook of Combinatorial Optimization, edited by Z. Du and P. Pardalos, Kluwer Academic Dordrecht, The Netherlands, 1999, pp. 75–149.

[32]Tillerson, M., and How, J., "Advanced Guidance Algorithms for Spacecraft Formation Flying," 2002 American Control Conference, Inst. of Electrical and Electronics Engineers, New York (to be published).

[33]Bertsimas, D., and Tsitsiklis, J. N., Introduction to Linear Optimization, Athena Scientific, Belmont, MA, 1997, pp. 17–19.

[34]Schouwenaars, T., DeMoor, B., Feron, E., and How, J., "Mixed Integer Programming for Multi-Vehicle Path Planning," European Control Conference, European Union Control Association, Porto, Portugal, 2001, pp. 2603–2608.

[35]Schouwenaars, T., "Mixed Integer Programming for Optimal Collision-Free Path Planning of Autonomous Vehicles," M.S. Thesis, Esat-Sista, Katholieke Univ. Leuven, Leuven, Belgium, May 2001.

[36]Richards, A., How, J., Schouwenaars, T., and Feron, E., "Plume Avoidance Maneuver Planning Using Mixed Integer Linear Programming," AIAA Paper 2001-4091, 2001.

[37]Kirk, D. E., Optimal Control Theory: An Introduction, Prentice–Hall, Upper Saddle River, NJ, 1970, pp. 260–262.

[38]Fourer, R., Gay, D. M., and Kernighar, B. W., AMPL, A Modeling Language for Mathematical Programming, Boyd and Fraser, Danvers, MA, 1993, pp. 291–306 (originally published by Scientific Press).