Avoidance Maneuver Planning Incorporating Station-Keeping Constraints and Automatic Relaxation

Joseph B. Mueller*, Paul R. Griesemer†
Stephanie Thomas‡

Princeton Satellite Systems, Plainsboro, NJ, 08536

Space debris is a rising concern for the sustained operation of our satellites. The population in space is continually growing, both gradually with a steady stream of new launches, and in sudden bursts, as evidenced with the recent collision between the Iridium and inactive COSMOS spacecraft. The problem is most severe in densely populated orbit regimes, where satellites face a sustained presence of close-orbiting objects. In general, the frequent occurrence of potential collisions with debris will have a negative impact on mission performance in two important ways. Firstly, repeated avoidance maneuvers diminish fuel and thus reduce mission life. Secondly, excursions from the nominal orbit during avoidance maneuvers may violate mission requirements or payload constraints. It is therefore important to consider both fuel minimization and station-keeping objectives in the avoidance planning problem.

In this paper, we formulate the avoidance maneuver planning problem as a linear program (LP). Avoidance constraints and orbit station-keeping constraints are expressed as linear functions of the control input. The relative orbit dynamics are modeled as a discrete, linear time-varying system that models both circular and eccentric orbits. The original non-linear, non-convex avoidance constraints are transformed into a time-varying sequence of linear constraints, and the navigation uncertainty is applied in a worst-case sense. Finally, the minimum-fuel avoidance maneuver problem is formulated with station-keeping constraints in a way that enables automatic relaxation of certain constraints to ensure feasibility.

I. Introduction

Space debris is a rising concern for the sustained operation of our satellites. The population in space is continually growing, both gradually with a steady stream of new launches, and in sudden bursts, as evidenced with the recent collision between the Iridium and inactive COSMOS spacecraft. The problem is most severe in densely populated orbit regimes, where satellites face a sustained presence of close-orbiting objects. This type of scenario occurs today in sun-synchronous orbits where hundreds of spacecraft share similar altitudes and inclinations. It promises to occur more frequently in the future, as more missions begin to adopt a distributed paradigm, with multiple close-orbiting spacecraft flying in controlled formations.

Several new mission concepts are being developed that utilize a distributed set of spacecraft orbiting in close-proximity. Some of the actively funded programs include the TPF-I mission to search for Earth-like planets, the MMS mission to measure Earth’s magnetosphere, and the DARPA F6 program to replace a single satellite with multiple free-flying components. As the technologies that enable formation flying continue to develop, more and more missions will undoubtedly seek to capitalize on the benefits brought by this distributed paradigm. A key challenge that must be addressed in any formation flying mission is the critical need for collision avoidance. This becomes a particularly important and complex task for missions like F6 that involve large numbers of satellites, and where each one is meant to serve as a module that can

*Senior Technical Staff, AIAA Senior Member, jmueller at psatellite.com
†Senior Technical Staff, AIAA Senior Member
‡Chief Engineer, AIAA Member

AIAA Infotech@Aerospace 2010
20 - 22 April 2010, Atlanta, Georgia

AIAA 2010-3525

Copyright © 2010 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.
be replaced. The benefits of this kind of decentralized design include greater redundancy, the potential for lower launch costs and, the ability to extend mission life by replacing individual components. An immediate consequence of this paradigm, however, is that each module in the system effectively becomes another piece of debris that all other maneuverable satellites must avoid.

The relative motion between spacecraft can be divided into two categories: 1) high relative velocity, such as the intersection of two non-coplanar orbits, and 2) low relative velocity, where the orbits are essentially coplanar with small differences in the orbital elements. High relative velocity objects are in close proximity only over short time scales (on the order of seconds), and therefore an avoidance maneuver would have to be implemented prior to the time of intersection. Fly-by’s with low relative velocity objects are happening more and more frequently, due to the crowding of popular orbit regimes, especially sun-synchronous and geo-stationary orbits. In these scenarios, the time scales of close-proximity motion are much longer, on the order of hours or days. It is therefore physically possible in these cases for a spacecraft to detect neighboring objects and to enact avoidance maneuvers to ensure safe separation distances.

Spacecraft with the capability to detect potential collisions, then plan and enact avoidance maneuvers can successfully mitigate the risk. In general, the frequent occurrence of potential collisions will have a negative impact on mission performance in two important ways. Firstly, repeated avoidance maneuvers diminish fuel and thus reduce mission life. Secondly, excursions from the nominal orbit during avoidance maneuvers may violate mission requirements or payload constraints. It is therefore important to consider both fuel minimization and station-keeping objectives in the avoidance planning problem.

The topics of optimal maneuver planning and collision avoidance for close-orbiting spacecraft have been studied by numerous researchers in recent years [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. Previous papers by Princeton Satellite Systems [16, 17, 18] have discussed the architectural considerations for large scale systems of formation flying satellites. Included in these papers are discussions of relative orbit dynamics in circular and eccentric orbits, the use of convenient parameter sets to describe desired periodic relative motion, and methods for planning reconfiguration maneuvers by formulating the problem as a linear program (LP). In 2008, a proposed LP-based method for robust avoidance maneuver planning was presented [19], and in 2009 the approach was extended to capture both low and high relative velocity encounters [20].

In this paper, we focus on avoidance planning for low relative velocity encounters, and consider the simultaneous objective of station-keeping during the avoidance maneuver. The maneuver planning problem is formulated as an LP, with avoidance and orbit station-keeping constraints expressed as linear functions of the control input. The relative orbit dynamics are modeled as a discrete, linear time-varying system that models both circular and eccentric orbits. The original non-linear, non-convex avoidance constraints are transformed into a time-varying sequence of linear constraints, and the navigation uncertainty is applied in a worst-case sense. The resulting maneuver can be solved efficiently as an LP with no integer constraints, and can guarantee collision avoidance with respect to bounded navigation uncertainty. Finally, the minimum-fuel avoidance maneuver problem is formulated with station-keeping constraints in a way that enables automatic relaxation of certain constraints to ensure feasibility. This method is compared to an ad hoc approach of constraint relaxation which reduces the overall dimension of the problem, but which requires the LP to be solved multiple times.

The paper is organized as follows. Section II discusses how the relative orbit dynamics are modeled as a linear system. Section III presents the method of generating the LP problem formulation, including station-keeping constraints, separation constraints, and accounting for uncertainty in the initial state. Section IV discusses the basic theory of constraint relaxation and provides a simple 2-dimensional example for illustration. Finally, Section V present simulation results of various avoidance maneuvers performed in conjunction with relaxed station-keeping constraints.
II. Relative Orbit Dynamics

The linear time-varying (LTV) dynamic equations of a satellite’s relative motion in the local-vertical / local-horizon (LVLH) frame are given as follows [4]:

\[
\frac{d}{dt} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = -2 \begin{bmatrix} 0 & 0 & -\nu \\ 0 & 0 & 0 \\ \nu & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} - \begin{bmatrix} -\dot{\nu}^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\dot{\nu}^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} 0 & -\dot{\nu} & 0 \\ -\dot{\nu} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + n^2 \left(1 + e \cos \nu \right) \begin{bmatrix} -x \\ -y \\ 2z \end{bmatrix} + \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}
\]

(1)

where \( \nu \) is the true anomaly, \( e \) is the eccentricity, \( n \) is the mean orbit rate, and \( a \) is the applied acceleration. The orthogonal axes are defined as \( z \) nadir, \( y \) cross-track (normal to the orbital plane) and \( x \) completing the right-handed system. Let the state vector of relative position and velocity be \( x = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \), and let the control vector of accelerations be \( u = [a_x, a_y, a_z]^T \). The system system is then written in state-space form as:

\[
x(t) = A(t)x(t) + B(t)u(t)
\]

(2)

where \( A(t) \in \mathbb{R}^{6 \times 6} \) and \( B(t) \in \mathbb{R}^{6 \times 3} \). In the case of zero eccentricity, the \( \dot{\theta} \) terms reduce to the mean orbit rate \( n \), and the \( \ddot{\theta} \) terms vanish. The dynamics would then simplify to an undamped, linear time-invariant (LTI) system, with a natural frequency equal to the mean orbit rate, and an unforced response to an initial state would be given by the familiar Clohessy-Wiltshire or Hill’s equations. In addition, for circular orbits, the \( x \) axis is always aligned with the velocity vector.

For the LTI system (circular orbits), the \( A \) matrix is constant, whereas for the more general LTV system, it varies with the true anomaly of the orbit. The continuous time system can be discretized with a zero-order hold at an appropriate sampling period \( \Delta t_k \) over \( N \) samples to become:

\[
x_{k+1} = A_k x_k + B_k u_k
\]

(3)

where \( k \in [0,1,\ldots,N] \) indexes time, \( A_k \) varies with \( k \) for non-circular orbits, and both matrices vary with \( \Delta t_k \). Applying the equation recursively from \( k = 0 \) to \( k = N - 1 \), we have the following expression for the final state:

\[
x_N = (A_{N-1} \cdots A_1 A_0) x_0 + \left[A_{N-1} \cdots A_1 B_0, ~ A_{N-1} \cdots A_2 B_1, \ldots, ~ A_{N-1} B_{N-2}, ~ B_{N-1}\right]
\]

(4)

More generally, the state at time \( t(j) \), \( j \in (1,N) \), is expressed as:

\[
x_j = (A_{j-1} \cdots A_1 A_0) x_0 + \left[A_{j-1} \cdots A_1 B_0, ~ A_{j-1} \cdots A_2 B_1, \ldots, ~ A_{j-1} B_{j-2}, ~ B_{j-1}, ~ 0\right]
\]

(5)

Let us now write this more compactly as:

\[
x_j = H_j x_0 + G_j \tilde{u}
\]

(6)
where \( \tilde{u} \in \mathbb{R}^{3N \times 1} \) is the stacked vector of controls, and \( H_j \in \mathbb{R}^{6 \times 6} \) and \( G_j \in \mathbb{R}^{6 \times 3N} \) are given from Eq. 5. This formulation allows us to express the relative position and velocity at any future time as a linear function of the original state \( x_0 \) and the control history \( \tilde{u} \).

For the sake of clear notation in the sections that follow, let the components of the relative state at time \( t_j \) be denoted as:

\[
\begin{align*}
x_j &= x_j(1) & \dot{x}_j &= x_j(4) \\
y_j &= x_j(2) & \dot{y}_j &= x_j(5) \\
z_j &= x_j(3) & \dot{z}_j &= x_j(6)
\end{align*}
\]

### III. General Problem Formulation

The decision variables in the optimization are the applied accelerations, \( \tilde{u} \). The optimization problem may be stated as follows: Given an initial relative state of some object, determine a control history that minimizes the required fuel subject to a) maintaining a minimum separation distance \( d \) from the object, and b) meeting a prescribed set of station-keeping constraints. In general, the compact formulation is given as:

**General Problem**

Minimize \( \sum_{j=1}^{N} c_j \left( |u_j(1)| + |u_j(2)| + |u_j(3)| \right) \)  
Subject to  
\[
\begin{align*}
a_i^T \tilde{u} - b_i &\leq 0 & i = 1, \ldots, M \\
0 &\leq \tilde{u} \leq u_{\text{max}}
\end{align*}
\]

where \( a_i, b_i \) represent the linear data associated with the \( i^{th} \) constraint. The 3 elements of \( \tilde{u}_j \) represent the \( x, y, z \) components of the acceleration vector at time \( t(j) \). The control in each axis is bounded from above at all times by the maximum acceleration, \( u_{\text{max}} \). The total cost to minimize is equivalent to the sum of the absolute accelerations applied in each axis. Using time-dependent weights on the controls, \( c_j \), the use of applied accelerations may be penalized more severely at certain times. This technique may be useful when scheduling the use of thrusters so that they coordinate properly with other systems on the spacecraft.

In general, the \( i^{th} \) constraint may be expressed as:

\[
q_i^T \left( \tilde{G} \tilde{u} + H x_0 \right) \leq p_i
\]

where \( \tilde{x} = \tilde{G} \tilde{u} + H x_0 \) is the full relative state vector stacked across all times. The vector \( q_i \) defines a linear combination of these states. Using this notation, the formulas for \( a_i \) and \( b_i \) may be written as follows:

\[
\begin{align*}
a_i^T &= q_i^T \tilde{G} \\
b_i &= p_i - q_i^T H x_0
\end{align*}
\]

#### 1. Station-Keeping Constraints

Station-keeping constraints are defined with respect to some reference orbital state. The reference may be another physical satellite, or it may be virtual object defined by a desired orbital element set. In any case, we define the initial relative state of the controlled satellite to this reference as \( x_0 \), and the relative state at time \( t(j) \) is \( x_j \).

The station-keeping constraints are defined as a box in the curvilinear co-ordinate frame centered around the reference. At each time \( t(j) \), the following 6 inequality constraints are applied:

\[
\begin{align*}
-x_B &\leq x_j \leq x_B \\
y_B &\leq y_j \leq y_B \\
z_B &\leq z_j \leq z_B \\
\end{align*}
\]

In addition to staying in this box during the maneuver, we wish to achieve zero along-track drift at the end of the maneuver. This is done by forcing the semi-major axis difference to be zero at the final time. In LVLH coordinates, the semi-major axis change is given as:

\[
\Delta a = -4z + 2\dot{x}/n
\]
Thus, the terminal constraint is:

\[-4z_N + 2 \dot{x}_N/n = 0\]  \hspace{1cm} (13)

2. Avoidance Constraints

When considering the potential collision between two satellites, it is important to distinguish between low and high relative velocity encounters. For satellites with high relative velocity, such as the intersection of two non-coplanar orbits, the time scale for close-proximity operations is extremely short. Alternatively, for low relative velocity encounters, where the orbits are essentially coplanar with small differences in the orbital elements, the satellites may stay in close proximity for hours or days. A previous paper discussed robust avoidance maneuver planning for both types of encounters [20]. In this paper, we focus solely on avoidance maneuvers for low relative velocity, long duration encounters.

The approach used for ensuring collision avoidance is to specify a set of constraints over time that force the relative position of the satellite to maintain some minimum separation distance. In defining the relative motion for collision avoidance, we must begin by defining a new initial relative state, \(x_{0,k}\). If we consider the general case of avoiding multiple objects, this is the relative state of the controlled satellite defined with respect to the \(k^{th}\) object to be avoided. It is worth pointing out that this is different than the previously defined initial relative state, \(x_0\).

An example of a valid avoidance constraint is as follows:

\[y_j^2 + z_j^2 \geq R^2\]  \hspace{1cm} (14)

This would ensure the relative position at time \(t(j)\) lies outside of a cylinder of radius \(R\), centered upon the along-track axis. It is practical to ignore the motion in the along-track direction, since this motion is most strongly influenced by differential disturbances and navigation error, and therefore most difficult to control. Also, by defining the separation constraints independent from the along-track motion, the satellites are allowed to drift past each other while maintaining safe separation in the radial / cross-track directions.

The problem with the above constraint, however, is that it is both nonlinear and non-convex. In general this can be solved as a nonlinear programming problem (NLP), but it is difficult to solve over a non-convex space. One way to model this in the context of an optimization problem would be to relax the non-convex constraint using Lagrangian relaxation, which would lead to a convex semidefinite programming problem [21]. In this case it is possible to achieve a solution which satisfies the initial non-convex constraint, but such a solution is not guaranteed.

An alternative is to approximate the cylinder using a set of linear constraints. Each linear constraint represents a tangent line that touches the cylinder. The separation constraint for each tangent line is:

\[\alpha_k y_j + \beta_k z_j \geq R \quad k = 1, \ldots, N_L, \forall j \in [1, N]\]  \hspace{1cm} (15)

where \(N_L\) is the number of lines used to approximate the circle, and the parameters are defined as \(\alpha_k = \cos \theta_k\) and \(\beta_k = \sin \theta_k\) for a discrete set of angles \(\theta_k\) between 0 and 2\(\pi\). The same set of \(N_L\) constraints are applied at all times \(t(j)\). Note, however, that in order to stay out of the avoidance region, it is only necessary to satisfy one of the \(N_L\) constraints at each time. In other words, it is only necessary (and possible) to be on one side of the circle, rather than all sides. In order to capture this logical criteria, it is necessary to introduce binary variables. This is achieved by modifying the constraint as follows:

\[\alpha_k y_j + \beta_k z_j \geq B_{j,k} R - W \sum_{i \neq k} B_{j,i} \quad k = 1, \ldots, N_L, \forall j \in [1, N]\]  \hspace{1cm} (16)

where \(B_{k} \in [0, 1]\) is a binary variable, and we impose the additional constraint:

\[\sum_{k=1}^{N_L} B_{j,k} = 1\]

Hence only one element of the \(B_j\) vector can take the value of 1, while the rest must be zero. Accordingly, when \(B_{j,k} = 1\), the summation of all other elements in \(B_j\) is zero, and the original constraint of Eq. 15 is realized. Otherwise, when \(B_{j,k} = 0\), the summation term becomes 1, so that the right hand side becomes \(-W\). Choosing \(W \gg 0\) ensures that the constraint is always satisfied. The problem in this case is a mixed-integer linear program, or MILP, because the objectives and constraints are all linear and some of the decision variables are binary integers.
Another interesting way to formulate the separation constraint is to define a single tangent line that rotates about the along-track axis over time, at a distance \( R \) from the axis [19, 20]. The constraint is defined as:

\[
\alpha_j y_j + \beta_j z_j \geq R \quad \forall j \in [1, N]
\]  

where \( \alpha_j = \cos \theta_j \) and \( \beta_j = \sin \theta_j \). The tangent line is made to rotate around the circle one time per orbit period, so that \( \theta_j = \theta_0 + nt_j \). This enables the the time-varying constraints to be consistent with the natural dynamics of the system, so that the constraints can be satisfied with an unforced periodic relative motion. This approach achieves the same desired result of ensuring a minimum separation distance at all times, but because it is linear and requires no binary variables, the problem formulation remains an LP. Note, however, that the parameter \( \theta_0 \) must be selected prior to finalizing the constraints.

The three different methods discussed above are illustrated in Figure 1. The original separation constraint of Eq. 14 is shown on the left as a pure circle. In the middle, the MILP constraints are seen to approximate the circle by forming a discrete set of tangent lines around the perimeter. Recall that only one (and any one) of the tangent line constraints is enforced at each time. On the right, the rotating plane method is shown, with a sequence of three tangent lines. In this case, a single specified tangent line constraint is enforced at each time.

An example of a natural, unforced relative trajectory that satisfies the separation constraints for \( R = 25 \) is shown in Figure 2. The oscillation in the radial (\( z \)) and cross-track (\( y \)) directions are large enough and phased properly so that the motion in the YZ plane surrounds the avoidance region. In this case, the \( y \) and \( z \) motion is aligned to form a circular path, but in general the path in the YZ plane forms an ellipse. In addition, the trajectory is drifting in the along-track direction, resulting in a corkscrew type of motion. If there were zero along-track drift, then the orbital-plane (XZ plane) motion would follow a \( 2 \times 1 \) ellipse.

The approach taken here is to use the rotating tangent plane approach, eliminating the need for binary variables so that the problem remains an LP. The separation constraints are defined as follows:

\[
\cos \theta_j y_j + \sin \theta_j z_j \geq R_j \quad \forall j \in [1, N]
\]  

where once again, \( \theta_j = \theta_0 + nt_j \). The separation distance \( R_j > 0 \) is a parameter that may be varied with time. The primary reason for changing \( R_j \) over time is to allow it to reduce its magnitude during the early part of the maneuver, so that the spacecraft has sufficient time to generate separation distance.

Using this approach, it is necessary to choose an appropriate value for \( \theta_0 \). In fact, because this is a parameter of the optimization problem, the problem may be solved multiple times over a range of values of \( \theta_0 \), so that the best solution may be selected.
In addition to enforcing a separation distance in the YZ plane, we may also impose a minimum drift rate between the satellite and the debris. This will cause the satellite to pass by the object in the along-track direction. The mean along-track drift per orbit is:

\[ D = \left( \frac{2\pi}{n} \right) \times (6zn - 3\dot{x}) \]  

(19)

This leads to one additional constraint for collision avoidance. Let \( D^* \) be the desired minimum drift rate, which may be positive or negative. The corresponding constraint may be expressed as follows:

\[ \frac{D^*}{|D^*|} (6z_jn - 3\dot{x}_j) \geq \frac{|D^*|n}{2\pi} \]  

(20)

Finally, we may impose a physical limit on the total delta-v used over the course of the maneuver, \( \Delta V_{\text{max}} \). This constraint is expressed as follows:

\[ \sum_{j=1}^{N} c_j (|u_j(1)| + |u_j(2)| + |u_j(3)|) \Delta T_j \leq \Delta V_{\text{max}} \]  

(21)

where \( \Delta T_j \) is the time-step used in the discretized LTV state-space model from time \( t_j \) to \( t_{j+1} \). This is equivalent to placing an upper bound on the nominal cost function. Because we are already trying to minimize the cost function, the optimal solution would either be above or below this bound, regardless of whether the constraint is applied. Thus, it is only meaningful to add this constraint when additional terms are present in the cost function. As will be shown later, additional terms are in fact added to the cost function when the station-keeping constraints are relaxed.

3. Robustness to Navigation Uncertainty

Consider the \( i^{\text{th}} \) inequality constraint of the optimization problem:

\[ a_i^T \hat{u} - b_i \leq 0 \]  

(22)

Uncertainty in an initial state \( x_0 \) may be modeled by defining \( x_0 \) as:

\[ x_0 = \hat{x}_0 + C\delta, \quad \delta \in \mathbb{R}^{n_x}, ||\delta|| = 1 \]  

(23)

where \( \hat{x}_0 \) is the estimated initial state, \( n_x \) is the number of states, and \( C \) is the diagonal matrix of the state covariance [22, 23]. The true initial state lies in an ellipsoid defined by this covariance matrix. The objective
for maneuver planning is to satisfy the inequality of Eq. 22 for all possible values of \( x_0 \). Therefore, the new constraint becomes:

\[
a_i^T \tilde{\mathbf{u}} - b_i + f_i \leq 0 \tag{24}
\]

\[
a_i^T \tilde{\mathbf{u}} - q^T H \hat{x}_0 - p_i + ||q^T H^{(i)} C|| \leq 0 \tag{25}
\]

where we define the added term \( f_i \) as:

\[
f_i = ||q^T H^{(i)} C|| \tag{26}
\]

The addition of the \( f_i \) term to each constraint effectively models the worst case perturbation in \( x_0 \) for that constraint. The new constraint remains linear in \( \tilde{\mathbf{u}} \), and so the problem remains an LP. The norm term depends only on known problem data. If uncertainty in the dynamics were considered, this would introduce second order terms in the decision vector, and would become a second order cone programming (SOCP) problem.

The new robust problem formulation is expressed as:

**Robust Problem**

\[
\text{Minimize } \sum_{j=1}^{N} c_j (|u_j(1)| + |u_j(2)| + |u_j(3)|) \tag{27}
\]

Subject to \( a_i^T \tilde{\mathbf{u}} - b_i + f_i \leq 0 \quad i = 1, \ldots, M \)

0 \leq \tilde{\mathbf{u}} \leq u_{\text{max}}

The along-track motion is particularly sensitive to navigation errors in the relative velocity. It is therefore important to choose the along-track drift rate bounds appropriately, given the size of the covariance matrix. For example, with any initial state uncertainty, it is impossible to realize an exact drift rate, so the tolerance must be greater than zero. In general, the size of the tolerance must grow with the size of the covariance matrix. In fact, we can compute the minimum feasible tolerance as:

\[
\epsilon_D = ||[6n, 0, 0, 0, 0, 0, 3] A^N C|| \left( \frac{2\pi}{n} \right) \tag{28}
\]

This relationship is illustrated in the results section.

**IV. Automatic Relaxation in Linear Programming**

1. **Background**

The presence of conflicting constraints on an optimization problem can lead to a situation where the solution space is empty. In order to avoid this failure mode, it is desirable to relax a set of constraints so that a solution is returned despite the fact that it may not satisfy all of the parameters of the problem. For example, we may place one constraint on the final position of the satellite to be greater than some distance from the debris and a second constraint on the maximum acceleration of the thruster. However, the position constraint may lie outside the space attainable by the thruster when operating at max thrust for the duration of the maneuver. These conflicting constraints would result in an infeasible problem formulation.

Alternatively, if the position constraint on the satellite in the above example were to be declared to be a “soft” constraint, which could be violated if necessary to produce a result, a solution would be returned that would guide the satellite to a point that is as near as possible to the constraint line while satisfying the “hard” constraint of maximum thrust. In this case, collision would be avoided, and, while the satellite would fall short of achieving the position constraint, it would be as close as possible to the desired location.

The constraint relaxation may be handled by adding weighted slack variables to the original problem formulation. The parameters will be added to the problem in such a way that problem formulations that contain feasible solutions will not be affected by them. In cases where the original problem would be infeasible, a solution would be returned that violates the soft constraints in a minimal way. For example, in the problem described above the satellite could not achieve the separation constraint while also satisfying the thrust constraint. In this situation the separation constraint would be relaxed so that the spacecraft would gain as much separation as possible in the time allotted.
The LP problem for the collision avoidance problem can be written in the following compact form:

\[
\begin{align*}
\text{Minimize} & \quad J = c^T x \\
\text{Subject to} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

where \( x \) is the vector of free parameters, \( c \) is a vector of weighting parameters on \( x \), \( A \) is the constraint matrix and \( b \) is the vector of constraint values. If inequality constraints are desired in the problem, they can be added to the LP through the addition of slack variables. For example, the inequality constraint

\[
x_1 + x_2 \leq 0
\]

can be transformed to an equality constraint with a slack variable:

\[
x_1 + x_2 + \alpha_1 = 0
\]

where \( \alpha_1 \) is the slack variable, which is added to the vector of unknown parameters \( x \). \( \alpha_1 \) is required to be greater than or equal to zero by the constraint in Eq. 31, therefore the remainder of the summation in Eq. 33 must be less than or equal to zero if the constraint is to be satisfied. The values of the slack variables are free and have no effect on the objective function. Accordingly, the weighting vector \( c \) should be appended with a zero so that the product \( c^T x \) is not changed by the presence of the slack variable.

Soft constraints can be relaxed through the addition of weighted slack variables, which are included in the cost function. The new augmented cost function includes the nominal cost plus the weighted sum of all slack variables associated with soft constraints. In principle, these weighted slack variables alter the problem in the following ways:

- When the satisfaction of all soft constraints is possible, the weighted slack variables are zero and thus do not affect the cost.
- When at least some soft constraints cannot be satisfied, slack variables relax the necessary constraints so that a solution becomes possible.
- The new solution minimizes an augmented cost function, including the nominal cost and the weighted sum of non-zero slack variables.

If the weight on the slack variables is sufficiently high, then the priority is to minimize the extent to which the soft constraints are violated. Otherwise, if the weight is sufficiently small, the priority remains to minimize the nominal cost function. One of the design challenges is to select a weighting that ensures the nominal cost is minimized with zero constraint violation when possible, but that also applies a proper balance to problem when constraints must be violated.

Consider the example constraint given in Eq. 33. If we wish to make this constraint a “soft” constraint that is able to be relaxed, we would subtract a weighted slack variable \( \beta_1 \).

\[
x_1 + x_2 + \alpha_1 - \beta_1 = 0
\]

As stated previously, the slack variable \( \alpha_1 \) can have any positive value without violating the inequality constraint. Conversely, the original inequality constraint is violated when the weighted slack variable \( \beta_1 \) is greater than \( \alpha_1 \). If \( \beta_1 \) is zero, then the constraint is satisfied. Ideally, the value of \( \beta_1 \) will be zero unless it is necessary to take value to allow a solution.

We can ensure that its value is minimized by giving its corresponding weighting parameter in the \( c \) vector a non-zero value. If the weighting parameter is a very large number, then the objective function is penalized greatly for the violation of the constraint. The linear program drives the contribution of the weighted slack variable to zero if possible. If it is not possible, then a solution can still be found, whereas satisfying a hard constraint would have been infeasible. In this situation \( \beta_1 \) becomes a measure of the distance of the solution from the constraint, and this distance, as part of the objective function, is minimized by the linear program.

The implementation of this procedure does not necessarily meet the objectives listed above in all cases. Specifically, it is possible for the weighted slack variable to be non-zero even for a case where a hard constraint could be satisfied. This occurs when the weight applied to the slack variable is not large enough to overwhelm other elements of the objective function.
A simple example to demonstrate this relationship is the problem of minimizing the parameter $x_1$ with a constraint that $x_1 \geq 1$. Clearly, the solution should be $x_1 = 1$. With a soft constraint, the objective function would be

$$J = c_1 x_1 + c_2 \beta_1.$$  \hfill (35)

The constraint equation would be

$$x_1 - \alpha_1 + \beta_1 = 1.$$  \hfill (36)

This equation can be solved for $\beta_1$ and plugged into Eq. 35 to get a new objective function that now includes a penalty function. Eq. 37 becomes the objective function in the case where $\beta_1$ is nonzero, or, equivalently, where the constraint is violated.

$$J = c_1 x_1 + c_2 (\alpha_1 - x_1 + 1).$$  \hfill (37)

If the derivative of this function is taken with respect to $x_1$ we can see the relationship between the weighting functions for this particular problem.

$$J' = c_1 - c_2.$$  \hfill (38)

According to this equation, if the value of $c_1$ is greater than $c_2$, then the slope of the objective function with respect to $x_1$ is positive. In this situation, the solver would continue to decrease the value of $x_1$ even though it would increasingly violate the constraint. Alternatively, the slope of the objective is negative if $c_2$ is greater than $c_1$. The solver would increase $x_1$ in this situation until the value of the slack was driven to its minimum (in this case, zero). Therefore as long as $c_2 > c_1$ the penalty function will have the desired effect on this problem. As a general result, we can conclude that large weights on the penalty functions will produce the desired results. The measure of largeness can depend on a particular problem.

### 2. Simple Example of Automatic Relaxation

To demonstrate the utility of automatic relaxation, we add a second constraint that makes the problem infeasible. Consider the following infeasible LP.

**Infeasible Problem**

\[
\begin{align*}
\text{Minimize} & \quad c_1 x \\
\text{Subject to} & \quad x \geq 1 \\
& \quad x \leq 0.5 \\
& \quad x \geq 0
\end{align*}
\]

Clearly both constraints cannot be met, and so the problem is infeasible. Let us now keep the first constraint and relax the second by introducing a weighted slack variable $\beta$. The problem becomes:

**Relaxed Feasible Problem**

\[
\begin{align*}
\text{Minimize} & \quad c_1 x + c_2 \beta \\
\text{Subject to} & \quad x \geq 1 \\
& \quad x - \beta \leq 0.5 \\
& \quad x, \beta \geq 0
\end{align*}
\]

With the addition of $\beta$, the problem is now feasible. This simple problem is illustrated graphically in Figure 3. The independent regions where each constraint is satisfied are shown in yellow and blue. The feasible space, shown in green, is the intersection of the other two areas. This illustrates how the addition of the $\beta$ dimension introduces feasibility.

Note that the solution may either be bounded or unbounded, depending upon the direction of the cost vector $(c_1, c_2)$. Because the purpose of constraint relaxation is to allow certain constraints to be violated only when necessary, we wish to minimize $\beta$. Therefore, we restrict the coefficient on $\beta$ to be positive, $c_2 > 0$. For a minimization problem, $c_1 > 0$ and the solution is $x = 1, \beta = 0.5$. If, instead, the objective is to maximize $x$, then the coefficient on $x$ changes to $c_1 < 0$. The optimal solution remains $(1, 0.5)$ as long as $c_2 > |c_1|$.
In other words, this solution holds as long as it is more important to minimize $\beta$ than it is to maximize $x$. If $c_2$ drops below the absolute value of $c_1$, then the optimal solution changes. In this case, the solution would change by moving up the line defined by $x - \beta = 0.5$, either going unbounded towards infinity or until another constraint is met.

### 3. Final Problem Formulation

Now that constraint relaxation has been introduced, we may present the final problem formulation. The complete problem includes the avoidance constraints from Equations 18 and 20, as well as a relaxed form of the station keeping constraints from Equations 11 and 13. The problem is given below:

**Complete Problem**

Minimize

$$\begin{align*}
\sum_{j=1}^{N} c_j (|u_j(1)| + |u_j(2)| + |u_j(3)|) + W||s||_1
\end{align*}$$

Subject to

$$\begin{align*}
-4z_N + 2\dot{x}_N/n &= 0 \\
\cos \theta_j y_j + \sin \theta_j z_j - f_j &\geq R_j \\
\frac{\partial}{\partial \theta_j} (6z_j n - 3\dot{x}_j) &\geq \frac{|R'|_{\theta}}{2\pi} \\
-x_B - s_{xLj} &\leq x_j \leq x_B + s_{xUj} \\
y_B - s_{yLj} &\leq y_j \leq y_B + s_{yUj} \\
z_B - s_{zLj} &\leq z_j \leq z_B + s_{zUj} \\
-u_{\text{max}} &\leq \tilde{u} \leq u_{\text{max}} \\
s &\geq 0
\end{align*}$$

The path separation constraint for avoidance includes the uncertainty term $f_j$, as defined in Eq. 26. The two decision vectors include $\tilde{u}$, the acceleration in each axis at discrete points in time, and $s$, the set of slack variables associated with relaxation of the station-keeping constraints. The components of $s$ include the slacks for the lower and upper bounds of all 3 dimensions, and so $s \in \mathbb{R}^{6N \times 1}$. Note that the cost function consists of the original summation over $\tilde{u}$, which is the total control effort, as well as the new term $W||s||_1$, the weighted one-norm of the relaxation slack variables. By selecting the weight $W$ to be sufficiently high, the optimal solution will enforce $s = 0$ automatically, if such a solution is feasible. Otherwise, the station-keeping constraints will be relaxed as necessary. In such a case, the cost will include the total control effort as well as the weighted sum of the non-zero slacks.
V. Simulation Results

This section presents the simulation results from a few different examples of collision avoidance planning. Together, these examples illustrate the effectiveness of using automatic constraint relaxation, as it provides a graceful method for handling multiple constraints in conflict.

For the results presented here, a LEO orbit of 350 km altitude is assumed and we use a maximum acceleration of 3 cm/s/s. This would correspond to 15 Newtons of thrust on a 500 kg spacecraft, for example. The controlled satellite is initialized at relative state $x_0 = 0$, so it begins precisely at its desired orbit state, in the center of the station-keeping box. The station-keeping bounds are arbitrarily defined as $\pm 0.6$ km cross-track, $\pm 1.2$ km radial, and $\pm 2.0$ km along-track. The avoidance constraint in the radial / cross-track plane is 1.0 km. We initialize a secondary object (representing a piece of debris or failed satellite) at 5.0 km along-track offset, 0.4 km radial, 0.1 km cross-track, and with a radial velocity of 0.74 meters per second. With this initial state, the object would pass within 350 meters of the satellite in less than 2 orbits. Disturbances from J2, solar pressure and atmospheric drag are presently ignored.

1. Collision Avoidance Maneuver

We first examine a maneuver planned where only the collision avoidance constraints are enforced. The station-keeping constraints are ignored. In this case, the uncertainty in the initial state estimate is also assumed zero. The plots in Figure 4 show different views of the relative trajectories of both the debris object and controlled satellite in a common reference frame. The controlled satellite starts at the origin of the frame, with the object drifting towards it in the along-track direction from an initial distance of 5 km. A maneuver is planned over 2 orbit periods. The cross-track / radial projection view shows how the satellite’s maneuver creates a sustained separation distance from the object.

![Figure 4. Relative Trajectory of Object and Maneuvering Satellite in Local Reference Frame](image)

The plot on the left of Figure 5 shows the time history of the relative position between the satellite and the object. The original separation distance in the cross-track / radial plane corresponds to the distance that would have evolved had the satellite not maneuvered. Clearly, the new separation distance created from the maneuver achieves the minimum required value of 1.0 km. Notice that the object continues to drift past the satellite in the along-track direction after the maneuver has completed.

The top right plot of Figure 5 shows the actual separation distance in the cross-track radial plane along with the required separation. Note that the required separation ramps up from 0.0 to 1.0 km over the course of a half-orbit period. Ramping up from zero is necessary in order to ensure feasibility, since the initial
separation may be small. The control history is shown in the lower plot. Three radial burns and one cross-track burn are required, amounting to a total delta-v of 2.67 m/s.

Figure 5. Time Histories of Relative Position, Separation Constraint and Applied Delta-V

2. Collision Avoidance with Relaxed Station-Keeping Constraints

We now consider the same initial conditions and avoidance constraints as in the previous example, but we add the station-keeping constraints. The initial part of the trajectory is identical to the previous example, with the satellite maneuvering to maintain the minimum required separation distance in the cross-track / radial plane. However, to do so violates the station-keeping constraints. Because the station-keeping constraints are relaxed, the relaxation slack variables are allowed to become non-zero. After 4 orbits, though, the minimum separation distance drops to zero, allowing the soft station-keeping constraints to be met.

Figure 6. Separation Constraint, Control History, and Relative Trajectory with Station-Keeping Constraints

The time history of the separation constraint and applied controls are shown on the left side of Figure 6. Several perspective views of the trajectory are shown on the right. The top-right plot shows the station-keeping box with a green dashed line, and a circle of the minimum separation distance with a red dashed
The hard avoidance constraints are enforced during the first 4 orbits, and so we see this portion of the trajectory staying outside the red circle, and at times also going outside of the green box. The avoidance constraints are effectively dropped after 4 orbits, because the satellite has already drifted about 5 km past in the along-track direction. A small maneuver is performed at near the 4 orbit mark. The objective is to minimize fuel and meet the station-keeping constraints so that there is no cost from the weighted slack variables. The cheapest option is to reduce the radial and cross-track amplitudes just enough so that the satellite stays within the station-keeping box. The total delta-v for this scenario is 3.31 m/s. Thus, an additional 0.64 m/s is required after the initial avoidance maneuver in order to move back inside the station-keeping box.

3. Robust Maneuvers with Respect to Bounded Covariance

The previous two examples did not account for uncertainty in the initial relative state estimate. We now include a model of the uncertainty by introducing a bounded covariance matrix, with 10 meters 1-sigma uncertainty for position in each direction, and 1 meter per second for velocity in each direction. The worst-case realization of this uncertainty is added to each avoidance constraint inequality.

The perspective views of the relative trajectory are shown in Figure 7. The avoidance maneuver appears similar to that of the previous examples, except that a larger separation distance is achieved. This is evident in Figure 8, which shows the time-varying separation constraint that exceeds the nominal 1.0 km distance. This additional buffer is the result of propagating the covariance matrix over time and allowing it to take on the worst-case value for each separation constraint. The minimum separation distance in the cross-track / radial plane now varies between 1.0 and 3.4 km. Note that because the system is linear, doubling the magnitude of the covariance matrix entries would double the magnitude of this time-varying buffer distance. The delta-v for this more conservative maneuver is much higher, at 16.29 m/s.

Figure 7. Relative Trajectory of Object and Maneuvering Satellite with Robust Avoidance Method

VI. Conclusions

The growing risk of collision with debris and inactive satellites motivates the need for reliable avoidance maneuver planning methods. Methods are required that can plan fuel-efficient avoidance maneuvers that are insensitive to relative navigation errors while also maintaining the desired station on-orbit. This paper develops a linear programming formulation for collision avoidance maneuver planning that incorporates station-keeping bounds as soft constraints. A time-varying separation constraint is defined in the radial / cross-track plane, enabling the original non-convex avoidance set to be modeled with a discrete set of linear constraints.
inequalities. The method of automatic constraint relaxation is used by incorporating into the cost function a weighted sum of the slack variables associated with the violation of station-keeping bounds. The resulting method enables conflicting station-keeping constraints and avoidance constraints to be included by allowing station-keeping bounds to be violated only when necessary. Uncertainty in the initial state is accounted for by using an approximate covariance matrix for the relative position and velocity, and incorporating the worst-case effect from this uncertainty into each separation inequality constraint. This produces an avoidance maneuver that will maintain the desired separation distance for any initial state lying in the uncertainty ellipsoid defined by the covariance matrix. Simulation results demonstrate the effectiveness of the robust, automatic relaxation method, producing avoidance maneuvers that initially satisfy the conservative separation distance, then contract with minimum fuel usage to stay within the station-keeping box after the debris has safely passed. This method may be expanded in the future by considering coordinated avoidance in the presence of multiple objects. It may also be utilized in the mission planning phase to approximate the delta-v that would be required to avoid known debris that occupies the desired orbital slots.

References


