Mitigation of the Impact of Sensing Noise on the Precise Formation Flying Control Problem

Short Title: Sensing Noise in Formations of Spacecraft
NASA Cooperative Agreement NCC5–729

Final Report

PI Dr. Jonathan P. How
Massachusetts Institute of Technology
Department of Aeronautics & Astronautics
Bldg. 33 Room 326, 77 Massachusetts Avenue
Cambridge, MA 02139-4307
617-253-3267 (phone)
617-253-7397 (fax)
jhow@mit.edu

GSFC Technical Officer: David A. Quinn
Guidance, Navigation & Control Division
Flight Dynamics Analysis Branch
NASA GSFC Code 595
301-286-6681 (phone)
301-286-0369 (fax)
David.A.Quinn@nasa.gov
1 Introduction

The work conducted for this grant is documented in the attached papers and thesis (see also Refs. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]).

References


[12] L. Breger and J. P. How, “GVE-based MPC for Formation Flying Satellites with Distur-


Analytical performance prediction for robust constrained model predictive control

ARTHUR RICHARDS*†, LOUIS BREGER‡ and JONATHAN P. HOW‡

†Department of Aerospace Engineering, University of Bristol, Queens Building, University Walk, Bristol, BS8 1TR, UK
‡Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

(Received 4 July 2005; revised 13 January 2006; in final form 15 March 2006)

This paper presents a new analysis tool for predicting the closed-loop performance of a robust constrained model predictive control (MPC) scheme. Currently, performance is typically evaluated by numerical simulation, leading to an extensive computation when investigating the effect of controller parameters, such as the horizon length, the cost weightings and the constraint settings. The analytic method, in this paper, avoids this computational burden, thus enabling a rapid study of the trades between the design parameters and the performance. Previous work developed an MPC formulation employing constraint tightening to achieve robust feasibility and constraint satisfaction despite the action of an unknown but bounded disturbance. This paper shows that the expected performance of that controller can be predicted using a combination of the gains of two linear systems, the optimal control for the unconstrained system, and a candidate policy used in performing the constraint tightening. The method also accounts for the possible mismatch between the predicted level of disturbance and the actual level encountered. The analytic results are compared with simulation results for several examples and are shown to provide accurate predictions of performance and its variation with the system parameters.

1. Introduction

This paper provides an analytical method of predicting the closed-loop performance of a recently developed robust model predictive control (MPC) scheme (Richards and How 2006). The MPC performance is rarely considered explicit, and is typically evaluated by the numerical simulation. Investigating the effects of various controller settings therefore requires an extensive computation. The contribution of this paper is a method of analytically predicting the performance of a system controlled by MPC, enabling the rapid investigation of the effect of controller parameters without recourse to extensive simulation.

MPC is a popular technology in the process industry (Richalet 1993, Maciejowski 2002) and has promising applications in other fields, such as in aerospace (Manikonda et al. 1999, Dunbar and Murray 2002, Inalhan et al. 2002, Richards and How 2003). MPC uses the online solution of a numerical optimization problem and can readily accommodate hard constraints, such as relative position tolerances (or “error boxes”) in spacecraft formation flight (Inalhan et al. 2002) and collision avoidance in unmanned aerial vehicle (UAV) guidance (Richards and How 2003). Stability and robustness results for constrained MPC are well addressed by existing work (see, for example, Mayne et al. 2000). In Richards and How (2006), the authors describe a formulation for MPC offering robust feasibility and show that feasibility of each optimization and satisfaction of the constraints are guaranteed, despite the action of an unknown but
bounded disturbance. This method is an extension of earlier work (Gossner et al. 1997, Chisci et al. 2001) involving tightening constraints and retaining a “margin” in each plan for compensation against future disturbances. The margin is chosen such that a predetermined feedback policy can always be employed to counteract the disturbance, although in practice the online optimization usually finds a better solution than the application of that policy. This method guarantees constraint satisfaction, which can, in turn, provide bounds on performance, but these bounds are typically conservative. In contrast, the new prediction method in this paper provides a direct estimate of the expected performance of the closed-loop system.

Figure 1 gives an overview of the performance prediction method. Figure 1(a) shows the system block diagram, in particular, the roles of the actual and the predicted disturbance levels. The predicted level is the designer’s estimate of the actual disturbance level and forms a disturbance model that is used to make the controller robust. The performance analysis is based on two observations of the variation of performance over the space of actual and predicted disturbance levels, shown in figure 1(b). Only the lower triangle is considered: if the actual disturbance is higher than the predicted level, feasibility is not guaranteed. The first observation is that, if the actual disturbance is sufficiently low, corresponding to the lightly shaded unconstrained region in figure 1(b), the controller behaves like the optimal finite-horizon regulator for the unconstrained system. This is not surprising in itself, but one of the contributions of this paper is a method for quantifying “sufficiently low” for this behavior as a function of system and controller parameters. Observe that the upper limit of the unconstrained region varies with the predicted disturbance, as this determines the constraint tightening. The second observation is that, at the limit of feasibility, shown by the heavy line on the right of figure 1(b), the controller behaves like the predetermined candidate control used to calculate the margin, and its performance is therefore predictable using the linear system tools. This behavior occurs because the MPC optimization has only one solution at the limit of feasibility. Between the end of the unconstrained region and the limit of feasibility lies the transition region, shaded dark gray in figure 1(b). In this region, we approximate the performance using a smooth interpolation between the unconstrained behavior and the predetermined control policy. The key technical challenges of performing the prediction are determining, analytically, the limit of feasibility and the upper limit of the unconstrained region of operation.

Section 2 defines the problem statement for performance prediction. Section 3 reviews the robust MPC algorithm from Richards and How (2006). The main result, providing analytical prediction of performance, is described in §4. Section 5 presents modifications to the prediction algorithm to include the effect of estimation uncertainty. Section 6 compares the

![Disturbance magnitude, $W_d$](image1)

![Plant](image2)

![MPC](image3)

![Disturbance prediction, $W_p$](image4)

(a) Framework for performance prediction

![Regions of operation](image5)

(b) Regions of operation

Figure 1. Overview of performance prediction method. In (a), $z$ is the performance output, and in (b), $W_{\text{max}}$ is the greatest level of predicted disturbance that allows a feasible optimization. The two shaded areas are those in which performance predictions can be made.
Analytical performance prediction for robust MPC

2. Problem statement

The objective is to predict the expected closed-loop performance of a discretized, linear time invariant system subject to constraints, acted upon by persistent, unknown but bounded disturbances, and controlled by the robust MPC from Richards and How (2006). This controller is known to guarantee the constraint satisfaction. The contribution of this paper is the development of the performance prediction method.

Let the system have the following dynamics:

\[ x(k+1) = Ax(k) + Bu(k) + Ew(k), \]
\[ y(k) = Cx(k) + Du(k), \]
\[ z(k) = Fx(k) + Gu(k), \]

where \( x(k) \in \mathbb{R}^{N_x} \) is the state vector, \( u(k) \in \mathbb{R}^{N_u} \) is the controlled input, \( w(k) \in \mathbb{R}^{N_w} \) is the disturbance input, \( y(k) \in \mathbb{R}^{N_y} \) is the constrained output and \( z(k) \in \mathbb{R}^{N_z} \) is the performance output. The problem considered in this paper is to predict approximately the root mean square (RMS) value of the performance output \( z \), i.e. \( \sqrt{E[z(k)^T z(k)]]} \).

The pair (A, B) is assumed to be controllable and the matrix E is assumed to have full row rank such that \( EE^T \) is positive definite. The matrices C and D are chosen by the designer to form the constrained outputs, which are required to remain within a bounded set

\[ y(k) \in \mathcal{Y} \quad \forall k, \]

where the set \( \mathcal{Y} \), also chosen by the designer, is a polytope defined by \( N_p \) inequalities

\[ \mathcal{Y} = \{ y | p_n^T y \leq g_n \ \forall n \in 1, \ldots, N_p \}. \]

Furthermore, the set \( \mathcal{Y} \) is assumed not to be of zero measure and to contain the origin, \( \mathbf{0} \in \mathcal{Y} \). This form of output constraints can capture both the input and the state constraints, or the mixtures thereof, such as a limited control magnitude or an error requirement.

The disturbance is unknown but bounded. It is assumed to be uncorrelated and uniformly distributed in a hypercube

\[ w(k) \in B_{\infty}(W_d) \quad \forall k, \]

where

\[ B_{\infty}(W) = \{ w | \|w\|_{\infty} \leq W \}. \]

We do not assume that \( W_d \) is known \textit{a priori}. The system is controlled using the robust MPC from Richards and How (2006), which is reviewed in §3. The controller will be designed to suit a disturbance level \( W_p \), which need not equal \( W_d \). Thus, the prediction method considers the implication of inaccurate disturbance modeling.

3. Robustly feasible MPC

This section reviews the robustly feasible MPC formulation from Richards and How (2006) with application to the problem from §2. The robust feasibility result is restated with an outline of the proof. Then in §4, a new method for predicting the resulting closed-loop performance is developed.

The core of any MPC is the optimization, which is performed online at each time step. In this case, the optimization minimizes a quadratic cost over a finite horizon subject to constraints tightened to suit the predicted disturbance level. We define the robust MPC optimization problem \( \mathbf{P}(x(k), W_p) \), starting from state \( x(k) \) with predicted disturbance bound \( W_p \), as follows:

\[
J^*(x(k), W_p) = \min_{u, x, y} \sum_{j=0}^{N} (x^T(k+j|k)Qx(k+j|k) + u^T(k+j|k)Ru(k+j|k))
\]

subject to \( \forall j \in \{0, \ldots, N\} \)

\[ x(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k), \]
\[ y(k+j|k) = Cx(k+j|k) + Du(k+j|k), \]
\[ x(k|k) = x(k), \]
\[ x(k+N+1|k) = 0, \]
\[ y(k+j|k) \in \mathcal{Y}(j; W_p), \]

where \( N \) is the planning horizon. The choice of \( N \) is the responsibility of the designer, but the analysis tool presented in this paper enables investigation of its effect on performance. \( R \) and \( Q \) are symmetric weighting matrices, also chosen by the designer, with \( Q \) positive definite and \( R \) positive semi-definite. The set \( \mathcal{Y}(j; W_p) \) denotes the constraints on the \( j \)th predicted step, tightened to accommodate...
a predicted disturbance level \(W_p\) (Richards and How, 2006). Tightening is performed by the recursion
\[
\mathcal{Y}(j; W_p) = \mathcal{Y}, \quad (12a)
\]
\[
\mathcal{Y}(j+1; W_p) = \mathcal{Y}(j; W_p) - (C + DK_H)\mathbf{L}_H(j)E\mathcal{B}_\infty(W_p)
\quad \forall j \in \{0, \ldots, N - 1\} \tag{12b}
\]
where the set \(\mathcal{B}_\infty(W_p)\) is the predicted disturbance set. As discussed in §2, the predicted disturbance limit \(W_p\) is not assumed to be equal to the actual limit \(W_a\). The operator ‘\(\sim\)’ denotes the Pontryagin difference (Kolmanovsky and Gilbert 1995) defined by
\[
A \sim B = \{a | a + b \in A \forall b \in B\}. \tag{13}
\]
A Matlab routine is available (Kerrigan 2005) for calculating the Pontryagin difference between two polytopes, and the result is also a polytope, with only the right-hand side of the inequalities changed (Kerrigan 2000), i.e. of the form
\[
\mathcal{Y}(j; W_p) = \{y | p^T_y \leq q_p(j; W_p) \forall n \in 1, \ldots, N_p\} \tag{14}
\]
where the values of \(q_p(j; W_p)\) are determined by the Pontryagin difference algorithm. The controller \(K_H\) in (12b) is chosen by the designer such that the static linear feedback control law \(u = K_Hx\) makes the system nilpotent in at most \(N\) steps. Also in (12b), define \(\mathbf{L}_H(j)\) as the state transition matrix for the closed-loop system under this control law
\[
\mathbf{L}_H(0) = I, \quad (15)
\]
\[
\mathbf{L}_H(j + 1) = (A + BK_H)\mathbf{L}_H(j) \quad \forall j \in \{0, \ldots, N - 1\}. \tag{16}
\]
The optimization \(P(x(k), W_p)\) is employed in the following algorithm.

**Algorithm 1 (Robustly Feasible MPC):**

1. Solve problem \(P(x(0), W_p)\).
2. Apply control \(u(k) = u^*(k|k)\) from optimizing sequence.
3. Increment \(k\). Go to 1.

**Theorem 1 (Robust Feasibility):** If \(P(x(0), W_p)\) has a feasible solution and \(W_p \geq W_a\), then under the control of Algorithm 1 and the action of a disturbance obeying (4), all subsequent optimizations \(P(x(k), W_p)\) are feasible.

Since the origin \(0\) is an invariant terminal constraint set and, for \(W_p \geq W_a\), \(w(k) \in \mathcal{B}_\infty(W_a) \subseteq \mathcal{B}_\infty(W_p)\), then the proof of robust feasibility in (Richards and How 2006) holds. For brevity, only an outline is given here. Assuming that a feasible solution is known for problem \(P(x(k), W_p)\), then a solution can be constructed for problem \(P(x(k + 1), W_p)\) by shifting the previous solution by one step and adding a perturbation using controller \(K_H\) to accommodate the disturbance. Since this solution can be shown to be feasible for problem \(P(x(k + 1), W_p)\) for all disturbances obeying (4), it follows that the optimization itself is feasible. Thus, feasibility at time \(k\) implies feasibility at time \(k + 1\), and the theorem follows by recursion.

**Remark 1:** The formulation shown above is specialized to suit the performance analysis method developed in this paper. Robust feasibility can be proven under much more general conditions (Richards and How 2006), with a generalized cost function and terminal constraints, with no requirement to find optimal solutions at each step, and without the assumption of any particular distribution of the disturbance within its bounding set.

### 4. Analytical performance prediction

This section presents the main result of this paper. The performance prediction is based on the following three assumptions.

A1 For sufficiently low levels of actual disturbance \(0 \leq W_a \leq W_c(W_p)\) (defined below), the expected steady-state performance of a system under the control of Algorithm 1, expressed as the RMS value of the performance output (1c) is approximated by
\[
\sqrt{\mathbb{E}[z(k)^Tz(k)]} \approx G(K_L)W_a, \tag{17}
\]
where
- \(G(K)\) is the gain from the infinity-norm of the disturbance to the two-norm of the output of the system under a static, stabilizing, linear control \(u(k) = Kx(k)\),
- \(K_L\) is the finite-horizon linear quadratic regulator (LQR) controller for the unconstrained system,
- \(W_c(W_p)\) is the level of actual disturbance at which the constraints, tightened for predicted disturbance \(W_p\), begin to influence the performance. This defines the line separating the transition region and unconstrained region in figure 1(b).
A2 At the limit of feasibility \( W_p = W_{\text{max}} \), the performance is approximated by

\[
\sqrt{E[z(k)^Tz(k)]} \approx G(K_H)W_a
\]

where \( W_{\text{max}} \) is the highest level of predicted disturbance \( W_p \) that has a non-empty set of feasible initial states for the optimization \( \mathcal{P}(x(k), W_p) \).

A3 In the transition region, between the unconstrained region and the limit of robustness \( W_c(W_p) \leq W_a \leq W_p \leq W_{\text{max}} \), the performance can be approximated as:

\[
\sqrt{E[z(k)^Tz(k)]} \approx \lambda(W_p, W_a)G(K_L)W_a + [1 - \lambda(W_p, W_a)]G(K_H)W_a,
\]

where \( \lambda(W_p, W_p) \) is an interpolation function such that \( \lambda(W_c(W_p), W_a) = 0 \) for any \( W_p \) (i.e. on the dividing line between the constrained and the unconstrained regions in figure 1(b) and \( \lambda(W_a, W_{\text{max}}) = 1 \) for any \( W_a \neq 0 \) (i.e. at the limit of feasibility). With this definition, the performance gain is \( G(K_L) \) at the edge of the unconstrained region and \( G(K_H) \) at the limit of feasibility.

These assumptions represent the observations expressed in figure 1(b). A1 captures the observation that at sufficiently small disturbance levels, the constraints do not impact the performance. This is reasonable, since the requirements placed on the constraint set \( \mathcal{Y} \) in §2 ensure that the origin is away from the constraint boundaries, and therefore if the state remains in a small region around the origin, the constraints do not have any effect on the optimization solution. A2 represents the observation that at the limit of feasibility, the only available solution to the optimization is the candidate solution. A3 approximates the performance in the transition region by smoothly interpolating between the known performance levels on its boundaries.

The performance approximations (17)–(19) can be combined into a single expression.

\[
\sqrt{E[z(k)^Tz(k)]} \approx \begin{cases} \lambda(W_p, W_a)G(K_L)W_a & 0 \leq W_a \leq W_c(W_p) \\ \lambda(W_p, W_a)G(K_L)W_a + [1 - \lambda(W_p, W_a)]G(K_H)W_a & W_c(W_p) \leq W_a \leq W_p \end{cases}
\]

This approximation, and the assumptions A1–A3 underlying these, will be tested in simulation in §6.

The following subsections describe how the quantities involved are calculated.

### 4.1 Gain under linear control

Under the assumptions concerning the disturbance made in §2, the covariance matrix of the disturbance for unit input level \( W_a = 1 \) is given by

\[
E[w(k)w(j)^T] = \begin{cases} 1 & k = j \\ 0 & \text{otherwise} \end{cases}
\]

Then, under the assumption that the state and input can be treated as normal random variables, the system gain is given by

\[
G(K) = \sqrt{\text{tr}[S(K)X(K)S(K)^T]} \tag{21}
\]

with \( S \) and \( X \) satisfying

\[
X(K) = (A + BK)X(K)(A + BK)^T + \frac{1}{3}EE^T \tag{22}
\]

\[
S(K) = F + GK \tag{23}
\]

where \( X(K) = E[x(k)x(k)^T] \) is the state covariance matrix under control \( K \) and can be found by solving the discrete Lyapunov equation (22). Since \( EE^T \) is assumed to be positive definite, the equation (22) has a unique positive definite solution \( X(K) \) for any stabilizing controller \( K \). The validity of this analysis will be demonstrated by examples in §6.

**Remark 2** (Alternative Performance Metric): It may be desirable to express the performance in terms of a quadratic cost

\[
\sqrt{E[u(k)^T\tilde{R}u(k) + x(k)^T\tilde{Q}x(k)]}
\]

where \( \tilde{Q} \) and \( \tilde{R} \) are weighting matrices for the cost under investigation and need not match those used in the optimization cost (6). This can be achieved by setting

\[
\begin{align*}
F &= \begin{bmatrix} \tilde{Q} & 0 \\ 0 & R \end{bmatrix} \\
G &= \begin{bmatrix} 0 \\ \tilde{R} \end{bmatrix}
\end{align*}
\]

using the Cholesky factors \( \tilde{Q}^{1/2} \tilde{Q} = \tilde{Q} \) and \( \tilde{R}^{1/2} \tilde{R} = \tilde{R} \).

### 4.2 Unconstrained control solution

The controller \( K_L \) is the unconstrained finite-horizon LQR solution. This is found using the following
algorithm, based on the dynamic programming derivation of LQR with a fixed terminal constraint that the system reaches the origin.

\begin{align}
P(N+1) &= 0 \\
S(N+1) &= I \\
\forall j \in \{N, N-1, \ldots ,0\}:
\end{align}

\begin{align}
K(j) &= -[I \quad 0]
\begin{bmatrix}
(R + B^T P(j+1)B) & B^T S(j+1) \\
S(j+1)B & 0
\end{bmatrix}^+ \\
\times \begin{bmatrix}
B^T P(j+1) \\
S(j+1)B
\end{bmatrix} \ A \\
S(j) &= S(j+1)(A + BK(j)) \\
P(j) &= Q + K(j)^T RK(j) \\
+ (A + BK(j))P(j+1)(A + BK(j)) \\
K_L &= K(0)
\end{align}

where \(M^T\) denotes the pseudoinverse of matrix \(M\). It is also necessary to express the predicted outputs associated with the unconstrained solution as a function of the initial state \(x(k)\). Write

\begin{equation}
y(k+j|k) = H(j)x(k),
\end{equation}

where

\begin{equation}
H(j) = (C + DK(j))L(j)
\end{equation}

and the matrix \(L\) is the state transition matrix given by recursion

\begin{align}
L(0) &= I, \\
L(j+1) &= (A + BK(j))L(j) \quad \forall j \in \{0, \ldots ,N\}.
\end{align}

These relations are used in §4.4 to find the limit of the unconstrained region.

4.3 Limit of feasibility

The problem \(P(x(k), W_p)\) becomes more constrained as the predicted disturbance \(W_p\) is increased. The limit of feasibility \(W_{max}\) is defined as the greatest value of \(W_p\) for which the set of feasible initial states for the problem \(P(x(k), W_p)\) is non-empty. Since the terminal constraint is the origin, the problem will be infeasible if the origin is not contained in the tightest constraint set \(\mathcal{Y}(N; W_p)\) defined in (12b). Therefore, the limit of feasibility is found by solving the following optimization:

\begin{equation}
W_{max} = \max_{W_p} W_p \\
\text{s.t.} \quad 0 \in \mathcal{Y}(N; W_p).
\end{equation}

By the definition of the Pontryagin difference (13), the condition

\[0 \in A \sim B\]

is equivalent to

\[b \in A \quad \forall b \in B\]

Further, if \(A\) is convex and \(B\) is a mapping of a unit hypercube of dimension \(N_v\) through a matrix \(M\), this can be expressed in terms of the vertices of \(B\)

\[Mv_i \in A \quad \forall i \in \{1, \ldots ,2^{N_v}\}\]

where \(v_i\) are the vertices of the unit hypercube, which are straightforward to evaluate. Extending this principle to the recursion for \(\mathcal{Y}(N; W_p)\) in (12b) gives the following optimization to find \(W_{max}\)

\begin{equation}
W_{max} = \max_{W_p} W_p \\
\text{s.t.} \quad W_p \sum_{j=0}^{(N-1)} (C + DK(j))L_H(j)Ev_j \in \mathcal{Y} \\
\forall (i_0, i_1, \ldots , i_{(N-1)}) \in \{1, \ldots ,2^{N_v}\} \times \cdots \times \{1, \ldots ,2^{N_v}\}
\end{equation}

Observe that the constraint has to be evaluated for all combinations of vertices of the unit hypercube. Since \(\mathcal{Y}\) is a polytope (3), this can be rewritten as

\begin{equation}
W_{max} = \min_{(n_0, n_1, \ldots , n_{(N-1)})} \left(\frac{\sum_{j=0}^{(N-1)} (C + DK_H(j))L_H(j)Ev_j}{P_n^{(N-1)}}\right)
\end{equation}

which can be readily found, despite the large numbers of constraints to check.

4.4 Limit of unconstrained operation

The quantity \(W_c(W_p)\) is the greatest value of the actual disturbance \(W_a\) for which the constraints, tightened for a predicted disturbance \(W_p\), do not significantly influence the control performance. This corresponds to...
Analytical performance prediction for robust MPC

the division between the unconstrained region and the transition region, shown in figure 1(b). To calculate $W_C(W_p)$, we find the value of actual disturbance $W_a$ for which, if the closed-loop system behaves like the unconstrained optimal control $u(k) = K_L x(k)$, then 95% of the solutions to the optimal unconstrained problem (25) satisfy the constraints. The choice of 95%, as opposed to any other probability, is arbitrary and will be checked in simulations in §6.

If the system behaves like the unconstrained optimal control, and assuming that the state can be treated as a Gaussian random variable, then with 95% probability, the state resides in an ellipsoid

$$\frac{1}{W_a} x^T(k) X(K_L)^{-1} x(k) \leq 9$$

(32)

where $X(K_L)$ is the state covariance matrix (22) under unconstrained control (24f). This can be rewritten as a norm bound

$$\| Y x(k) \|_2 \leq 3 W_a$$

(33)

where the matrix $Y^T Y = X^{-1}(K_L)$. Since $X(K_L)$ is the positive definite solution to the Lyapunov equation (22), the matrix $Y$ exists and is invertible. Since the constraints (11) are of the form

$$p_n^T y(k+j) \leq q_n(j; W_p)$$

we use (33) to derive a bound (for 95% probability) on the quantity $p_n^T y(k+j)$ as follows:

$$p_n^T y(k+j) = p_n^T H(j) x(k)$$

$$= p_n^T H(j) Y^{-1} Y x(k)$$

$$\leq 3 W_a \| Y^{-T} H(j)^T p_n \|_2,$$

where $H(j)$ are given by (25). Therefore, using the limits $q_n(j; W_p)$ for the tightened constraints (14), the unconstrained solution is feasible, with 95% probability, if

$$3 W_a \| Y^{-T} H(j)^T p_n \|_2$$

$$\leq q_n(j; W_p) \forall n \in \{1, \ldots, N_p\} \forall j \in \{0, \ldots, N\},$$

(34)

then the value of $W_C(W_p)$ can be found by finding the greatest value of $W_a$ for which all conditions (34) hold, which can be done by simply checking each constraint and taking the minimum value

$$W_C(W_p) = \min_{n=1, \ldots, N_p, j=0, \ldots, N} \frac{q_n(j; W_p)}{3 \| Y^{-T} H(j)^T p_n \|_2}.$$  

(35)

4.5 Interpolation function

The function $\lambda(W_a, W_p)$ is used in the transition region for interpolation between the unconstrained regime and the limit of feasibility

$$\lambda = \left( \frac{W_a - W_C(W_p)}{W_a} \right)^r,$$

(36)

where $r > 2$ gives a smooth transition away from the unconstrained performance. As $W_p \to W_{max}$ then $W_C(W_p) \to 0$, hence also $\lambda \to 1$ for $W_a \neq 0$. Also, $\lambda \to 0$ as $W_a \to W_C(W_p)$.

4.6 Performance prediction algorithm

Finally, the calculations described in the preceding subsections are combined in the following algorithm for performance prediction.

Algorithm 2 (Performance prediction):

Data: $A, B, C, D, E, F, G, Y, K_H, N, Q, R, W_a, W_p$

(1) If $W_a > W_p$, return 0 (not robust). Stop.

(2) Calculate $W_{max}$ using (30). If $W_p > W_{max}$, return 0 (infeasible). Stop.

(3) Calculate the unconstrained controller $K_L$ using (24f), gains $G(K_L)$ using (21) and form the constraints $q_n(j; W_p)$ (14) using (12a) and (12b).

(4) Calculate $W_C(W_p)$ using (35).

(5) If $W_a \leq W_C(W_p)$, return $G(K_L) W_a$ (unconstrained region). Stop.

(6) Calculate gain $G(K_H)$ and interpolation function $\lambda$ using (36) and return

$$\lambda(W_p, W_a) G(K_L) W_a + [1 - \lambda(W_p, W_a)] G(K_H) W_a.$$

5. Estimation error

This section extends the performance prediction method of §4 to include the effect of uncertain state estimates. Richards and How (2005) show how the robust MPC from Algorithm 1 can accommodate uncertainty in the states by modifying the constraint and uncertainty sets, $Y$ and $W$. This section begins by reviewing that method and then describes how to predict the performance of the resulting MPC.

Assume the state estimate includes an additive error

$$\hat{x}(k) = x(k) + Me(k),$$
where the error $e(k) \in \mathbb{R}^{N_t}$ is uniformly distributed in a hypercube with the same norm limit as the disturbance bound (4)

$$e(k) \in B_\infty(W_p).$$

Also assume that the estimation error is white and independent of the disturbance $w(k)$. Differences in magnitude between the estimation error and the disturbance are handled by varying the matrices $E$ and $M$. It can be shown (Richards and How 2005) that the dynamics of the estimate are given by:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + \hat{E}w(k), \quad (37)$$

where

$$\hat{E} = [E \mid M \mid -AM], \quad \hat{w}(k) = \begin{bmatrix} w(k) \\ e(k+1) \\ e(k) \end{bmatrix}.$$ 

Therefore, Algorithm 1 guarantees the robust feasibility and the constraint satisfaction if $E$ is replaced by $\hat{E}$ in (12b) and $\hat{w}(k) \in B_\infty(W_p)$.

The performance prediction algorithm for the problem with estimation error is identical in form to Algorithm 2 applied to the estimate system (37) and with modifications to some of the quantities involved. The unconstrained solution is unchanged from (24f). The following subsections describe the modifications to the other calculations.

### 5.1 Gain under linear control

When the control is behaving like a linear feedback $u(k) = K\hat{x}(k)$ acting on the estimate, the dynamics of the estimate (37) can be rewritten as

$$\begin{bmatrix} \dot{\hat{x}}(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} A + BK & -AM \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\hat{x}}(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} E & M \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w(k) \\ e(k+1) \end{bmatrix}.$$ 

Note that this system appears unusual as it is driven by a “look-ahead” input $e(k+1)$. This is permissible in this case as only the statistical properties of the signal $e(k)$ are assumed to be known for the performance analysis. The performance gain under linear control from (21) can be rewritten for the estimation error case

$$G(K) = \sqrt{\text{tr}[\hat{S}(K)\hat{X}(K)\hat{S}(K)^T]}$$

with $\hat{S}$ and $\hat{X}$ satisfying

$$\dot{\hat{X}}(K) = \hat{A}\hat{X}(K)\hat{A}^T + \frac{1}{3}\hat{E}\hat{E}^T,$$

$$\dot{\hat{S}}(K) = [F + GK \mid -FM],$$

where

$$\hat{A} = \begin{bmatrix} A + BK & -AM \\ 0 & 0 \end{bmatrix}, \quad \hat{E} = \begin{bmatrix} E & M \\ 0 & 0 \end{bmatrix}.$$

### 5.2 Limit of feasibility

The calculation of the limit of feasibility is modified by replacing $Ew(k)$ with $\hat{E}w(k)$, and so (31) becomes

$$W_{\text{max}} = \min_{\{\{n, k, h, \ldots, k_{(n-1)}\}\}} \left\{ \frac{q_n}{\sum_{j=0}^{(N_t-1)} (C + DK_H(j))L_H(j)\hat{E}v_j} \right\}$$

where $v_j$ denotes the vertices of the hypercube containing $\hat{w}(k)$.

### 5.3 Limit of unconstrained operation

The calculation of $W_c(W_p)$ involves both the constraint tightening and the deviation due to the actual uncertainty, so the modifications of both the previous subsections are combined. The expression for the limit of unconstrained operation (35) becomes

$$W_c(W_p) = \min_{n \in \{1, \ldots, N_t\}} \hat{q}_n(j, W_p)$$

where $\hat{q}_n(\cdot)$ denotes the tightened constraint limits using the effective process noise $\hat{E}w(k)$ and $\hat{Y}$ is calculated from the covariance matrix for the augment state $\hat{X}(K)$ such that $\hat{Y}^T\hat{Y} = E[I\hat{x}(k)^T\hat{x}(k)]$. The matrices $H(j)$ depend only on the unconstrained LQR solution and are not changed by the inclusion of estimation error.
6. Examples

This section demonstrates the performance prediction approximation using Algorithm 2 by comparing the results of the new analytical prediction method with simulation results for a variety of systems. The aim here is to validate the assumptions A1–A3 underlying the prediction method in §§4 and 5 by showing that the predictions match the results from the simulations. Furthermore, the computation times for both the predictions and the simulations are compared to identify the computational savings associated with the new analytical performance prediction.

The first example investigates the variation of performance with the actual and the predicted disturbances, exploring the space shown in figure 1(b). The second example shows the variation of performance with constraint limits, demonstrating the application of the analytical performance prediction to study trades between design parameters. The third example investigates the effect of horizon length on performance, which is an important design issue for MPC that is often only addressed through extensive simulations. The fourth and final example in this section considers the effect of state uncertainty on performance. Section 7 provides a more detailed application to the problem of spacecraft formation flight.

The demonstrations in this section all involve two second-order systems, one neutrally stable and the other unstable. The respective system A matrices are

\[
A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.9 & 0.6 \\ 0.2 & 0.5 \end{bmatrix}
\]

with the following parameters common to both examples

\[
B = \begin{bmatrix} 0.5 \\ 1.0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}, \quad E = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}
\]

The constraints are a unit bound on the infinity-norm (i.e. a unit magnitude bound on each element)

\[
\mathcal{Y} = \{ y \in \mathbb{R}^3 \mid \| y(k) \|_\infty \leq 1 \}
\]

The optimization costs penalize the control energy with a smaller weighting on the state

\[
Q = I_2, \quad R = 100
\]

The problem is solved with a horizon of \( N = 6 \) steps and the interpolation (36) uses a power of \( r = 4 \).

6.1 Effect of expected and actual disturbance levels

The expected disturbance \( W_p \) and actual disturbance \( W_a \) are varied between 0 and the appropriate \( W_{\text{max}} \), precalculated for each system. A simulation of 1000 time steps is performed at each setting \( (W_a, W_p) \) over a \( 20 \times 20 \) grid. Figure 2 compares the performance predictions with simulation results for the metric of control effort. The predictions are found using Algorithm 2 with the performance output matrices \( F = [0 \ 0] \) and \( G = 1 \). In the prediction plots (figure 2a and 2c), the unconstrained region is white and the transition region shaded grey. In both the cases, the predictions closely match the results of the simulations.

The simulation results in figure 2(b) and (d) support the assumptions A1–A3. At low disturbance levels, the control RMS surface is flat as the controller behaves like the unconstrained LQR, validating assumption A1. At the limit of feasibility, seen as the high ridge on the right-hand side of plots, the performance plot follows a straight line again as the MPC behaves like the nilpotent candidate controller, validating assumption A2. The smooth transition between the LQR behaviour and the nilpotent candidate behaviour validates assumption A3.

A key motivation of developing the new analytical performance method was to reduce the computation required to perform studies of this type. It took 23 min to perform all the simulations used to generate figure 2(d), but less than 2 min to generate figure 2(c) using the analytical prediction method. It is clear, therefore, that the new method provides a comparatively rapid way of identifying the performance trends during the controller design process.

Figure 3 compares the predicted RMS position errors with the simulation results using the matrices \( F = [1 \ 0] \) and \( G = 0 \). Again, the predicted and the simulated performances are in agreement. Notice that at high levels of actual disturbance, i.e. in the transition region (shaded grey), the RMS position error reduces as the expected disturbance level increases. This is because the MPC behaves more like the nilpotent regulator as \( W_p \) is increased and less like the unconstrained LQR, which penalizes the control effort and therefore leads to large state deviations.
6.2 Effect of constraint settings

Figure 4 shows an example in which the constraint levels are varied. The example uses the neutrally-stable system matrix \( A_1 \). The predicted disturbance \( W_p \) is fixed at 80\% of the maximum \( W_{\text{max}} \) and the actual disturbance \( W_a \) at 80\% of \( W_p \). The constraints are

\[
|y_1(k)| \leq Y_1, \quad |y_2(k)| \leq Y_2, \quad |y_3(k)| \leq 1
\]

where \( Y_1 \) and \( Y_2 \) are variable limits on the position and velocity, respectively, whose effect is to be investigated. In the transition region, shaded grey in figure 4(a), the exact level of performance is underestimated by the prediction method. However, the predictions still capture important trends in the performance. The contour plots of the predictions (figure 4c) and the simulation data (figure 4d) are almost identical, apart from variations caused by randomness in the simulations. The "island" pattern in the top-right of figure 4(d) indicates that the performance surface is roughly flat in this region, with the small changes in simulation results dipping up and down across the contour line. The limiting values of \( Y_1 \) and \( Y_2 \) at which the constraints become significant are accurately predicted, as are the limits of feasibility.
The performance prediction accurately identifies regions of different sensitivity, as shown in figure 4(e). In region A, the performance is sensitive to the position constraint setting $Y_1$ but not to the velocity constraint setting $Y_2$. In region B, the performance is insensitive to both $Y_1$ and $Y_2$. In region C, the performance is sensitive to the velocity constraint setting $Y_2$, but not to the position constraint setting $Y_1$. So, for example, if the initial controller design settings are in the region B or C, the position control tolerance $Y_1$ can be tightened as far as the boundary with region A without incurring any penalty in control effort. These results show that the prediction method can readily be used to assess the impact of design decisions such as constraint settings.

6.3 Effect of horizon length

Figure 5 shows another trade study example, again involving the neutrally-stable system. The variables under investigation are the planning horizon $N$ and the control magnitude constraint $Y_3$, with the state constraints held constant,

$$|y_1(k)| \leq 1, \quad |y_2(k)| \leq 1, \quad |y_3(k)| \leq Y_3$$
Figure 4. Comparison of predicted performance and simulation results for a system with varying constraints.
Figure 5. Comparison of predicted performance and simulation results for a system with varying control constraint and horizon.
Again, there is some inaccuracy in the predicted performance levels in the transition region, but the contour plots show close agreement. This result demonstrates how the performance prediction method can be employed to make a methodical choice of the prediction horizon. By definition, MPC requires the designer to choose a horizon, but this is commonly achieved by judgement, backed up by extensive numerical simulation. The contour plot, accurately determined by the performance prediction method, enables regions of operation to be identified based on the sensitivity of the performance to the two design variables, shown in figure 5(e). These regions capture important trends in performance and can be used to indenitify where savings are possible. For example, in regions A and B, the performance is insensitive to the horizon length \( N \). Therefore, if the initial design settings are in either of these regions, the horizon length can be reduced as far as the edge of these regions, giving a shorter computation without incurring a performance penalty. Similarly, if the settings are in region C, where the performance is independent of control authority \( Y_1 \), the authority can be limited further, up to the boundary with region D, with no performance penalty.

### 6.4 Estimation error

The example in this section investigates the effect of estimation error on the neutrally stable example system. The disturbance and estimation error levels were bounded as follows:

\[
|e_1(k)| \leq 1, \quad |e_2(k)| \leq 1, \quad |w(k)| \leq 0.1
\]

and the state error matrix included two scaling parameters

\[
M = \begin{bmatrix} N_{\text{pos}} & 0 \\ 0 & N_{\text{vel}} \end{bmatrix}
\]

such that the position and velocity errors are bounded by \( N_{\text{pos}} \) and \( N_{\text{vel}} \), respectively. Figure 6 compares predicted control RMS with results from simulations for values of \( N_{\text{pos}} \) and \( N_{\text{vel}} \) in the range [0.01, 1]. Figure 6(e) shows the design space divided into regions of operation, based on sensitivity. In region A, the performance is independent of position noise but is sensitive to velocity noise. In region B, the performance becomes highly sensitive to both the noise levels.

### 7. Spacecraft formation flying application

This section describes the application of the new analysis method to the design of MPC for a spacecraft in a formation. Formation flying for spacecraft is an attractive technology for several forthcoming missions (Leitner et al. 2002). This approach has significant advantages over a single spacecraft, such as greater science return due to longer observation baselines, and increased flexibility. The problem of controlling spacecraft in formation is extremely challenging, requiring precise control without excessive fuel use while subject to many constraints imposed by hardware and mission requirements. Previous work (Inalhan et al. 2002) has shown that MPC is well suited to this problem, and more studies (Carpenter and Scheisser 2001, How and Tillerson 2001) have shown that sensing uncertainty is a significant driver of control design. Therefore this section uses the analysis from §5 to investigate the effect of estimation error on performance. In particular, this section considers the selection of the horizon length, replanning frequency and error box size for minimum fuel use.

The specific example of spacecraft control being examined is the relative motion in a circular orbit with the state \( x = [x \ y \ \dot{x} \ \dot{y}]^T \) and the dynamics given by Hill’s equations of motion (Hill 1878)

\[
\dot{x} = \begin{bmatrix} \frac{3n^2 x}{2} & 0 & 0 & 2n \\ 0 & \frac{3n^2 y}{2} & 0 & 2n \\ 0 & 0 & n & 0 \\ 0 & 0 & 0 & n \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_x + \begin{bmatrix} u_x \\ u_y \end{bmatrix}
\]

where \( n \) is the orbital frequency and \( u_x \) and \( u_y \) are the accelerations in the \( x \)- and \( y \)-directions, respectively. The examples in this section involve a low Earth orbit with frequency \( n = 0.001 \text{ rad/s} \).

The system is discretized assuming that the control inputs are impulsive velocity changes. The time step for discretization is equivalent to the replanning period and the results for a range of values are presented later in this section. Since, for the formation flying problem, the performance is dominated by the effects of the velocity estimation error (How and Tillerson 2001), the state error includes only velocity noises, hence \( M = [0_{2 \times 2} \ I_{2}]^T \). The controller is required to keep the spacecraft within a square error box with side lengths \( 2Y_{\text{max}} \). Part of the analysis later in the section investigates the effect of \( Y_{\text{max}} \) on fuel use. The output matrices are \( C = [I_2 \ 0_{2 \times 2}] \) and \( D = 0_{2 \times 2} \) and the constraint set is \( Y = B_{\infty}(Y_{\text{max}}) \). For the constraint tightening, a candidate controller
Figure 6. Comparison of predicted performance and simulation results for a system with varying position and velocity noise.
\(K_H\) can be found that drives to a state in the radial/in-track plane of the Hill's frame to the origin in 4 steps (which restricts the horizon length to \(N \geq 4\)). The performance metric for the analysis is the control effort \(E[u(k)\|u(k)]\).

### 7.1 MPC parameter selection

This section uses the analysis method to identify the effect of two key MPC parameters, the planning horizon length and the replanning frequency (the reciprocal of the discretization time step), on the spacecraft fuel use. Figure 7 shows predictions of control effort across a range of these two parameters. The error box size is fixed at \(Y_{\text{max}} = 5\text{m}\) and the velocity noise level at \(N_{\text{vel}} = 3\text{mm/s}\). The horizontal line marked “approximate critical plan length” roughly corresponds to the division between the unconstrained regime and the transition region. Above this line, the control effort becomes insensitive to the plan length. The line marked “approximate critical replan frequency” marks a trough in the control effort. Therefore, the globally optimal settings for the plan length and replanning frequency in terms of control effort are at the intersection of these two lines. At these settings, the analysis predicts a fuel use rate of 0.440 mm/s/orbit. A simulation with the same settings gave a result of 0.442 mm/s/orbit, indicating that the prediction is accurate.

### 7.2 Constraint trades

This section investigates the effect of velocity estimation error \(N_{\text{vel}}\) and error box size \(Y_{\text{max}}\) on control effort. Carrier phase differential global positioning (CDGPS) is a commonly examined method of sensing relative state and has been shown to produce estimates of velocity with sensing error on the order of 1 mm/s (Busse 2003). Figure 8 shows contours of constant expected control effort for the relation between error the box size and the noise level. Consider a scenario in which the noise level is known to be 3 mm/s, slightly higher than a realistic CDGPS estimate. Setting the error box size to 1 m gives an infeasible problem. If the error box size is increased to 3 m, the problem becomes feasible. Further increases give a significant decrease in fuel use, up to a size of about 7 m. Beyond this point, which lies on the dividing line between the unconstrained behavior and the transition region, the fuel use becomes insensitive to the error box size. This is a useful result as it tells us that there is no point trading the control accuracy (error box size) for control effort beyond the line of transition.

![Figure 7. Effect of plan length and replan frequency on fuel use.](image-url)
8. Conclusion

This paper has presented a method of analytically predicting the expected closed-loop performance of a system using a form of robust MPC. The analysis includes the effect of persistent disturbances and the state estimation errors. The new analysis method enables trade studies on the effect of controller parameters to be performed without extensive numerical simulation. The prediction method has been demonstrated for several example systems and is compared to simulation results. It has been shown to correctly identify the trends in performance as a function of disturbance levels and constraint settings.

The new analytical performance method has been shown to identify the performance trends as functions of controller design parameters using significantly less computation than the existing approach, namely extensive numerical simulation. A logical next step would be to embed the new prediction method within controller synthesis tools.

Acknowledgements

This work was funded in part under Cooperative Agreement NCC5-729 through the NASA GSFC Formation Flying NASA Research Announcement. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References


Gauss’s Variational Equation-Based Dynamics and Control for Formation Flying Spacecraft

Louis Breger∗ and Jonathan P. How†
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
DOI: 10.2514/1.22649

Formation flying is an enabling technology for many future space missions, and this paper presents several modeling and control extensions that would enhance the efficiency of many of these missions. In particular, a new linear time-varying form of the equations of relative motion is developed from Gauss’s variational equations. These new equations of motion are further extended to account for the effects of \( J_2 \), and the linearizing assumptions are shown to be consistent with typical formation flying scenarios. It is then shown how these models can be used to initialize general formation configurations and can be embedded in an online, optimization-based, model predictive controller. A convex linear approach for initializing fuel-optimized partially \( J_2 \) invariant orbits is developed and compared with analytic approaches. All control methods are validated using a commercial numerical propagator.

The simulation results illustrate that formation flying using this model predictive controller with \( J_2 \) modified Gauss’s variational equations requires fuel use that is comparable to using unmodified Gauss’s variational equations in simulations that do not include the \( J_2 \) effects.

I. Introduction

Formation flying of multiple spacecraft is an enabling technology for many future space science missions including enhanced stellar optical interferometers and virtual platforms for Earth observations [1,2]. Formation control objectives typically focus on controlling the relative states of the spacecraft, the dynamics of which can be captured using variants of Hill’s and Lawden’s equations for low-Earth-orbit (LEO) missions [3]. However, both of these approaches linearize the nonlinear relative spacecraft motions about a reference orbit, which is only valid for small separation distances of the satellites in the formation relative to the reference orbit radius. For larger separations, these equations of motion can no longer be used to cancel relative drift rates (initialization) or to accurately predict the effect of inputs (control) [4]. For example, the four spacecraft of the planned magnetospheric multiscale (MMS) mission [5] will be placed in a tetrahedron-shaped relative configuration with sides ranging between 10 and 1000 km, which far exceeds the separations for which Hill’s and Lawden’s models are valid for a full highly elliptic orbit (HEO) even with the correction terms introduced in [6]. Furthermore, these models do not accurately capture the effects of Earth’s oblateness, which Schaub and Alfriend [7] showed can lead to very inefficient control designs. This paper develops a new linearized modeling approach that is valid for widely spaced formations in highly elliptic orbits, accurately captures the effects of Earth’s gravity, and can be embedded in an optimization-based controller that is suitable for real-time calculations.

The relative dynamics used in this paper are based on a form of Gauss’s variational equations (GVEs) that have been modified to include the effects of \( J_2 \). GVEs are convenient for specifying and controlling widely separated formations because they are linearized about orbital elements, which are expressed in a curvilinear frame in which large rectilinear distances can be captured by small element perturbations [8]. This bypasses the linearization error created by representing the entire formation in a single rectilinear frame, which was the approach used in [3]. The use of GVE dynamics as opposed to Hill’s dynamics incurs the cost of computation associated with the use of multiple sets of time-varying equations of motion. Specifying a formation’s relative geometry in terms of differential orbital elements is an exact approach that does not degrade for large spacecraft separations. However, the advantage of using GVEs for control could be reproduced by using a separate Lawden frame for each spacecraft in the formation while still using orbital element differences to represent the formation relative geometry. Given that a nonlinear transformation and rotation is required to switch between a local-vertical–local-horizontal (LVLH) frame and orbital element differences and that GVEs are already linearized in an orbital element frame, it is both simpler and computationally more efficient to use orbital element differences to specify the formation configuration and GVEs for control.

Many formation control approaches have used GVEs for nonlinear continuous control [9–11] and also for impulsive control [12,13]. This paper introduces a control law that generally does not fire continuously and, more importantly, makes explicit its objective to minimize fuel use, which is measured in \( \Delta V \) in this paper. The control approach optimizes the effects of arbitrarily many inputs over a chosen planning horizon. Plans are regularly reoptimized, forming a closed-loop system [14]. By extending previous planning approaches [3,15] to use GVEs, we can optimize the plans for spacecraft in widely separated highly elliptic orbits. Results are presented to show that the GVE-based planning system is more fuel efficient than the four-impulse method in [12]. In addition control optimized online has the advantage of being capable of handling many types of constraints, such as limited thrust capability, sensor

Nomenclature

\[
\begin{align*}
\alpha &= \text{semimajor axis} \\
b &= \text{semiminor axis} \\
e &= \text{eccentricity} \\
h &= \text{angular momentum} \\
i &= \text{inclination} \\
M &= \text{mean motion} \\
n &= \text{orbit frequency} \\
p &= \text{semilatus rectum} \\
r &= \text{magnitude of radius vector} \\
\theta &= \text{argument of latitude} \\
\Omega &= \text{right ascension of the ascending node} \\
\omega &= \text{argument of perigee}
\end{align*}
\]

Received 23 January 2006; revision received 10 April 2006; accepted for publication 18 April 2006. Copyright © 2006 by Massachusetts Institute of Technology. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/07 $10.00 in correspondence with the CCC.

∗Research Assistant, Department of Aeronautics and Astronautics; lbreger@mit.edu.
†Associate Professor, Department of Aeronautics and Astronautics; jhow@mit.edu.
noise robustness, and error-box maintenance [3]. We also extend the virtual center approach to formation flying in [16] to GVEs and present a decentralized implementation of that algorithm.

A limitation of the orbital element approach in [15] is that it does not account for the effects of the $J_2$ disturbance, which impacted the closed-loop performance in full nonlinear simulations. This paper extends the use of the relative orbital elements in [15] to the $J_2$-modified relative state transition matrix in [17] and develops and evaluates several approaches for including the effects of thruster inputs. The resulting $J_2$-modified GVEs are used to form a set of linear parameter-varying dynamics that can be embedded in an optimization-based control system. The combination creates a controller that retains the advantages of the GVE-based controller in [15] but uses a more accurate dynamics model, thereby improving plan tracking and fuel efficiency. In particular, simulations are presented to show that the new controller in the presence of $J_2$ disturbances requires comparable levels of fuel to the approach in [15] when no $J_2$ disturbances are simulated in the model.

II. Relative Orbital Elements and Linearization

Validity

GVEs are derived in [18] and are reproduced here for reference:

$$\begin{bmatrix}
d \\
a \\
e \\
i \\
\Omega \\
o \\
M
\end{bmatrix} = \begin{bmatrix}
\begin{pmatrix}
\sin f & \cos f & 0 \\
\cos f & -\sin f & 0 \\
0 & 0 & 1
\end{pmatrix}
\end{bmatrix}
\begin{bmatrix}
u_r \\
u_0 \\
u_h
\end{bmatrix}
$$

(1)

where the state vector elements are $a, e, i, \Omega, o,$ and $M$. The other terms in the variational expression are $p, b, h, \theta, r,$ and $n$. All units are in radians except for the semimajor axis and radius (in meters), the angular momentum (in kilogram meters squared per second), the orbit frequency (in 1/seconds), and the eccentricity (dimensionless).

The input acceleration components $\mathbf{u}_r$, $\mathbf{u}_0$, and $\mathbf{u}_h$ are in the radial, in-track, and cross-track directions, respectively, of an LVLH frame centered on the satellite and have units of meters per second squared. Although the traditional Keplerian form of the orbital elements is used in this paper for conceptual clarity, later uses of transformations from [17,19] require a conversion to the nonsingular form described in those references.

The form of the GVEs can be more compactly expressed as

$$\dot{e} = A(e) + B(e)u$$

(2)

where $e$ is the state vector in Eq. (1); $B(e)$ is the input effect matrix; $u$ is the vector of thrust inputs in the radial, in-track, and cross-track directions; and $A(e) = \begin{pmatrix} 0 & 0 & 0 & 0 & \sqrt{\mu/a^3}^T \end{pmatrix}^T$, where $\mu$ is the gravitational parameter.

In a formation, the orbital element state of the $i$th satellite is denoted $e_i$. The states of the vehicles in the formation can be specified by relative orbital elements by subtracting the state of an arbitrarily chosen spacecraft in the formation, which is designated as $e_i$.

$$\delta e_i = e_i - e_d$$

(3)

For a desired orbit geometry, a set of desired relative elements, $\delta e_d$, will specify the desired state $e_d$ of each spacecraft in the formation. Approaches for choosing and coordinating the desired spacecraft states will be addressed in Secs. IV and VI.

$$e_d = e_1 + \delta e_d$$

(4)

The state error for the $i$th spacecraft in the formation, $\xi_i$, is then defined as

$$\xi_i = e_i - e_{di} = \delta e_i - \delta e_{di}$$

(5)

Note that the definition of state error given in Eq. (5) is independent of the choice of which spacecraft state is represented by $e_i$. The form of GVEs in Eq. (1) is for perturbations of orbital elements. To re-formulate these equations for perturbations of relative orbital elements [19], the GVEs for $e_i$ and $e_{di}$ are combined:

$$\dot{\xi}_i = \dot{e}_i - \dot{e}_{di} = A(e) - A(e_{di}) + B(e)u_i$$

(6)

where the term $B(e_{di})u_d$ has been excluded because thrusting does not affect the desired state of the spacecraft. The unforced dynamics can be linearized by introducing the first-order approximation [19] (the subscript $i$ is henceforth omitted for notational simplicity)

$$A(e) - A(e_d) \approx \frac{\delta A}{\delta e} \bigg|_{e_d} (e - e_d) \Rightarrow \xi = A^*(e_d)\xi$$

(7)

where the matrix $A^*(e_d)$ is all zeros except for the lower-leftmost element, which is $-3n^2/2a$, and where the sparsity of $A^*$ arises from the sparsity of the $A$ function in Eq. (2). With this approximation, the differential GVE expression in Eq. (6) can be rewritten as

$$\dot{\xi} = A^*(e_d)\xi + B(e_d)u = A^*(e_d)\xi + B(e_d + \xi)u$$

(8)

In this case the control of the relative error state, $\xi$, is nonlinear because the control effect matrix $B$ is a function of the state. Schaub and Junkins in [19] account for this nonlinearity in a continuous nonlinear control law that was shown to be asymptotically stable. The control approach developed in this section uses linearized dynamics to predict the effect of future control inputs. Linearizing the matrix $B$ in Eq. (8) yields

$$\dot{\xi} \approx A^*(e_d)\xi + \frac{\delta B}{\delta e} \bigg|_{e_d} \xi u$$

(9)

where the term $B^*(e_d)$ is a third-rank tensor and the quantity $B^*(e_d)\xi$ is a matrix with the same dimensions as $B(e_d)$. For convenience, define

$$\Delta B(e_d, \xi) = B^*(e_d)\xi$$

(10)

resulting in the new state equation

$$\dot{\xi} = A^*(e_d)\xi + (B(e_d) + \Delta B(e_d, \xi))u$$

(11)

Note that if $\Delta B$ is much smaller than $B(e_d)$, then the first-order term can safely be ignored, yielding the approximate linearized dynamics

$$\dot{\xi} = A^*(e_d)\xi + B(e_d)u$$

(12)

which can be controlled by any one of a variety of linear control techniques, including the model predictive controller discussed in Sec. IV.

The critical requirement for linear control and planning is that the term $\Delta B$ has a much smaller influence on the state dynamics than the term $B(e_d)$. However, $\Delta B$ is a linear function of the state error $\xi$, which can be arbitrarily large. The amount of acceptable error due to linearization will be a function of the mission scenario, but the linearization assumption will typically be valid only for small values of the state error. It is reasonable to expect that the values of state error will be small because the linearization is only in separation from an orbit where the dynamics linearization is valid. This section examines several example orbits that are representative of space missions that might occur in low and high Earth orbits. In
each case, the range of acceptable error is found to be large enough to accommodate expected mission performance requirements. The magnitude of the acceptable error can be computed by comparing the induced norm of the difference between the control influence matrix at its desired state, $B(e_d)$, and at the actual position of the spacecraft, $B(e)$. In Eq. (10), the first-order approximation of this term was defined as $\Delta B$. In the following examples, $\Delta B_{\text{true}}$ is defined as

$$\Delta B(e_d, \xi)_{\text{true}} = B(e) - B(e_d) = B(e_d + \xi) - B(e_d)$$

and will be calculated numerically. The cutoff point of acceptable linearization error is when the norm of $\Delta B$ exceeds some (possibly mission dependent) fraction of the norm of $B(e_d)$. To investigate this cutoff point, the following examples consider many random values of $\xi$ in the set $||\xi||_2 = r$ and calculate $\Delta B_{\text{true}}$. The $\Delta B_{\text{true}}$ with the largest 2 norm will be used to test the validity of the linearization for a given $r$. This procedure is repeated for multiple $r$ to find the largest $||\xi||_2$ for which the linearization is considered valid. Other methods of examining the linearization error of $B$ are possible, but the approach used in this paper was chosen because of its ease of implementation and consistent results for particular mission types.

A. Example: Low Earth Orbit
An example low Earth orbit is

$$e_d = \begin{pmatrix} 6.9 \times 10^6 & 0.005 & 0.610865238 & 2\pi \pi & 3.82376588 \end{pmatrix}^T$$

(14)

The matrix corresponding to $B(e_d)$ is

$$B(e_d) = \begin{pmatrix} -5.6794478 & 1808.6011 & 0 \\ -0.000082308780 & -0.00020502572 & 0 \\ 0 & 0 & 0.00010304404 \\ 0.020528419 & -0.032987976 & -0.00011890944 \\ -0.00792326 & 0.032987564 & 0 \end{pmatrix}$$

(15)

where $||B(e_d)||_2 = 1808.61$. The effect of perturbing $e_d$ for a given norm bound on $\xi$ is shown in Fig. 1. The plots show that a 1% linearization validity cutoff of 0.01, i.e., $||\Delta B(e_d, \xi)_{\text{true}}||_2 \leq 0.01||B(e_d)||_2$, can be achieved by ensuring that $||\xi||_2 \leq 8.16 \times 10^{-3}$. This bound on $\xi$ allows for orbital element perturbations that equate to rectilinear distances on the order of 25 km and velocities on the order of 40 m/s. Typical error-box sizes for LEO formation flying missions are 10–100 m in size [20], decidedly inside the linearization range of the orbit examined.

B. Example: Highly Elliptical Earth Orbit
One motivation for using GVEs as the linearized dynamics in a planer is recent interest in widely spaced highly elliptical orbits [5]. An orbit of this type is

$$e_d = \begin{pmatrix} 4.20957 \times 10^7 & 0.818181 & 0.174532925 & 2\pi & 0 \pi \end{pmatrix}^T$$

(16)

Repeating the same procedure used for the LEO case, it is determined from Fig. 2 that a 1% linearization validity cutoff can be achieved provided that $||\xi||_2 \leq 3.66 \times 10^{-3}$. In this case, the bound on $||\xi||_2$ corresponds to rectilinear distances of approximately 50 km and velocities of 2 m/s. As in the LEO case, these distances are far larger than expected error-box sizes. Because any planned trajectory would be expected to remain inside an error box at all times, the range of state errors in which the linearization is valid will not be exceeded. Unlike the LEO case, error boxes for widely separated missions, such as MMS, may be much larger than 10 m to a side, even approaching kilometers. The 1% cutoff ensures that error boxes of up to 5% of the distance between MMS satellites (1000 km during the most widely spaced phase of the mission) are acceptable [5].

Validating the linearization for additional reference orbits is a straightforward computational exercise. For example, repeating the
validation process for the LEO examples used in Sec. VI and the HEO examples used in the simulations in Sec. VII yielded valid ranges of separation that were far larger than the expected error-box sizes.

III. $J_2$-Modified Gauss's Variational Equations and Linearization Validity

Just as the GVEs in Eq. (2) express the motion of a Keplerian orbit, the equations of motion of the mean orbital element state vector $e_m$ describe the average motion of an orbit influenced by Earth oblateness effects and are given by

$$
\dot{\vec{e}}_m = \vec{A}(e_m) + \frac{\partial \vec{e}_m}{\partial \vec{u}} \vec{u}
$$

(18)

where $\vec{A}$ is explicitly a function of the mean state and implicitly a function of $J_2$, see [19]. Although Eqs. (2) and (18) appear similar, there are some important differences. In particular, Eq. (2) describes the motion of a spacecraft’s osculating orbit and is the form of the classical GVEs. Section II established that it is valid and effective to linearize the GVEs and use them for model predictive control. However, the GVEs incorporate neither the absolute nor the relative effects of $J_2$ on a satellite’s orbit. Conversely, Eq. (18) describes the motion of an orbit in a set of mean orbital elements, where the secular effects of $J_2$ are incorporated and harmonics are removed. This form of the dynamics is useful for controlling the secular drift between satellites in a formation but does not describe the physical motion and has limited applicability for missions with high precision relative state constraints. Furthermore, Eq. (18) is nonlinear in terms of the relative state, which accurately captures the system dynamics but complicates the optimization of the control inputs. The following shows that, by using the linearized propagation and rotation matrices developed in [17], a linearized form of the equations of relative motion in Eq. (18) can be derived that incorporates the oscillating effects of $J_2$, is linear parameter varying, and is valid for large spacecraft separations and reference orbit eccentricities.

The control influence matrix for mean element motion is derived using the transformation matrices between the mean and osculating motion. The following identity is used to define these transformations:

$$
\frac{\partial \vec{e}_m}{\partial \vec{u}} = \left( \frac{\partial \vec{e}_m}{\partial \vec{e}} \right) \left( \frac{\partial \vec{e}}{\partial \vec{u}} \right)
$$

(19)

From the appendix of [7], the relation between the mean orbital element state vector and the osculating orbital element state vector can be written as $e_m = f(e)$, so that

$$
\vec{e}_m = \frac{\partial f(e)}{\partial e} \vec{e} \Rightarrow \frac{\partial \vec{e}_m}{\partial \vec{u}} = \frac{\partial f(e)}{\partial e} B(e)
$$

(20)

Substituting Eq. (20) and the $B$ matrix from Eq. (2) into Eq. (19) gives

$$
\frac{\partial \vec{e}_m}{\partial \vec{u}} = \frac{\partial f(e)}{\partial e} B(e)
$$

(21)

which yields the equations of motion of the mean orbit in terms of the osculating orbital state vector $e$ (the mean elements may be considered a function of the osculating elements) and an input vector $\vec{u}$ as

$$
\vec{e}_m = \vec{A}(e_m) + \frac{\partial f(e)}{\partial e} B(e) \vec{u}
$$

(22)

The actual mean orbit $e_m$ is now defined in terms of a desired mean orbit $e_{\text{mod}}$ and a vector offset $\xi_m$

$$
e_m = e_{\text{mod}} + \xi_m
$$

(23)

Rearranging this expression and applying Eq. (18) gives

$$
\vec{e}_m - \vec{e}_{\text{mod}} = \vec{A}(e_m) - \vec{A}(e_{\text{mod}}) + \frac{\partial \vec{e}_m}{\partial \vec{u}} \vec{u}
$$

(24)

where the term $(\partial \vec{e}_{\text{mod}}/\partial \vec{u}) \vec{u}$ is omitted because the desired orbit is fixed and not subject to thrusting. Similar to the preceding section, the following linearization approximation can be made [19]:

$$
\vec{A}(e_m) - \vec{A}(e_{\text{mod}}) \approx \vec{A}^*(e_{\text{mod}}) \xi_m
$$

(25)

which is then used to find the equations of motion of the mean orbit element offset $\xi_m$,

$$
\dot{\xi}_m = \vec{A}^*(e_{\text{mod}}) \xi_m + \frac{\partial \vec{e}_m}{\partial \vec{u}} \vec{u}
$$

(26)

where the terms of the matrix function $\vec{A}^*$ are given in [21]. Equation (26) provides a linear description of the motion of the relative mean orbital elements. However, the mean orbit describes where the spacecraft is in an average sense, whereas the osculating orbits specify the actual position of spacecraft. Thus, to maximize the ability of the planner to exploit natural dynamics and operate with tight performance constraints, it is preferable to plan in terms of the osculating orbit. The approach in this paper uses a hybrid of the osculating and mean orbits to capture both the effects of $J_2$ and to plan in a way that accounts for the actual motion of the spacecraft. Having developed the relative dynamics in terms of the mean elements, we now convert to an osculating state.

Using the notation in Eq. (5), formation relative dynamics can be specified in terms of the osculating orbit $e$, an osculating desired orbit $e_d$, and an osculating orbital offset $\xi$ between them. The mean elements are expressed as functions of the osculating elements by rearranging the state error form in Eq. (23). This is used to create a relative state and a linearized rotation matrix for the transition between the mean and osculating equations of relative motion. Given that $e_m = f(e)$ and $e_{\text{mod}} = f(e_d)$, and then using Eq. (5), Eq. (23) can be rewritten as

$$
\xi_m = f(e) - f(e_d) \approx \frac{\partial f(e)}{\partial e} \Big|_{e_d} \xi
$$

(27)

by using the same linearization approach in [19]. Defining the matrix function $D$ (available in [17]),

$$
D(e_d) = \left. \frac{\partial f(e)}{\partial e} \right|_{e_d}
$$

(28)

and substituting into Eq. (21) and then into Eq. (26) yields

$$
\dot{\xi}_m = \vec{A}^*(f(e_d)) \xi_m + D(e_d) B(e_d) \vec{u}
$$

(29)

This form of the relative equations of motion is nonlinear in terms of the osculating absolute state $e$. Making the linearizing assumption (accuracy of the linear approximations are discussed later in this section)

$$
D(e_d) B(e) = D(e_d + \xi) B(e_d + \xi) \approx D(e_d) B(e_d)
$$

(30)

allows the relative equations of motion to be rewritten as

$$
\dot{\xi}_m = \vec{A}^*(f(e_d)) \xi_m + D(e_d) B(e_d) \vec{u}
$$

(31)

which has a desired osculating orbit $e_d$ and is linear in terms of the relative mean state $\xi_m$. The equations of motion in Eq. (31) are still not suited to control of the osculating relative orbit in the presence of $J_2$ because they describe the derivative of the mean state. The following section derives a form of discrete dynamics that use a relative osculating orbit as their state.

A. Extension to Discrete Time

To use Eq. (31) in an optimization-based controller of the type used in [15], it must first be discretized. Gim and Alfriend [17]
introduce the state transition matrix \( \Phi \), which is the discrete form of the continuous matrix \( A^* (f(e_d)) \), and is defined such that

\[
\zeta_m(t_1) = \Phi^* (e_{md}(t_0), t_1, t_0) \zeta_m(t_0)
\]  

(32)

where \( t_0 \) and \( t_1 \) are the times of the initial and final states, respectively, and are provided as arguments to the state vectors. The analytic definition of the matrix \( \Phi^* \) (an implicit function of \( J_2 \) and a highly nonlinear function of the mean absolute elements) is included in the appendices of [17].

The dynamics in Eq. (31) can be formulated exclusively in terms of the osculating state. Using Eq. (27), define \( D^{-1} \) as

\[
D^{-1}(e_d) = \frac{\partial e}{\partial f(e)|_{e_d}}
\]  

(33)

Substituting Eqs. (28) and (33) into Eq. (32) yields

\[
\zeta(t_1) \approx D^{-1}(e_d(t_1)) \Phi^* (e_{md}(t_0), t_1, t_0) D(e_d(t_0)) \zeta(t_0)
\]  

(34)

The analogous discrete form of the control influence matrix \( B \) on the osculating state is then given by

\[
\Gamma(e_d(t_0), t_1, t_0) = \int_{t_0}^{t_1} D^{-1}(e_d(t)) \Phi^* (e_{md}(t), t_1, t_0) D(e_d(t)) B(e_d(t)) \, dt
\]  

(35)

Thus, combining Eqs. (32) and (35) yields the discrete time equations of motion

\[
\zeta(t_1) \approx D^{-1}(e_d(t_1)) \Phi^* (e_{md}(t_0), t_1, t_0) D(e_d(t_0)) \zeta(t_0) + \Gamma(e_d(t_0), t_1, t_0) \mathbf{u}
\]  

(36)

which are the linear parameter-varying discrete equations of motion for a relative osculating orbit in the presence of \( J_2 \).

B. Validity of the Linearization Approximations

In [6] Breger showed that the approximation \( B(e_d) \approx B(e_d + \zeta) \), which is used to derive Eq. (36), is a sufficiently close approximation for levels of state error, \( \zeta \), that would normally be expected in spacecraft formation flying missions. To use Eq. (36) for linear control, it must also be shown that the linearized rotation and transition combination \( D^{-1}(e_d(t_1)) \Phi^* (e_{md}(t_0), t_1, t_0) D(e_d(t_0)) \) remains a close approximation for expected values of \( \zeta \). In [17] Gim and Alfriend showed that this matrix has low linearization error for a wide range of reference orbit eccentricities and spacecraft formation baselines in excess of 10 km. By specifying the chief orbit in [17] as the desired spacecraft state, the transition matrices then allow a maximum state error of 10 km, which is much larger than the error that would be tolerated in most proposed spacecraft formation flying missions.

C. Calculating the \( \Gamma \) Matrix

The discrete control effect matrix is defined as a matrix integral in Eq. (35). One way to calculate this matrix is by computing its derivative and numerically integrating. However, in practice this is a computationally intensive approach that may not be consistent with real-time controller implementation. A number of alternate approaches exist. This subsection discusses several of those techniques and compares their accuracy and computation times.

1. Continuous Integration

The continuous integration method for getting the discrete input matrix \( \Gamma \) from time \( t_0 \) to time \( t_1 \) is

\[
\Gamma_{\text{true}} = \int_{t_0}^{t_1} \left\{ D^{-1}(e_d(t_1)) \Phi^* (e_{md}(t_1), t_1, t_0) D(e_d(t_0)) M(e_d(t_0)) \right\} \, dt
\]  

(37)

where

\[
Mx = \delta e
\]  

(38)

and the analytic form of \( M \) can be found in [19]. The vector \( x \) is in the LVLH coordinate system and has the form

\[
x = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T
\]  

(39)

where the positions \( x, y, \) and \( z \) are in meters and the velocities \( \dot{x}, \dot{y}, \) and \( \dot{z} \) are in meters per second. This approach should use no additional linearization assumptions beyond those in [17] because the inputs and their coupling effects are incorporated continuously. Although the outputs of the integration in Eq. (37) are very accurate, the approach itself requires significant computational effort (see Sec. III.C.5).

2. Discretized Integration

An approach to integrating \( \Gamma \) that is not as computationally intensive as the numerical integration is to approximate the integral in Eq. (37) discretely. This discrete approach introduces an additional time step \( \Delta t \), which is the duration of each discrete term in the new approximate \( \Gamma \). Conceptually, this approach is treating \( \Gamma \) as a series of smaller input effects, each based on a small time-invariant assumption. The discretized integration is

\[
\Gamma_{\text{disc}} = \sum_{i=1}^{n} D^{-1}(e_d(t_1)) \Phi^* (e_{md}(t_0 + i\Delta t), t_1, t_0)
\]

\[
+ i\Delta t)D(e_d(t_0 + i\Delta t)) M(e_d(t_0 + i\Delta t)) \left[ \begin{array}{c} \frac{\Delta t^2}{2} I_3 \frac{\Delta t^2}{2} I_3 \end{array} \right]
\]  

(40)

where \( (n+1)\Delta t = t_1 - t_0 \). Here, a number of small double integrator assumptions are made, each one assuming that inputs of \( \Delta t \) seconds can safely ignore coupling effects.

3. Rectilinear Dynamics Discretization

Another approach to finding \( \Gamma \) is to use the discrete input matrix from a set of rectilinear equations of motion and to rotate it into relative orbital elements

\[
\Gamma_{\text{rect}} = M(e_d(t_0)) \Gamma_{\text{LVLH}}
\]  

(41)

where \( \Gamma_{\text{LVLH}} \) is the discrete input matrix for a set of rectilinear equations of motion (e.g., inputs to a double integrator system, Hill’s equations, or Lawden’s equations). In Sec. V, an LVLH-based version of the approximated \( \Gamma \) matrix for Lawden’s equations [22] is evaluated.

4. Gauss’s Variational Equation-Based Discretization

An alternate approach to computing \( \Gamma \) is to use the continuous GVEs by taking the matrix exponential of the continuous matrix \( A^* \) from Eq. (7)

\[
\Gamma_{\text{GVE}} = e^{A^*(t_1-t_0)} B(e_d(t_0))
\]  

(42)

where \( B \) is the GVE matrix. This approach assumes the effects of \( J_2 \) on the input matrix are negligible.

5. Validating \( \Gamma \)

The continuous integration computation of \( \Gamma \) in Eq. (37) should not require any additional validation beyond the verification that the linearization assumptions in the component matrices \( D, \Phi^* \), and \( M \) are valid. To compare the matrices calculated using Eqs. (40–42),
their norm can be divided by that of the matrix generated using Eq. (37) ($\Gamma_{\text{true}}$) to find the normalized error

$$\epsilon_{\text{disc}} = \frac{|\Gamma_{\text{true}} - \Gamma_{\text{approx}}|_2}{|\Gamma_{\text{true}}|_2} \quad (43)$$

where $\Gamma_{\text{approx}}$ is the $\Gamma$ matrix computed using one of the approximate methods. If the error $\epsilon$ is kept sufficiently low (typical cutoff might be 0.01), then the approximate method would be considered valid. Figure 3 shows the values of $\epsilon$ computed for the three approximate methods (the continuous integration method is taken as the true $\Gamma$). Time step increments are used in Fig. 3, but an alternate validation method using true anomaly would be appropriate if steps of true anomaly are being used for plan implementation [23]. In the figure, “GVEs w/o $J_2$” refers to the $\epsilon$ for $\Gamma_{\text{GVE}}$, “Discrete Approximation ($n = 50$)” refers to the $\epsilon$ for $\Gamma_{\text{disc}}$, and “Lawden LTV Method” (where LTV stands for “linear time varying”) refers to the $\epsilon$ for $\Gamma_{\text{rect}}$. The methods using Eqs. (41) and (42) are significantly more accurate than the discrete approximation. This difference can be corrected by refining the discretization time step; however, Table 1 shows that the discrete method using $n = 50$ already requires more computation time to evaluate. Hence, in the LEO example examined, the GVE- and Lawden-based approximations are both faster to compute and more accurate than the discrete approximation method for all discretization times. Although the methods in Eqs. (41) and (42) are marginally less accurate than the continuous integration, they are approximately 25 and 625 times as fast to compute, respectively.

Figure 4 shows how the evaluation of $\Gamma$ using Eq. (42) degrades as the discretization time step is increased for the highly eccentric orbit ($\epsilon \approx 0.8$) case examined in Sec. II. In the figure, $\Delta \Gamma$ refers to the difference between the $\Gamma$ calculated using Eqs. (40–42), respectively. For each time step, a series of $\Delta \Gamma$ matrices are evaluated, and the matrix with the largest induced 2 norm is used to represent the discretization error. Figure 4 indicates that the 86 s time step used in the simulations in Sec. VII is associated with just over 2% error between the $\Gamma$ matrices for the GVE-based calculation method. As the time step grows larger, the discrete approach to computing $\Gamma$ becomes marginally better than the other methods; however, it is still undesirable given that it is more than 250 times slower to compute than the method in Eq. (42).

### IV. Model Predictive Control Using Gauss’s Variational Equations

In [3], Tillerson et al. showed that given a valid set of linearized dynamics and a desired trajectory, a model predictive controller for a spacecraft formation can be designed that allows for arbitrarily many convex terminal and intermediate state conditions, as well as sensor noise robustness requirements. This controller is implemented on each spacecraft in the formation, and it is using a linear programming formulation. The general form of the optimization performed by the controller is

$$\min |U|_1 \quad \text{subject to } AU \leq b \quad (44)$$

where the matrix $A$ and the vector $b$ are formed based on the input dynamics and problem constraints and $U$ is a vector of potential control input vectors

$$U = [u_i(1)^T \quad u_i(2)^T \quad \cdots \quad u_i(n-1)^T \quad u_i(n)^T]^T \quad (45)$$

where each vector $u_i(k)^T$ is the input for spacecraft $i$ at step $k$ for an $n$ step plan.

To use the linearized GVE-based dynamics developed in Eq. (12) in the model predictive control (MPC) formulation, the dynamics can be discretized using a zero order hold assumption according to the procedure described in [24]. To use the linearized $J_2$-modified GVE-based dynamics developed in Eq. (31) in the MPC formulation, the discrete dynamics in Eq. (36) are used. Solutions to the optimization posed in Eq. (44) usually take the form of classical “bang-off-bang” optimal control laws. Figure 5 shows a typical plan to correct a small orbital element error. Note that although only two elements begin with errors, the optimized solution requires some elements to deviate from their desired states to minimize overall fuel use.

Solving the optimization in Eq. (44) with 1000 discretization steps and a terminal constraint has always required less than 0.05 s on a 3 GHz computer. Formulating the matrices used in the optimization has always taken under 10 s, far less than the 86 s discretization time step. The time required to formulate the problem will increase as the discretization step is made smaller and additional constraints are added. Although the computation numbers are very small, a more complicated formulation could still be implemented in a real-time system by specifying that thrusting not begin for several time steps into the plan. This will result in a plan that does not require action.

### Table 1 Average durations (in seconds) required to compute $\Gamma$ matrices using various methods

<table>
<thead>
<tr>
<th>Calculation method</th>
<th>Average computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma_{\text{true}}$</td>
<td>1.02 s</td>
</tr>
<tr>
<td>$\Gamma_{\text{disc}}$ ($n = 50$)</td>
<td>0.42 s</td>
</tr>
<tr>
<td>$\Gamma_{\text{rect}}$</td>
<td>0.041 s</td>
</tr>
<tr>
<td>$\Gamma_{\text{GVE}}$</td>
<td>0.0016 s</td>
</tr>
</tbody>
</table>
until some specified time in the future when it is certain that the formulation and optimization will have been completed.

Figure 6 shows the error between a planned trajectory using the HEO example and the actual implemented trajectory when a trajectory is implemented without replanning (i.e., open loop). The majority of the orbit has near-perfect tracking, with error entering only near perigee, when nonlinearity effects are most significant. Note that the maximum trajectories following errors for each element in Fig. 6 are approximately $[7 \times 10^{-9}, 8 \times 10^{-10}, 1 \times 10^{-9}, 1 \times 10^{-8}, 2 \times 10^{-8}]$. This compares favorably with the maximums for the same trajectory following error using a controller based on the equations in Sec. II with no $J_2$ effects taken into account: $[1 \times 10^{-8}, 1.5 \times 10^{-8}, 4 \times 10^{-8}, 8 \times 10^{-8}, 1 \times 10^{-7}]$. The maximum norm of the trajectory at any time is $\|\mathbf{e}\|_2 \approx 5 \times 10^{-5}$, which is significantly below $3.66 \times 10^{-3}$, the maximum norm of acceptable linearization error for this orbit, which was determined in Sec. II. Thus, it is valid to use the same linearized dynamics and controller to create a new plan from this terminal position. Repeatedly implementing new plans from a given initial error within the valid linearization range has always yielded terminal errors that were smaller and, hence, valid as initial conditions for replanning.

A. Error-Box Constraints Using Relative Orbital Elements

Several approaches have been developed to specify formation flying mission performance constraints. Generally, the goal of formation flying control is to keep the formation from “drifting apart” and to maintain some relative geometry. This requirement has been translated into maintaining orbits that have the same period and specifying desired relative states for spacecraft to follow. Both goals can be accomplished simultaneously by specifying relative desired points that have identical periodicity. Then to ensure that the spacecraft do not drift and that the formation geometry is maintained adequately, the control objective is to keep the spacecraft within some region around its desired point. This region is defined as a dead-band in [25] and similarly as an error box in [3].

Maintaining a spacecraft within an error box has several advantages over tracking a desired point: it does not require fuel to be used to correct minor deviations from the desired orbit, it better captures mission constraints that typically require only satellites to be in desired positions within some acceptable error, and it allows “breathing room” for the controller to account for modeling errors. In addition, the method of planning based on GVEs proposed in Sec. IV relies on the validity of the linearization analyzed in Sec. III, which degrades as the difference between the actual orbital element state and the orbital element state that has been linearized increases. If the error box used for a particular mission is smaller than the linearity range, which it typically would be (see Sec. II), the constraint that the spacecraft remain in the box provides an additional means of verifying that the linearity assumptions will be satisfied.

Several approaches can be taken to create an error box. Position error boxes are demonstrated in [3], which is a convenient bounding mechanism for a formation flying mission because it coincides well with science requirements on the accuracy of the formation geometry shape. When the formation geometry is specified in orbital elements, it is most convenient to use a six-dimensional error box with bounds on each of the state elements. This approach, while simple and convenient for enforcing acceptable relative drift levels, does not map well into the position error-box constraints typical of previous performance specifications. To transition between LVLH error states, $x$, and relative orbital element error states, $\xi$, a first-order rotation matrix $M(e_d)$ is used [see Eq. (38)]. It is possible to enforce relative position and relative velocity error-box constraints using the $M(e_d)$ matrix by formulating the optimization problem in Eq. (44) with constraints at every step $k$ where it is desired that the spacecraft remain inside an error box

$$x_{\text{min}} \leq M^{-1}(e_d)\xi \leq x_{\text{max}} \quad (46)$$

where $x_{\text{max}}$ and $x_{\text{min}}$ denote opposing corners of the error box. To exclusively enforce a partial state error box (e.g., a position box), $M^{-1}(e_d)$ can be premultiplied by an additional matrix $H$ in Eq. (46) to retain only the desired components of the state.
B. Formation Flying: Coordination Using Gauss’s Variational Equations

The model predictive controller described in Sec. IV is designed to be decentralized, with a fully independent controller being run on each spacecraft. The controller designs trajectories that will keep a spacecraft $i$ inside an error box centered about the spacecraft’s desired orbit, $e_{0,i}$. In Sec. II, the desired orbits are defined with respect to the actual orbit of an arbitrary satellite in the formation, $e_1$, using differential orbital element vectors, $\delta e_{d,i}$, in the same manner used in [19]. In a system where initial conditions are chosen infrequently, it may be desirable to introduce additional coordination into the formation. When spacecraft each track desired states with no coordination, the control task is referred to as formation keeping [3]. Alternately, formation flying occurs when the spacecraft controllers collaborate to achieve formation-wide fuel minimization. This coordination can be achieved by calculating a central point that minimizes the overall weighted state error of each spacecraft in the formation. Approaches to implementing closed-loop coordination of this type are presented in [16,26]. The virtual center approach in [16] is a centralized calculation of the error-minimizing center based on fuel weighting and derived from measurements available through carrier-phase differential global positioning system (CDGPS) relative navigation of the type described in [27]. An equivalent approach can be used to find an error-minimizing reference orbit for a formation described in differential orbital elements.

Measurements from a CDGPS relative navigation system are assumed to be in the form of relative LVLH states [27], $x_i$ [see Eq. (39)], for each satellite in the formation. The measurements will be relative to an arbitrary absolute satellite state, $e_1$, in the formation, which is assumed to be at the origin of the LVLH frame. In addition to relative states, the global positioning system (GPS) sensors on each satellite can be expected to compute a less accurate estimate of the spacecraft’s absolute state. Given an estimate of the absolute state in Earth centered inertial (ECI) coordinates, $X_{ECI}$, and the relative states $x_i$, the differential states $\delta e_i$ in Eq. (3) can be computed in several ways. The matrix $M(e_1)$ in Eq. (38) could be computed and used to create a first-order approximation of the relative differential element states. However, an exact conversion can be calculated by forming estimates of the absolute states of each of the satellites based on their relative measurements

$$X_{ECI} = X_{ECI} + x_i$$

(47)

The absolute states $X_{ECI}$ can be converted to Keplerian orbital elements, $e_i$, of each satellite using a well-known procedure described in [21]. The relative measurements are then recovered in terms of differential orbital elements, $\delta e_i$, using Eq. (3). Desired differential elements, $\delta e_{d,i}$, are then specified with respect to an unknown virtual center state, $\delta e_{v}$, which is specified with respect to the absolute state $e_1$. The error of spacecraft $i$ with respect to the virtual center, $\xi_{vi}$, is given by [16]

$$e_i - \delta e_{d,i} - \delta e_v = \xi_{vi}$$

(48)

which can be placed in the standard least-squares form

$$b_i - C_i \delta e_v = \xi_{vi}$$

where $b_i = e_i - \delta e_{d,i}$, $C_i$ is a 6 $\times$ 6 identity matrix, and $\delta e_{v}$ denotes the location of the virtual center with respect to $e_1$ in differential orbital elements. By concatenating the $b_i$, $C_i$, and $\xi_{vi}$ vectors for each spacecraft, the statement of error for the entire formation is written $b - C \delta e_v = \xi$, where $b = [b_1^T, \ldots, b_N^T]^T$, $C = [C_1, \ldots, C_N]^T$, and $\xi = [\xi_{v1}, \ldots, \xi_{vN}]^T$. The solution that minimizes the error vectors globally in a weighted least-squares sense is

$$\delta e_v = (C^TWC)^{-1}C^Twb$$

(49)

where $W$ is a weighting matrix that can be used to bias the center location according to the fuel-use rates of different satellites in the formation, as well as to weight orbital elements individually based upon the amount of control required to alter them (obtainable from the GVEs for $e_1$). This calculation can be decentralized and reduces the following iterative form [15]

$$\delta e_{v} = b_1, \quad \delta e_{v} = \delta e_{v-1} + \frac{w_i}{\bar{w}_{i-1} + w_i} (b_i - \delta e_{v-1})$$

(50)

where $w_i$ is the weight of the $i$th estimate and $\bar{w}_i = w_1 + w_2 + \cdots + w_i$. In this formulation, a spacecraft $i$ must pass its current state estimate, $\delta e_{v,i}$, and the scalar $\bar{w}_i$ to the next spacecraft for a new estimate of the optimal center position to be formed. Using this method, the error-minimizing fuel-weighted virtual center can be known in one full cycle around a formation.

V. Comparison to Other Gauss’s Variational Equation-Based Impulsive Control Schemes

The optimized controller developed in Sec. IV can be applied to a range of spacecraft control problems. This section uses that controller for the specific problem of correcting state error over a finite horizon to compare its performance and capabilities with other methods. GVEs have been used to design many Lyapunov and fixed impulse control systems [9,10,12,13,18]. Several research groups have proposed control laws for formation flying spacecraft that use GVEs to design impulsive thrusting maneuvers for orbit correction. A method of producing optimized impulsive plans for very low eccentricity orbits is presented in [28], but this approach does not extend to the higher eccentricities required for MMS missions. Another approach to using GVEs for formation control is to derive a continuous proportional-derivative controller satisfying a Lyapunov equation [9–11,19]. Control algorithms of this type have been shown to be asymptotically stable in most cases [11] but belong to a class of control systems that fire continuously. Continuous firing is generally not desirable for space missions because it is often disruptive to the science mission, it typically must be coupled with attitude maneuvers, and it expends fuel (nonreplenishable aboard a spacecraft) continuously.

The method of formation control in [13] is based on GVEs and uses a single corrective thrust computed using a nonlinear optimization. Although this method is guaranteed to find the optimal single-thrust correction for an arbitrary time period, it is not guaranteed (or likely) to find the optimal multiple-thrust correction. In addition, this approach is restricted to use in low Earth orbits and is only designed to correct errors in the semimajor axis, eccentricity, and inclination. An approach presented in [29] uses a pseudoinverse to the GVE control effect matrix to calculate a single corrective impulse. This approach is not guaranteed to be fuel optimal for any case and is not accurate for correcting position errors.

Schaub and Alfriend [12] describe a controller that uses four impulses over the course of an orbit to correct arbitrary orbital element perturbations. Because of its more general applicability, this section will compare that approach to the MPC controller presented in Sec. IV. Both methods are designed to drive the elements of a state error $\xi$ to zero over a fixed time interval. The four-impulse approach has not been presented in the context of performance criteria (e.g., trajectory or terminal error boxes, robustness to disturbances) or constraints (e.g., maximum thrust level), and so the comparisons in this section will use an MPC controller formulation that minimizes fuel use while driving the error state to zero in a fixed time and has no other constraints. In addition, for the purposes of comparison, no J2 effects are used in either planning approach.

The algorithm in [12] can be summarized in four steps to be taken over the course of an orbit. When the argument of latitude is 0 or $\pi$ radians, implement a velocity change (impulsive thrust), $\Delta v_{b} = [h/(r \cos \theta)] \Delta \xi$, in the cross-track direction of an LVLH frame centered on the spacecraft to cancel the inclination error component of $\xi$. When the argument of latitude is $\pi/2$ radians, implement a velocity change, $\Delta v_{b} = [h \sin \iota/(r \sin \theta)] \Delta \Omega$ in the cross-track direction to cancel the ascending node error. At perigee and apogee, implement $\Delta v_{c}$ and $\Delta v_{c}$, respectively, in the radial direction to cancel the argument of perigee and mean anomaly errors

$$\Delta v_{c} = -\frac{n a}{4} \frac{(1 + e)^2}{\eta} \left[(\omega + \Omega) \cos i + \Delta M\right]$$

(51)
\[ \Delta v_{th} = \frac{na}{4} \left[ \frac{(1-e)^2}{\eta} \left( \Delta a + \Delta \Omega \cos i + \Delta M \right) \right] \]  
\( (52) \)

Also at perigee implement \( \Delta v_{th} \), and at apogee implement \( \Delta v_{th} \) in the in-track direction, to cancel the semimajor axis and eccentricity errors

\[ \Delta v_{th} = \frac{na}{4} \left( \frac{\Delta a}{a} + \frac{\Delta e}{1+e} \right) \]  
\( (53) \)

\[ \Delta v_{th} = \frac{na}{4} \left( \frac{\Delta a}{a} - \frac{\Delta e}{1-e} \right) \]  
\( (54) \)

Using the notation and the HEO reference orbit from Sec. II, the following example compares the MPC method with the control approach reviewed in this section. Note that in comparison to the MPC approach, the four-impulse method is very simple to implement. However, the two approaches have different rates of fuel use for identical tasks. For the state error

\[ \zeta = \begin{bmatrix} 10^{-9} & 10^{-7} & 10^{-7} & 10^{-7} & 10^{-7} \end{bmatrix}^T \]  
\( (55) \)

the four-impulse method requires 1.42 mm/s of fuel to correct the state error over the course of an orbit and the MPC method requires 0.549 mm/s of fuel. In this example, the model predictive controller was given a full orbit time horizon. However, the same control objective could have been achieved in less time, but using more fuel. A series of 1000 orbital element state error vectors, \( \zeta \), was generated in which each perturbed element was a random number between \( \pm 10^{-6} \). For each of the error vectors, both control methods were used to generate plans for eliminating the error. On average, the MPC maneuvers required only 51% of the fuel required by the four-impulse maneuver. Further controller comparisons are presented in Sec. VII.

VI. Fuel-Optimized Semi-\( J_2 \)-Invariant Initial Conditions

Section IV presented a model predictive controller that can be used to create optimized plans for relative orbit control in the presence of \( J_2 \) disturbances. In a spacecraft formation, it is critically important both to conserve fuel when maneuvering and to maneuver to a state that will, over time, conserve fuel. The latter is an initial-condition (IC) problem, the specifications of which will depend on the unique requirements of a particular mission. However, in any spacecraft formation, a primary goal will be to prevent the vehicles from drifting apart, because that will typically end the mission. If the spacecraft in a formation tend to drift apart, then periodic maintenance maneuvers will be required to restore the formation. Initial conditions are called invariant if they eliminate drift, thereby allowing the spacecraft to maintain their relative orbits without expending fuel. In the context of this paper, invariance does not necessarily imply any form of relative boundedness at other points along the reference orbit, but this could be addressed by including additional error-box constraints to the optimization developed in this section.

For purely Keplerian orbits, invariance translates into a requirement that all spacecraft in a formation have the same semimajor axis. For example, this requirement was solved for analytically using Lawden’s equations of motion in [30]. However, Earth oblateness effects (\( J_2 \)) make orbits based on the Keplerian invariance solution drift apart. In fact, when the effects of \( J_2 \) are considered, very few perfectly invariant orbits exist. Hence, it is more common for a \( J_2 \) “invariant” orbit to instead be truly invariant only in several dimensions where it is possible to cancel the relative effects of \( J_2 \). Analytic conditions based on this partial invariance have been introduced [7,31]. The following presents an alternate approach that uses the dynamics in Sec. II and [17] in a convex linear optimization to find initial conditions that balance the objective of not drifting in the presence of relative \( J_2 \) effects against the objectives of minimizing the fuel use required to achieve these initial conditions and retaining a specified geometry for the formation.

To begin, specify that orbits are invariant if their relative orbital offset, \( \delta \), in Eq. (5) remains unchanged over a period of time so that \( \delta(e(t)) = \delta(e(t_0)) \), where \( t_0 \) is the duration of interest (typically an integer number of orbits). Then, using the state transition matrix from Eq. (34) gives the constraint

\[ \delta(e(t_1)) = \delta(e(t_2)) = D^{-1}(e(t_2))\tilde{\Phi}^n(e(t_1), t_0)J(e(t_1))\delta(e(t_1)) \]  
\( (56) \)

Defining the matrix function \( \tilde{\Phi}_{dp} = D^{-1}(e(t_{k+1}))\tilde{\Phi}^n(e(t_1), t_{k+1})J(e(t_1)) \) gives the invariance condition

\[ \delta(e(t_1)) = \Phi_{dp} \delta(e(t_1)) \rightarrow (\Phi_{dp} - I)\delta(e(t_1)) = 0 \]  
\( (57) \)

where \( I \) is a \( 6 \times 6 \) identity matrix. As mentioned above, the resulting geometry of the no-drift (complete invariance) condition is too restrictive for many missions, but partially invariant conditions can be obtained by minimizing the weighted norm of the invariance condition

\[ \min_{\delta(e(t))} \|W_{\Phi}(\Phi_{dp} - I)\delta(e(t_1))\| \]  
\( (58) \)

where the weighting matrix \( W_\Phi \) is introduced to extract states of interest to penalize particular types of drift. Note that if \( W_\Phi \) is the matrix \( M(e(t)) \) [see Eq. (37)], which rotates the differential osculating elements \( \delta e \) into an LVLH frame, then the elements of the LVLH state can be directly penalized (e.g., extracting only position states could penalize meters of drift). This enables the drift formulation to penalize the distance from the desired geometry in a Cartesian frame, as opposed to just using orbital elements. Penalizing true separation distance finds initial conditions that will maintain the formation shape, an important consideration for missions that require specific geometric configurations [5].

The overall problem statement then is, given a spacecraft at offset \( \delta e(t_{k+1}) \), design a control input sequence \( U(t), t \in [t_k, t_{k+1}] \), that generates a set of initial conditions at \( t_0 \) that balances the tradeoff between the ensuing drift by time \( t_1 \), the fuel cost of achieving these initial conditions, and the extent to which the formation geometry is maintained. The semi-invariant initial-condition optimization cost function is

\[ C^* = \min_U \{Q_u \|W_{\Phi}(\Phi_{dp} - I)\delta(e(t_0)) + \Gamma U\| + Q_e \|W_{\Phi}^n\delta(e(t_0)) + \hat{\Gamma} U\| \} \]  
\( (59) \)

where \( C^* \) is the optimal cost, \( W_\Phi \) is a weighting matrix to specify the type of geometry penalty, \( Q_u \) is a weighting on fuel minimization, \( Q_e \) is a weighting on desired formation geometry, and \( Q_d \) is a weighting on drift. Using Eq. (56), \( \delta(e(t_1)) = \Phi_{dp} \delta(e(t_{k+1})) + \hat{\Gamma} U \), where \( \hat{\Gamma} \) is a row of convolved \( \Gamma \) matrices [see Eq. (35)] that propagate the effects of a vector of inputs \( U \) [see Eq. (45)] at each time step of the maneuver [3].

\[ \hat{\Gamma} = \begin{bmatrix} \Phi(n,1) \hat{\Gamma}(0) & \Phi(n,2) \hat{\Gamma}(1) & \cdots & \Phi(n,n-1) \hat{\Gamma}(n-2) & \Phi(n,n-1) \hat{\Gamma}(n-1) \end{bmatrix} \]  
\( (60) \)

where \( \Phi(k,j) = D^{-1}(e(j_{t_k}))(\tilde{\Phi}^n(e(j_{t_k}), j_{t_k})J(e(j_{t_k}))) \), \( \hat{\Gamma}(k) = \Gamma(e, (k+1) j_{t_k}, t_{k+1}) \), and \( t_j \) is the discretization time step. The cost function uses the initial state of each spacecraft in the formation as the desired geometry, and so the geometry weighting penalizes deviations from the open-loop state propagation. Note that a simple modification to the cost function could separate the initial geometry from the desired geometry. The optimization in Eq. (59) can be easily implemented as a linear program if 1 norms are used, permitting efficient fast online solutions [32]. As expected, a sufficiently high weighting on invariance results in a minimizing control input \( U^* \), where \( \hat{\Gamma} U^* = -\Phi_{dp} \delta(e(t_0)) \). Alternately, a sufficiently high \( Q_e \) (with
an identity matrix for $W_0$ results in $\hat{\Gamma} U^* = [0 0 0 0 0 0]^T$ because the inputs will all be zero to maintain the original geometry. Figure 7 shows drift rates and fuel costs for a series of initial conditions generated by the optimization method with a half-orbit planning horizon as $Q_d$ is changed. For this example, the 1-norm is used to penalize both drift and fuel use, and both $W_d$ and $W_f$ are set to the position rows of $M$ to penalize only position drift and geometry separation. With a very low $Q_d$, $Q_f$ will dominate, resulting in no control use. The zero drift point corresponds to high fuel use because it necessitates driving the spacecraft to nearly the same orbits. A range of possible optimized initial conditions lie in between those extrema. The high-drift unmodified initial conditions lie very close to the vertical drift axis but drop to just over 0.5 m of drift with a minimum of fuel use. Further drift reductions are possible, but at greater fuel cost. It is readily apparent from the graph that using additional $\Delta V$ will produce diminishing returns in terms of reducing drift.

Initial conditions generated by other $J_2$-invariant conditions should lie either on or above the optimized result. The ■ in Fig. 7 represents the initial condition based on the $J_2$-invariance condition in case 1 of [31], which requires no mean period drift. The point is nearly optimal for this example, but this is not guaranteed to be the case for other problems. The ◆ indicates the drift that occurs when using semi-invariant initial conditions that allow perigee drift [7]. In both specific cases, the partial invariance conditions allow for a range of possible initial conditions. Both analytic cases occur near the same drift levels as the initial conditions found by the optimizing approach. This indicates the optimized ICs may, in those cases, be meeting the same invariance criteria, while simultaneously finding ICs that minimize fuel use. The optimization-based approach enables the identification of a range of fuel-optimized initial conditions that can be used to better meet the requirements of a specific mission.

Figure 8 shows a number of optimizations of the same orbit and desired offset; however, in this case, both $Q_f$ and $Q_d$ are made significant whereas $Q_u$ is varied. The figure shows the cost associated with changing the formation geometry ($\| [W, \hat{U} U]|^T \|$) versus the fuel cost ($\|U\|^T$). When $Q_u$ is very high relative to $Q_d$, the optimized ICs, $\delta \epsilon(t_c)$, are equivalent to the open-loop propagation of $\delta \epsilon(t_0)$ (i.e., no fuel is used). This corresponds to 12 m of drift over an orbit. As $Q_d$ is increased, the optimized ICs are farther from $\Phi_{\delta \epsilon} \delta \epsilon(t_0)$, but the resulting drift is lower because $Q_d$ has a greater effect on the solution. The solution that achieves 0.25 m of drift with almost no geometry cost represents a compromise between the desired formation geometry and the drift resulting from the effects of relative $J_2$. At the cost of slightly repositioning the formation (velocity changes are not penalized in the $W_d$ used for this example), the drift over an orbit has been reduced by over 4 m. When invariance dominates (the lower-right corner of the figure), the optimized initial conditions cancel almost all of the orbital offset $\delta \epsilon$, indicating that geometry goals have been ignored.

VII. Formation Maintenance on a Highly Elliptic Mission

The control system described in Sec. IV was demonstrated on a segment of a mission similar to MMS, where four spacecraft create a regular tetrahedron geometry once per orbit to perform science observations. The orbits of the four spacecraft are widely separated and highly elliptical, presenting a challenge for many optimal formation specification and control approaches in the literature [23,30]. Using the tetrahedron initial-condition optimization approach in [15] and the model predictive approach in Sec. IV, the four spacecraft were controlled in a fully nonlinear simulation with Earth oblateness effects, atmospheric drag effects, and other realistic disturbances using a commercial orbit propagator [33]. The control objective in this simulation is to achieve a set of tetrahedron initial conditions once per day near the formation orbit apogee. The reference orbit for the formation is the highly eccentric orbit used in Sec. II B.

To implement the MPC scheme in Sec. IV using the dynamics developed in Sec. III, the approximate recitilinear method of calculating $\Gamma$ is used. The dynamics matrices $\Gamma$, $\Phi$, and $D$ are all functions of the desired orbital elements, which are parameter varying. To obtain accurate trajectories for the absolute desired orbital elements for the formation, they are integrated numerically with $J_2$ disturbance effects included and then used to generate the linear propagation matrices used in the optimization. The time step used in the simulation is 86 s, providing approximately 1000 discretization points in each day-long orbit. Constant time step duration was chosen to approximate typical flight computer operation, but plans with varying time steps can also be designed by planning in increments of true anomaly. Tillerson and How [23] discuss how these can be implemented. The planning horizon length for this simulation is one full day.

Figure 9 shows the rate at which fuel was used over the course of 1 week of formation flying. The formation fuel-use rate converges to approximately 12 mm/s per day ($\approx 1$ orbit) for each satellite. Note that all previous simulations had to be limited to the case where the $J_2$ effects were disabled to ensure that the formation remained stable. In that case, for the same configuration, but without the effects of $J_2$ included in the controller dynamics or the simulation dynamics, the results showed an average $\Delta V$ of 11.5 mm/s per satellite per orbit. These nearly equivalent fuel-use rates for simulations with and without $J_2$ indicate that linearized modeling of $J_2$ effects in a controller of the type presented in Sec. IV is sufficient to prevent the disturbances from dominating fuel usage. The state error for one of the spacecraft in the formation is seen being driven to the origin in Fig. 10. Trajectories followed during this simulation fall within the range of acceptable state error determined for the linearizing assumptions used in Sec. III.
A variant of GVEs that incorporates the effects of $J_2$ was used to derive a set of linearized relative dynamics of orbital motion and extend previous work on planning-based controllers. This choice of a linear parameter-varying (LPV) dynamics model to design the controller allows a compromise between simple, but inaccurate, linear models (e.g., Hill’s equations) and high fidelity, but often difficult to control, nonlinear models. In particular, by accounting for $J_2$ disturbances in the dynamics, the planning controller can exploit these dynamics for improved fuel efficiency. The linearization assumptions used in the approach were shown to be valid for typical spacecraft error-box sizes. The LPV model was used in a model predictive controller, and the combination was shown to be more fuel efficient than a previously published technique. The overall control system ($J_2$-modified GVE-based dynamics embedded in the model predictive controller) was also used to specify and control a large (1000 km sides) tetrahedron-shaped formation in an MMS-like orbit for a period of 20 days using a commercial propagator with realistic disturbances. The results showed that the controller is reliable and that formation flying using this MPC with $J_2$-modified GVEs requires fuel use that is comparable to using unmodified GVEs in simulations that do not include the $J_2$ effects.

**Acknowledgments**

This work was funded under Cooperative Agreement NCCS-729 through the Goddard Space Flight Center Formation Flying NASA Research Announcement. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration. The authors would like to thank Terry Alfriend for his ideas and suggestions.

**References**


Differential Semimajor Axis Estimation Performance Using Carrier-Phase Differential Global Positioning System Measurements

Jonathan P. How,∗ Louis S. Breger,† and Megan Mitchell‡

Massachusetts Institute of Technology, Cambridge, Massachusetts 02139-4307

Kyle T. Alfriend§

Texas A&M University, College Station, Texas 77843

and

Russell Carpenter∥

NASA Goddard Space Flight Center, Greenbelt, Maryland 20771

DOI: 10.2514/1.23083

This paper investigates the impact of navigation errors on the navigation and control aspects of formation flying spacecraft. The use of carrier-phase differential Global Positioning System measurements in relative navigation filters is analyzed, with a particular focus on the semimajor axis error. Semimajor axis error is shown to be the sum of two positive quantities that are related to the satisfaction of the balance and correlation requirements. Previous publications have suggested that a “good” navigation filter would yield estimates of the along-track velocity and radial position that are strongly correlated. However, practical experience with filters based on the carrier-phase differential Global Positioning System measurements has shown that this seldom occurs, even when the estimation accuracies are very good. Analytical methods and numerical simulations are used to show that the optimal semimajor axis estimate from a Kalman filter does not satisfy these requirements for any combination of measurement and process noise. Numerical examples with a fully nonlinear extended Kalman filter appear to bear out these conclusions. The combination of these simulations and analysis provides new insights on the crucial role of the process noise in determining semimajor axis knowledge.

I. Introduction

In any space mission, accurate orbit predictions are often at least as important as accurate solutions at epoch. Accurate predictions are required for such purposes as acquisition, pointing, conjunction analysis, and maneuver planning. For problems such as rendezvous, formation flying, constellations, and debris avoidance, accurate predictions of the relative orbital motion may be even more significant. Because linearized two-body motion, which is unstable along the orbit track, is the dominant dynamic mode, the most significant means to achieve accurate prediction is to control absolute and/or relative along-track error growth. Minimizing along-track error growth for a Keplerian system (i.e., no atmospheric disturbances or Earth oblateness effects) requires accurate absolute and/or relative knowledge of the orbital energy [equivalently, period, semimajor axis (SMA), or mean motion], as well as accurate knowledge of the ratio of the radial and along-track components of orbital speed (equivalently, flight-path angle). These requirements constrain the ratios and, hence, correlations between errors in along-track position and radial velocity, as well as between errors in radial position and along-track velocity.

These relationships were recognized and documented for near-circular orbits at least as far back as the time of Apollo 8 [1] and were specified in terms of the components of the navigation error covariance matrix at least as early as 1987 [2]; the cited works, though publicly available, were not widely known, although the relationships themselves appear to have been widely known as industry folklore. Some of the first documented successful Global Positioning System (GPS) spaceflight experiments were found to not fully satisfy these constraints [3, and subsequent analysis of precise relative GPS navigation in the context of formation flying has demonstrated that increasing the GPS accuracy leads to similar conclusions [4,5]. This work presents simple analytical experiments to demonstrate that the Kalman filter does not satisfy the previously mentioned relationships, which are referred to here as the correlation and balance properties. In particular, it is shown that no combination of measurement noise and process noise in a simplified Kalman filter can achieve solutions that have previously been understood to generate accurate orbit predictions. Numerical examples with a fully nonlinear extended Kalman filter (EKF) appear to bear out these conclusions. This conclusion is reminiscent of experiences that the author of [6] claimed as his motivation for deriving a nonwhite process noise model based on gravity errors for sequential orbit determination. Reference [7] makes similar conclusions concerning physically connected bias models for drag. One contribution of the present work is to present a solid foundation for such empirical observations.

II. Background

In formation flying missions, accurate knowledge of the difference in semimajor axes or, equivalently, the difference in orbital energy between the vehicles in a formation is important [3–5,8]. A difference in semimajor axes means that the two vehicles have different orbital periods and thus they will drift out of formation...
unless regular control effort is applied [9]. The output of a carrier-phase differential GPS (CDGPS) Kalman filter includes the relative formation state in a local vertical horizontal (LVLH) reference frame, where the radial, in-track, and cross-track directions define the x, y, and z axes. Understanding the relationship between position and velocity accuracies and semimajor axis accuracy is key to evaluating the output of this type of filter.

Although [4] develops the navigation error analysis from absolute state relations, the results can be reformulated for the relative case. The relative navigation error equations, shown next, relate semimajor axis error to position and velocity errors. Note that this discussion is limited to circular reference orbits. The semimajor axis $a$ of vehicle $i$ is

$$a_i = \frac{2}{r_i - \sqrt{v_i^2}} \mu$$

(1)

where $r$ and $v$ are the radius and speed in the Earth-centered inertial (ECI) reference frame, and $\mu$ is the gravitational constant of the Earth. Equation (1) is used to find the difference in semimajor axes of vehicles $i$ and $j$:

$$\Delta a_{ij} \approx 2(r_j - r_i) + (2/n)(v_j - v_i)$$

(2)

where one vehicle is assumed to be in a near-circular orbit and the other within close proximity.

The force-free solution to the HCW equations is

$$\dot{x} = r_0 \sin nt + \frac{2\gamma_0}{n} \cos nt$$

$$\dot{y} = \frac{2\gamma_0}{n} \cos nt + \frac{4\gamma_0}{n} \sin nt$$

$$\dot{z} = \frac{6\gamma_0}{n} nt$$

(3)

The force-free solution to the HCW equations is

$$x(t) = \frac{\dot{x}_0}{n} \sin nt - \left(\frac{2\gamma_0}{n} + 3x_0\right) \cos nt + \left(\frac{2\gamma_0}{n} + 4x_0\right)$$

$$y(t) = \frac{2\gamma_0}{n} \cos nt + \left(\frac{4\gamma_0}{n} + 6x_0\right) \sin nt + \left(\gamma_0 + \frac{2\gamma_0}{n}\right)$$

$$z(t) = \frac{6\gamma_0}{n} nt$$

(4)

Compensating for the Coriolis effect of differentiating in the rotating HCW reference frame, the difference in semimajor axes (the $ij$ subscript is subsequently omitted) can be approximately written in terms of the relative radial position $x$ and the in-track velocity $\dot{y}$, as

$$\Delta a \approx 2[x + (\dot{y}/n)]$$

(5)

This expression for the differential semimajor axis applies throughout the orbit, but using Eq. (4), it can be rewritten as

$$\Delta a \approx -3(\dot{y} + 6n\gamma_0)(-2/3n) = -(3\gamma_0 + 6n\gamma_0)(-2/3n)$$

(6)

which shows that it is directly related to the secular drift term in the in-track solution to the HCW equations in Eq. (4). Clearly, if the difference in semimajor axes is zero, then there will be no secular drift between the spacecraft. This is the linearized form of the more general energy-matching condition, which is that spacecraft with equivalent semimajor axes will not drift apart. The standard deviation of the approximate differential semimajor axis estimate, $\sigma_{\Delta a}$, is given by [4]

$$\sigma_{\Delta a} = 2\sqrt{4\sigma_x^2 + \left(\frac{4}{n}\right)\rho_{x\gamma}\sigma_x\sigma_y + (1/n^2)\sigma_y^2}$$

(7)

The parameters $\sigma_x$, $\sigma_y$, and $\rho_{x\gamma}$ are derived from the error covariance matrix for the relative LVLH state estimate:

$$\mathbf{\hat{x}} = [x \ y \ \dot{y}]^T$$

with estimation error $\mathbf{\hat{x}} = \mathbf{x} - \mathbf{x}$, which is assumed to be unbiased, $E[\mathbf{x}] = \mathbf{0}$, and have a covariance

$$E[\mathbf{\hat{x}}\mathbf{\hat{x}}^T] = \begin{bmatrix}
\sigma_x^2 & \rho_{x\gamma}\sigma_x\sigma_y & \rho_{x\gamma}\sigma_x\sigma_y \\
\rho_{x\gamma}\sigma_x\sigma_y & \sigma_y^2 & \rho_{x\gamma}\sigma_x\sigma_y \\
\rho_{x\gamma}\sigma_x\sigma_y & \rho_{x\gamma}\sigma_x\sigma_y & \sigma_y^2
\end{bmatrix}$$

(8)

Defining the balance index as

$$\mathbf{b a l} = \left|1 - \frac{2n\sigma_x}{\sigma_y}\right|$$

(9)

then the semimajor axis standard deviation in Eq. (7) can be rewritten as

$$\sigma_{\Delta a} = 2\sqrt{\left(\frac{\mathbf{b a l} \cdot \sigma_y}{n}\right)^2 + \frac{4\sigma_x\sigma_y}{n}(1 + \rho_{x\gamma})}$$

(10)

This relationship makes it clear that the two conditions for zero semimajor axis standard deviation are: 1) correlation requirement $\rho_{x\gamma} = -1$, that is, the radial position and in-track velocity are linearly correlated and 2) balance requirement $\mathbf{b a l} = 0$ or, equivalently, that $\sigma_x = 2n\sigma_y$. Reference [4] proposed a method for representing the relationship between $\sigma_x$, $\sigma_y$, $\rho_{x\gamma}$, and $\sigma_{\Delta a}$ that is illustrated in Fig. 1. The $x$ and $y$ axes of the plot are the standard deviations of the position and velocity estimation errors. Contours of constant semimajor axis standard deviation are shown on the figure. Each contour is associated with a value of $\rho_{x\gamma}$; several values of $\rho_{x\gamma}$ are shown for each level of $\sigma_{\Delta a}$. The diagonal of peaks indicates where $\sigma_x = 2n\sigma_y$. Along the diagonal of peaks, the lines of constant semimajor axis experience a “bump” that increases in size as the correlation tends toward $-1$. This bump corresponds to increasing cancellation between the error in $x$ and $\dot{y}$ that results from increasing correlation in these errors. Essentially, if the errors have high correlation and the proper balance, the higher error levels can be tolerated while maintaining low levels of semimajor axis error. Each point on the graph corresponds to a unique set of $\sigma_x$ and $\sigma_y$. However, many points on the graph are intersected by more than one contour of constant semimajor axis. The correlation determines the specific contour on which the system lies.

Remark 1: Figure 2 shows contours for a particular semimajor axis standard deviation error ($\sigma_{\Delta a} = 1$ m) for varying values of correlation. The plot also shows lines of constant balance index. Clearly, $\mathbf{b a l}$ should be zero when the balance requirement is met. The plot shows that if the balance is off, that is, $\mathbf{b a l} \approx 0.5$ or $\mathbf{b a l} \approx | -1 |$, then the effects of the correlation are reduced. This reinforces the observation that for systems governed by the HCW equations, both the correlation and balance requirements must be satisfied to obtain a significant reduction in the semimajor axis error.

III. Detailed Filter Analysis

This section presents analytical relationships for a Kalman filter between the process and measurement noise levels and the SMA balance and correlation requirements. The link between the two is established using the algebraic Riccati equation. If they exist, analytic solutions to the Riccati equation provide insights into the effect of the parameters in the estimation problem, but these are typically very difficult to obtain for high-order systems. However, for a Kalman filter with a measurement update every $\Delta t \approx 1$ s for a 90-min orbit [11,12], the coupling between motion in the $x$ and $y$ directions is very weak and the dynamics can be well approximated as two weakly coupled double integrators. This section uses this
approximation to perform a detailed analysis of the solution of a continuous Kalman filter, first for a system with position measurements and then for a system with position and velocity measurements. The following sections extend the numerical analysis to the more complicated case with a discrete Kalman filter.

Starting with the continuous Kalman filter using the in-plane components of Eq. (3)

$$\ddot{x} - 2n\dot{y} - 3n^2x = f_x$$  \hspace{1cm} (11)$$

where $f_x$ and $f_y$ are disturbance accelerations with identical spectral densities $\sigma_0^2$. These equations of motion can be written in state space form using $X = [x, \dot{x}, y, \dot{y}]^T$ and dynamics matrices $A_0$ and $B_0$. When only position measurements are available, the output matrix is

$$H_b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$  \hspace{1cm} (13)$$

Fig. 1 Contours of constant semimajor axis vs position and velocity accuracy. Contours given for three levels of correlation.

Fig. 2 Contours of constant balance (straight lines) illustrate regions where correlation affects SMA accuracy. Numbers indicate balance index and asterisks indicate points of perfect balance on the SMA contours.
with sensing noise given by $\sigma_n^2$ on each measurement. These dynamics can be transformed to a new system of equations with the state $\tilde{X} = [x, x', y, y']^T$, where $(*)' = (*)&t$, $\tilde{T}_pX$, and

$$T_p = \text{diag}(1, \Delta t, 1, \Delta t)$$

yield

$$x'' = 2\epsilon y' + 3\epsilon^2 x + (\Delta t)^2 f_x$$

and

$$y'' = -2\epsilon x' + (\Delta t)^2 f_y$$

with $\epsilon = n\Delta t$. Note that when $\epsilon \ll 1$, which is true for the cases of interest in this section, the $x$ and $y$ orbital dynamics can essentially be written as double integrators for which the solution of the Riccati equation is easily found. The combined dynamics then consist of these two double integrators with coupling terms of order $\epsilon$ and $\epsilon^2$.

Remark 2: A Kalman filter, by definition, gives the minimal variance estimates of the relative states [13], which can then be used to produce the approximate minimum variance estimate of the semimajor axis difference. This follows because the relative semimajor axis is approximately a linear combination of radial position and in-track velocity, as shown in Eq. (5). The state vector could be transformed into a state that explicitly includes the semimajor axis:

$$x, = \begin{bmatrix} \Delta a \\ \dot{x} \\ \dot{y} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 2/n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{x} \end{bmatrix} = T_X$$

where $T$ is the invertible transformation matrix between the nominal state and the transformed state. The Kalman filter objective function is related to the magnitude of the estimation error, which can be transformed as follows:

$$(\dot{x}, - x)'(\dot{x}, - x) = (\dot{x} - x)'T_T(\dot{x} - x) = \dot{x}^T S \dot{x}$$

From [13], the optimal estimate for the state $x$ is found by minimizing the cost function:

$$J = E[\dot{x}^T S \dot{x}]$$

where $S$ is any positive semidefinite matrix. The key point is that the optimal estimate is independent of the choice of $M > 0$ [13]. Because we can choose $M = I$, or $M = S$, as in Eq. (16), then the optimal estimate for $x_i$ will be related to the optimal estimate for $x_i$ by the linear transformation $T$. Thus, a Kalman filter estimating relative position and velocity necessarily will also yield the minimum variance estimate of the approximate relative semimajor axis, to within the error associated with the linearization.

A. Correlation with Position Measurements

The dynamics in Eqs. (14) and (15) are written in state space form as

$$\tilde{X}' = \tilde{A} \tilde{X} + \tilde{B} f$$

where $\tilde{A} = \Delta t(T_pA_0T_p)$, $\tilde{B} = \Delta t(T_pB_0)$, and $\tilde{H} = H_0$. The differential filter Riccati equation for the original system [Eqs. (11) and (12)] must also be transformed using $T_p$, and the result is that, at steady state

$$0 = \tilde{A} P + P \tilde{A}^T + \tilde{B} Q \tilde{B}^T - P \tilde{H} R^{-1} \tilde{H} P$$

where $Q = (\sigma_0^2/\Delta t)I_2$, and $R = (\sigma_1^2/\Delta t)I_2$, with the factors of $1/\Delta t$ resulting from the transformation. To proceed, the covariance for the transformed state is represented as a Taylor expansion in $\epsilon$.

$$P = P_0 + \epsilon P_1 + \epsilon^2 P_2 + \ldots$$

$$\tilde{A} = A_0 + \epsilon A_1 + \epsilon^2 A_2 + \ldots$$

in the Riccati equation and grouping terms in the same power of $\epsilon$, it is possible to solve for the expansion of the covariance matrix ($P_{0\epsilon}$, $P_{0\epsilon^2}$, and $P_{0\epsilon^3}$, $k = 0, 1, \ldots$). The $\tilde{A}$ matrix in Eq. (17) gives the expansion matrices $A_0$ and $A_1$:

$$A_0 = \begin{bmatrix} 4 & 0 & 0 & 2/n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Define the vectors

$$x = [x \ x']^T$$

and

$$y = [y \ y']^T$$

Then $P_{0\epsilon} = 0$ for the solution of the two independent double integrators. Using

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \Delta \epsilon t^2 \end{bmatrix} \quad H = [1 \ 0]$$

the first-order solution to the Riccati equation is

$$P_{0\epsilon} = P_{0\epsilon^2} = \begin{bmatrix} P_{011} & P_{012} \\ P_{021} & P_{022} \end{bmatrix} = \begin{bmatrix} \sqrt{2} \sigma_0^2 \sigma_R^2 & \sigma_0 \sigma_R \Delta t \\ \sigma_0 \sigma_R \Delta t & \sqrt{2} \sigma_0^2 \sigma_R^2 \Delta t \end{bmatrix}$$

Substituting into the Riccati equation and isolating $O(\epsilon)$ terms yields

$$P_0 A_0^T + P_1 A_0 + A_0 P_1 + A_1 P_0 = P_0 D P_1 + P_1 D P_0$$

$$P_1 (A_0^T - D P_1) + (A_0 - P_0 D) P_1 = -(P_0 A_0^T + A_1 P_0)$$

where

$$D = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \quad C = \frac{\Delta \epsilon t}{\sigma_R} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Using the knowledge that $P_{0\epsilon} = P_{0\epsilon^2}$,

$$(A_0^T - D P_1) = \begin{bmatrix} E & 0 \\ 0 & E \end{bmatrix}$$

and

$$A_0 - P_0 D) = \begin{bmatrix} E^T & 0 \\ 0 & E^T \end{bmatrix}$$

Defining $G$ as

$$G = A_1 P_{0\epsilon} - P_{0\epsilon} A_1^T = \begin{bmatrix} 0 & -2P_{0\epsilon^2} \\ 2P_{0\epsilon^2} & 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 & G \\ -G & 0 \end{bmatrix}$$
Substituting into the Riccati equation yields
\[
\begin{bmatrix}
P_{1x} & P_{1y} & P_{1y} & P_{1x}
\end{bmatrix}
= - \begin{bmatrix}
0_2 & E
\end{bmatrix}
\begin{bmatrix}
E^T & 0_2
\end{bmatrix}
\begin{bmatrix}
P_{1x} & P_{1y} & P_{1y} & P_{1x}
\end{bmatrix}
\]
which is a system of three matrix equations
\[
P_{1x}E + E^T P_{1x} = 0_4
\]
\[
P_{1y}E + E^T P_{1y} = 0_4
\]
\[
P_{1y}E + E^T P_{1y} = - G
\]
The solution to Eqs. (34) and (35) is \(P_{1xx} = P_{1yy} = 0\), and Eq. (36) gives
\[
P_{1x} = - \frac{2\sigma^2_{\rho_{0j}}}{\Delta t P_{0j0j}} P_{1y} = - P_{1y} P_{1y} = P_{1yy} = 0
\]
Then the correlation coefficient can be found from
\[
\rho_{xy} = \frac{P_{xy}}{\sqrt{P_{xx}P_{yy}}} = - \frac{\rho_{xy}^2 / \Delta t}{\sqrt{P_{xx}P_{yy}^2 / \Delta t}} \approx \rho_{xy}^2 / \Delta t
\]
Using the expressions previously given, it follows that
\[
\rho_{xy} \approx - \frac{2(n\Delta t)^2 \sigma^2_{\rho_{0j}}(\sigma_{\rho_{0j}}^2 + \sigma_{\rho_{0j}}^2 + \sigma_{\rho_{0j}}^2)}{\Delta t^2} = - \frac{\sqrt{2\rho_{xy}^2}}{\sigma_{\rho_{0j}}} \Delta t \tag{39}
\]
which, upon substitution in Eq. (7), gives a semimajor axis error of
\[
\sigma_{a_{\Delta t}} \approx \frac{(2^{3/4} / \pi)\sigma_{\rho_{0j}}^3}{\sigma_{\rho_{0j}}^2 / 4}
\]
Finally, applying the state transformation \(T_p\) to \(P_{0yy}\) in Eq. (24), it follows that, because
\[
\sigma_{\rho_{0j}} \approx \frac{(2^{3/2} \sigma_{\rho_{0j}}^2) / 2}{\sigma_{\rho_{0j}}^2}
\]
then the semimajor axis error can be rewritten as
\[
\sigma_{a_{\Delta t}} \approx \frac{2 / \pi}{\sigma_{\rho_{0j}}^2}
\]
Note that in Eq. (42) the resulting semimajor axis variance is only a function of the in-track velocity variance, even though the semimajor axis error is a function of the radial position and in-track velocity errors. This occurs because the radial position variance and correlation terms are equal and of opposite sign, and so they cancel each other.

Equations (40) and (42) are the predictions of the correlation and semimajor axis variances from a perturbation analysis of a continuous Kalman filter that has two decoupled double integrators as the first approximation. Therefore, some discussion of the effect of these approximations on the validity range of the answers is in order. However, as will be seen, these equations agree well with the simulation results shown in later sections. The small parameter in the perturbation analysis is \(\epsilon = n\Delta t\), therefore, as \(\epsilon \to 0(1)\), they will no longer be valid. For low Earth orbits \(n \approx 0.001\), consequently, when \(\Delta t > 100\ s\), the approximation should be expected to start degrading and when \(\Delta t \approx 1000\ s\) it would not be expected to be valid. In addition, by definition \(|\rho_{xy}| \leq 1\), which gives that
\[
\sigma_{a_{\Delta t}} \approx \frac{2 / \pi}{\sigma_{\rho_{0j}}^2}
\]
Because \(n \approx 0.001\), this implies that \(\sigma_{\rho_{0j}}\) should be no more than six orders of magnitude larger than \(\sigma_{\rho_{0j}}\). For example, consider that many space-rated GPS receivers produce differential carrier-phase measurements with millimeters of error. Equation (43) indicates that to achieve \(\rho_{xy} = -1\), the dynamics environment of the vehicle would have to be modeled to nanometer-level accuracy, which is currently not possible. For a typical scenario \([11]\), \(\sigma_{\rho_{0j}} = 1 \times 10^{-6} \ m / s^{3/2}\) and \(\sigma_{\rho_{0j}} = 5 \times 10^{-3} \ m / s^{3/2}\), which gives that
\[
\rho_{xy} \approx 1 \times 10^{-3} \sqrt{5 \times 10^{-3}} \approx 0.0707 < 1
\]
which is consistent with the low levels of correlation found in current CDGPS results.

### B. Examination of Balance Requirement
One strategy for minimizing the semimajor axis error is to achieve a high correlation and to simultaneously have a balance of errors given by
\[
\sigma_{\rho_{0j}} / \sigma_{\rho_{0j}} = 2n
\]
For the transformed state described by Eqs. (14) and (15), this requirement is
\[
\sigma_{\rho_{0j}} / \sigma_{\rho_{0j}} = 2n \Delta t
\]
Equation (24) showed that for the HCW equations, to first order, the standard deviations for in-track velocity and radial position estimates are
\[
\sigma_{\rho_{0j}} \approx \sqrt{P_{0j0j}} = \left(2^{3/2} \sigma_{\rho_{0j}}^2 / \Delta t^2 \right)^{1/2}
\]
\[
\sigma_{\rho_{0j}} = \sqrt{2^{3/2} \sigma_{\rho_{0j}}^2 / \Delta t^2}
\]
Substituting into Eq. (44) gives the analytic expression for the balance condition
\[
\sigma_{\rho_{0j}} / \sigma_{\rho_{0j}} = \frac{\sqrt{2^{3/2} \sigma_{\rho_{0j}}^2 / \Delta t^2}}{\sqrt{2^{3/2} \sigma_{\rho_{0j}}^2 / \Delta t^2}} = \Delta t \frac{\sigma_{\rho_{0j}}}{\sigma_{\rho_{0j}}}
\]
Using Eq. (44), perfect balance requires that
\[
\Delta t \frac{\sigma_{\rho_{0j}}}{\sigma_{\rho_{0j}}} = 2n \Delta t \Rightarrow \sigma_{\rho_{0j}} / \sigma_{\rho_{0j}} = 1 / 2n
\]
This expression can be substituted into Eq. (40) to find the correlation that is achieved when the balance is correct
\[
\rho_{xy} = -n \sqrt{\sigma_{\rho_{0j}} / \sigma_{\rho_{0j}}} = -n (1 / 2n) = -0.5
\]
This analysis shows that, for the Kalman filter, achieving the required balance is incompatible with achieving a correlation of \(-1\).

### C. Correlation with Position and Velocity Measurements
In this example, the system is augmented with a velocity sensor to investigate if direct velocity measurements can be used to achieve the correlation and balance requirements. The sensor noise and measurement matrices for the transformed double integrator dynamics now take the form
\[
R = \frac{1}{\Delta t} \begin{bmatrix} \sigma_{\rho_{0j}}^2 & 0 \\ 0 & \sigma_{\rho_{0j}}^2 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \\ 0 & \Delta t^{-1} \end{bmatrix}
\]
\[
H = \begin{bmatrix} 1 & 0 \\ 0 & \Delta t^{-1} \end{bmatrix}
\]
and the process noise term remains \( Q = (\sigma_Q^2/\Delta t)I_2 \). Substituting into the Riccati equation and solving for the terms in the first-order covariance matrix yields

\[
P_{01} = \sqrt{2\sigma_{\alpha_2}^2/\sigma_Q^2} \left[ \frac{1 + 0.5K}{1 + K} \right] P_{02} = \sqrt{2\sigma_{\alpha_2}^2/\Delta t^2} \left[ \frac{1 + 0.5K}{1 + K} \right] P_{012} = \frac{\sigma_{\nu\nu} \sigma_{Q\Delta t}}{(1 + K)}
\]

where \( K = \sigma_Q^2/\sigma_{\alpha_2}^2/\sigma_{\nu_2}^2 \). Substituting Eq. (52) into the HCW expansion in Eq. (26) yields a matrix equation in the same form as Eq. (33), however, the matrix \( D \) is now redefined as

\[
D = \begin{bmatrix} \tilde{C} & 0_x \\ 0_y & \tilde{C} \end{bmatrix} \quad \tilde{C} = \begin{bmatrix} \Delta t/\sigma_{\nu_2} & 0 \\ 0 & 1/(\sigma_{\nu_2}^2 \Delta t) \end{bmatrix}
\]

This set of matrix equations also indicates that \( P_{1x} = P_{1y} = 0_x \). The remaining solution is of the form

\[
P_{1x} = P_{1y} = 0_x \]

Solving for the correlation coefficient \( \rho_{\nu_2} \) gives

\[
\rho_{\nu_2} \approx \frac{\rho_{\nu_2}}{(P_{01}P_{012})^{1/2}} \approx -\frac{\sigma_{\alpha_2}}{\sigma_Q} \left[ 1 + 0.5K \right] \]

From Eq. (56), it can be seen that as the velocity measurement becomes more accurate (\( K \) increases), the correlation magnitude is reduced. This is consistent with the observation that the filter makes more use of dynamics when there are fewer measurements. Substituting this value for \( \rho_{\nu_2} \) into the expression for semimajor axis standard deviation yields

\[
\sigma_\alpha = \frac{2^{3/4}}{n} \sigma_{\alpha_2}^{3/4} \sigma_Q^{1/4} \left[ \frac{\sigma_{\alpha_2} \sigma_{\nu_2} + 4\sigma_r^2 n^2 + 2\sigma_r^2}{\sigma_{\alpha_2} \sigma_{\nu_2} + \sigma_r^2 n^2 + 2\sigma_r^2} \right]^{1/2} \times \left[ \frac{1 + 0.5K}{1 + 0.5K} \right]^{1/2} \left[ 1 + 0.5 \left( \frac{\sigma_{\nu_2}}{\sigma_{\nu_2}} \right)^2 \frac{1}{1 + 0.5K} \right]^{1/2}
\]

where the new semimajor axis standard deviation is no longer solely a function of the velocity accuracy. Equation (57) predicts that SMA knowledge can be improved by reductions in process noise and improvements in position and velocity sensing accuracy.

### IV. Analysis on a Linear Planar Model

A Kalman filter produces the minimum variance estimate of the approximate relative semimajor axis, but does not necessarily meet the balance and correlation requirements. Numerical simulations in this section show the filter cannot be forced to meet both the balance and correlation requirements by adjusting the input parameters, supporting the analytical conclusions of the previous section. There are many parameters that must be specified for a Kalman filter and all of these affect the accuracy of the estimates. The dynamics and measurement models, the operating environment, and the extent to which nonlinearities are accentuated may all affect the process \( Q \) and sensor \( R \) noise intensities. The set of sensor data made available to the filter will determine the measurement matrix \( H \). Also, the time step can affect the performance of the discrete filter. The relationship between design parameters and navigation accuracy is investigated and this should lead to a better understanding of what changes might be required to improve navigation.

A CDGPS navigation filter has nonlinearities in both the system dynamics and the measurement equations, and because the set of visible GPS satellites changes, the measurement matrix \( H \) and the geometric dilution of precision (GDOP) will change, and the state vector length will grow or shrink as the set of estimated carrier biases changes [12]. These factors make it difficult to understand direct relationships between the filter parameters and the navigational accuracies. Thus, the following starts with a simplified linear planar model (LPM) to develop insights into the behavior of a relative navigation filter using CDGPS. Table 1 summarizes the Kalman filter parameters considered in the LPM simulations.

The system dynamics in the LPM are taken from the solutions of the HCW equations for radial and in-track position and velocity. Out-of-plane motion is ignored, because it does not affect semimajor axis error. The dynamics model is not varied in the simulations, but its effective accuracy is modified by evaluating filter performance with different values of \( Q \). Most state-of-the-art GPS receivers can provide as many as 12 pseudorange measurements. Similarly, the LPM includes two or more direct measures of position that span the orbital plane. Variations of the measurement model included changing the angle between two position measurements and increasing the number of measurements included. Also, the level of noise associated with the measurements is varied. Different discrete time steps \( \Delta t \) are considered in these simulations. Note that as the time step changes, the relative importance of the dynamics model and the measurements will change. For example, if highly accurate measurements are provided at a very fast rate, the dynamics model might be of little importance. Conversely, once the estimate has converged, a perfect propagator might not need any future measurements. Changing the time step should illuminate how the filter might favor one of these extremes.

The simplified LPM problem will show how each parameter affects the radial position error and in-track velocity error, the correlation of the two, and, ultimately, the semimajor axis error. For each design variation, the position and velocity variances are found by numerically solving a discrete algebraic Riccati equation. The position and velocity error variances from Eq. (18) are used in Eq. (7) to compute the corresponding semimajor axis error.

### A. Two LPM Examples for Correlation Demonstrations

Because the measurements span the orbital plane, it is intuitive that the correlation between radial position and in-track velocity will increase only when the estimate depends more on the dynamics model embedded in the filter. The following examples (see results in Figs 3–6) induce this behavior:

1) The angle between the two measurements is increased from 0 to 90 deg, as in Fig. 3. This causes the direct observability of the orbital plane to decrease as the two measurements become aligned and results in the term \( H^T P^{-1} H \) in the Kalman filter losing full rank. Because the measurements no longer span the full orbital plane, the dynamics model must be employed to create estimates of all \( x \) and \( y \) states.

2) The number of measurements is reduced from six to one, causing changes in the direct observability.

The results from these examples are discussed next.

**Example 1:** This example provides an excellent view of how the span of the measurement matrix affects the correlation and semimajor axis error. The two position measurements have equal accuracy and are initially aligned with the \( x \) and \( y \) axes. The angle between the two measurements is gradually decreased until the
measurement directions are nearly colinear at a 45-deg angle in the x–y plane. Figure 5 shows that the correlation between the two remains low until the angular separation is less than ~20 deg. When the angular separation becomes smaller than a degree, the correlation drops sharply from ~0.7 toward ~1. Clearly, as the measurements approach alignment, the uniqueness of the information that each contributes decreases, and the results show that the correlation rapidly approaches ~1 as the system dynamics become more important. However, when the measurements no longer span the orbital plane, the results also show that the position and velocity errors increase dramatically. Because the balance requirement is not met exactly, the increased errors in position and velocity outweigh the advantage of the increased correlation, and the final semimajor axis error is large. In the example shown, the two measurement directions converged on 45 deg, but similar results were seen for other terminal angles.

Example 2: This example, seen in Fig. 4, looks at the effect of changing the number of measurements, which are equally spaced in the orbital plane. When a second measurement is added, the values of $\sigma_x$ and $\sigma_y$ drop sharply (see Fig. 6), $\rho_{xy}$ changes from close to ~1 to approximately ~0.5, and the balance index increases. The state estimates are further from meeting the correlation and balance requirements, but the semimajor axis knowledge improves. In cases where correlation and balance conditions cannot be met simultaneously, the goal for achieving low semimajor axis error can still be achieved by minimizing estimation errors. The results of this example also show that, as further measurements are added, the position, velocity, and semimajor axis errors all improve, although the correlation decreases. This is consistent with behavior seen in the first example.

Remark 3: In these examples, the filter can be forced to rely on its dynamics model to extract good state information in the radial and in-track directions. This increased reliance is accomplished by degrading the available sensing information. The increases in $\sigma_x$ and $\sigma_y$ that accompany the higher correlation consistently result in a larger semimajor axis error. This is a trait of a filter that is measurement-dependent, not a symptom of deficiency. When the correlation between radial position and in-track velocity is low, the best strategy for determining relative semimajor axis is to estimate both quantities with the highest possible accuracy, and the Kalman filter does exactly that. The behavior seen in both variations of this simplified example provide insight into the relative navigation system using CDGPS measurements. A GPS-based navigation filter will typically have many position-related measurements available (9–12). The estimator can be forced to have better correlations by degrading the measurements and boosting the importance of the dynamics model. However, these examples show that this would result in increased semimajor axis error and is a suboptimal strategy.

B. LPM Simulations with $Q$ and $R$

This section investigates the relationship between the Kalman filter parameters and the resulting estimate accuracy. The levels of measurement and process noise are indicators of how well the sensors and the dynamics are modeled, and their relative values determine how the filter weighs new measurement information against the current state estimate propagated using the dynamics.
Meeting the balance and correlation requirements discussed in Sec. I corresponds to being on the bump in Fig. 1. The LPM results shown here occupy the region above the \( \sigma_i = 2n\sigma_x \) line, which occurs because the balance requirement is not met and is dominated by the velocity errors for this position-measurement-based CDGPS system. This leads to the question of how changing filter inputs will move the output closer to or further from the bump. To answer this question, the LPM simulation was run for a range of measurement and process noise levels. For each unique assignment of \( Q \) and \( R \), the resulting error variances for radial position and in-track velocity, \( \sigma_x \) and \( \sigma_y \), were recorded. The corresponding semimajor axis error \( \sigma_a \) was calculated using Eq. (7). Fig. 7 shows lines of constant \( Q \) and \( R \) on axes of \( \sigma_x \) and \( \sigma_y \), where \( Q = \text{diag}(Q_x, Q_y, Q_z) \), \( R = \sigma_c^2 I_2 \), \( \sigma_c \) is the continuous process noise covariance, and \( \sigma_i \) is the discrete measurement covariance. The diagonal of peaks on this graph indicates the location of the bump, where the balance and correlation requirements are met (which means \( \sigma_i = 2n\sigma_x \)). By moving from one line of constant \( Q \) or \( R \) to another, one can see how decreasing the process or measurement noise would change the resulting position and velocity error.

Several graphs are presented to demonstrate the relationship between \( Q \), \( R \), and \( \sigma_{ax} \). First, Fig. 8 reproduces Fig. 7 and lines of constant semimajor axis error are added. The lines for constant \( Q \) and \( R \) are dimmed for clarity. Note the lines of constant semimajor axis are horizontal, which corresponds with the horizontal sections of the semimajor axis contours on Fig. 1. The effect of changes in \( Q \) and \( R \) on \( \sigma_{ax} \) can be assessed by looking at the constant lines for all three values. Because the lines of constant semimajor axis are horizontal, improvement in \( \sigma_{ax} \) can only be accomplished by moving in the vertical direction on the graph, which is equivalent to decreasing \( \sigma_i \).
Whether this requires decreasing $Q$ or $R$ depends on the angles between the horizontal lines of constant $\sigma_{\Delta \alpha}$ and the contours of constant $Q$ and $R$.

Two regions are indicated in Fig. 9. Region 1 (upper left hand portion of the graph) contains lines of constant $Q$ and $R$ that are essentially horizontal and vertical, respectively. Improving the measurement noise in this region (or, essentially, moving horizontally in the graph) has minimal effect on semimajor axis knowledge. Decreasing $\sigma_{\Delta \alpha}$ would require improving the process noise to enable vertical movement on the graph. Region 2 is closer to the line of $\sigma_{\alpha} = 2n\sigma_{\dot{\alpha}}$. Here, the lines of constant $Q$ and $R$ are no longer parallel and perpendicular to the horizontal lines of constant semimajor axis. This means that reducing either $Q$ or $R$ could improve $\sigma_{\Delta \alpha}$. This shift from region 1, where sensing improvement has virtually no effect on $\sigma_{\Delta \alpha}$, to region 2, where it does, is also shown in Fig. 10 by plotting contours of constant semimajor axis on axes of $Q$ and $R$. As expected, region 1 has horizontal lines of constant $\sigma_{\Delta \alpha}$, which directly shows that decreasing the measurement noise has little effect on semimajor axis knowledge. The portion of the graph where the contours of constant $\sigma_{\Delta \alpha}$ are sloped corresponds to region 2. In this region, improvements in the measurements or the dynamics models (i.e., reduced process noise) would contribute to improved semimajor axis knowledge.

C. Discrete Simulations for Varying $\Delta t$

One additional way to put more emphasis on the dynamics model is to reduce the measurement rate. Prior analysis of the CDGPS filter by Busse [12] used a 1 Hz rate, although much longer time steps were considered in [11]. One difficulty with increasing the propagation time is that the nonlinearity in the orbit propagation becomes much more significant and more sophisticated models and propagation algorithms must be used, especially for the error covariance [14,15]. The following investigation of the effect of varying measurement update rate by changing the discretization time step uses the linear model and thus ignores these effects.
To do this, a family of discrete Riccati equations was solved using HCW dynamics at various discretization times. In each case, the constant spectral density matrix $Q_c$ associated with the process noise of the continuous dynamics was converted to the appropriate discrete process noise $Q_d$ using the conversion algorithm in [16] (DISRW in Chapter 9). The simulation results in Fig. 11 were done for several values of $\sigma_{q_d} = \{10^{-5}, 10^{-6}, 5 \times 10^{-7}\}$ m/s$^{3/2}$ ($Q_c = \sigma_{q_d}^2 I_3$) and a constant sensing noise covariance $R = (5 \times 10^{-3}$ m)$^2$. The plot shows that, as the discretization time step is increased, the correlation coefficient tends to $-1$. This trend is expected, because longer propagation times will force increased filter dependence on the equations of motion, translating into increased overall correlation. However, the proper position to velocity error balance is not achieved, causing the SMA accuracy to degrade. The plot clearly shows the role of the process noise $Q_c$ in determining the SMA error growth. One caveat about these results is that the correlation approaches $-1$ for filter time steps larger than 1000 s, for which the nonlinearity in the system dynamics, that this analysis ignores, may begin to play an important role [11].

Comparing this result with Fig. 2, it can be seen that although the balance index is not zero, it is producing the effect of lowering the overall semimajor axis in combination with the correlation. However, the canceling effects of correlation and balance are not sufficient to prevent the semimajor axis error from growing rapidly as the time step is increased. Thus, decreasing the measurement update rate to produce higher correlation is not a viable strategy for reducing semimajor axis error.

These examples considered the navigation filter accuracy for a linear planar model, and the next section explores similar questions using a full nonlinear GPS model (NGM).

V. Nonlinear GPS Model

The LPM simulations provide a base for discussion of the relationship between noise and filter performance in the full nonlinear GPS model. The NGM adds the clock and bias terms to the filter state definition and introduces uncertainties associated with the clock, ionosphere, absolute state error, etc. A similar exploration of the relationship between $Q$, $R$, and $s_{\Delta a}$ is undertaken.

Two sets of NGM simulations, discussed next, will illustrate the relationship between process and measurement noises and estimator performance. The NGM analysis begins with a very simple model.
that focuses on the relative orbital dynamics and GPS measurements and takes steps to reduce nonlinearities associated with their equations. The second set of simulations incorporates elements that more closely resemble a real-world scenario.

The NGM simulation environment is an extension of tools developed by Busse [11]. The user can specify which nonlinear effects and perturbations are included in the truth model and how they are accounted for in the filter, which is useful when comparing the NGM and the LPM results. The NGM simulation propagates all relevant vehicle truth states and provides simulated measurements to the estimator. Because a large number of simulations were required to observe relationships between filter parameters and the navigation accuracy, it was not practical to employ hardware-in-the-loop simulations that could provide the filter with actual measurements. The absolute truth states for the vehicles were used to create the simulated GPS measurements. The relative truth states were created by differencing the absolute states.

As in the LPM experiments, for each set of simulations, the NGM was run for an array of values of $Q$ and $R$. Each full GPS simulation was run for 3000 s, to allow sufficient time for the filter to converge to steady state. The position and velocity errors were determined from the standard deviation of the estimation error, not from a Riccati equation as in the LPM cases. Note that in the nonlinear case, Eq. (7) was found to give inaccurate estimates of the semimajor axis error, and so orbital elements calculations were required. The absolute semimajor axes were differenced at each time step and used to find the estimation error for the differential semimajor axis. The square root of the variance of the steady-state estimation errors was recorded.

The goal of the first simulation set is to uncover the basic relationships between process and measurement noises and the semimajor axis error. These first simulations use the full NGM state, with clock and carrier bias terms included, and incorporates simulated GPS measurements. However, to maintain LPM-like simplicity, many real-world effects, such as ephemeris error, clock error, communication outages, and measurement cycle slips, are not included. Truncation error is reduced by placing the vehicles 1 m apart, and running the filter at a 0.1 s time step. The small separation distance also ensures the line-of-sight vectors to the GPS satellites for the two vehicles to be nearly identical (another assumption in the filter design).

The results of the first simulation set are shown in Fig. 12. The contours of constant $Q$, $R$, and $\sigma_{\Delta a}$ are very similar to those seen using the LPM. As in the LPM, Region 1 contains lines of constant $Q$ and $R$ that are essentially horizontal and vertical. Improving the measurement noise in this region would have no effect on semimajor axis knowledge. Reductions in process noise are required to reduce $\sigma_{\Delta a}$. In region 2, improvements in the measurements or the process noise would contribute to improved semimajor axis knowledge. These observations confirm that the results of a simplified version of the NGM approach those of the LPM.

The second set of NGM simulations incorporates all available error sources in the models for the environment and sensors, including errors introduced by clock, ephemeris, and absolute state uncertainty. This setup is more representative of a real-world CDGPS filter in the LEO environment. The results of this set of simulations are shown in Fig. 13, with lines of constant $Q$, $R$, and $\sigma_{\Delta a}$ on the axes of $\sigma_{\Delta a}$ and $\sigma_{\Delta a}$. The line of $\sigma_{\Delta a} = 2n\sigma_{ \xi }$ is included as a reference, but will not necessarily represent the same transition point that it did in the LPM. The region that coincides with typical CDGPS accuracy is indicated on the plot.

Contours of constant process noise tend to the horizontal direction and contours of constant measurement noise tend towards vertical. The parallelograms formed by the contours of constant $Q$ and $R$ are indicative of general agreement between the LPM and NGM results (see point A in Figs. 8 and 13). However, going from the LPM to the NGM, the horizontal lines of constant semimajor axis become positively sloped. This indicates that in the NGM, position accuracy will have some effect on $\sigma_{\Delta a}$, where in the LPM, it had no effect. The slope of the NGM contours is still small, suggesting the velocity error still dominates semimajor axis knowledge.

This leads to a key observation about the performance of our CDGPS-based relative navigation filter. Reducing either process or measurement noise can have a positive effect on semimajor axis knowledge. Previous work assumed that a coarse dynamics model would be sufficient, because the measurement updates are performed very frequently and, based on that premise, a very accurate filter was developed [11]. However, this NGM analysis suggests future work that incorporates a higher-fidelity dynamics model, thereby
reducing the process noise, could improve the semimajor axis knowledge.

A departure from LPM behavior is seen in the lines of constant measurement noise. In the LPM case, reducing measurement noise always resulted in reductions of $\sigma_v$. In region 1 of the LPM (Fig. 9) the lines of constant $R$ are vertical and orthogonal to the lines of constant semimajor axis. In other words, by increasing the sensor accuracy, it was always possible to keep improving $\sigma_v$. At some point, though, increasing sensor accuracy stopped improving semimajor axis knowledge. However, in the NGM cases, the contours of constant $R$ bunch together as the sensor noise becomes very small, which implies that further reductions will not impact either $\sigma_{\Delta a}$ or $\sigma_v$. The fact that $\sigma_v$ does not continue to improve as the measurement noise is decreased suggests that aspects of the NGM that reflect realistic phenomenon, such as clock error, become dominant. This leads to the second observation that noise sources other than the CDGPS measurements and the relative orbital dynamics model, such as from the clock and absolute state error, can limit the relative navigation filter performance. When these real-world errors are introduced in the NGM, there is a limit to how much the position, velocity, and semimajor axis performance can be improved. In contrast to the LPM, increasing sensor accuracy will not always result in similar increased position accuracy in the NGM.

Although some differences appear to be due to the nonlinearities in the NGM, the general trends, especially in the regions of the plots where typical CDGPS filter results are found, are retained. The similarities between the linear and nonlinear results suggest the observations from LPM-based filter behavior extend to filters based on the nonlinear dynamics and measurements. In particular, the insights gained from the LPM examples about the relationships between correlation, balance, and navigation errors, and between the process/measurement noises and the semimajor axis errors, are relevant to the more complex CDGPS filters.

VI. Conclusions

This paper was motivated by the desire to determine what metrics should be used to characterize the performance of a carrier-phase differential Global Positioning System relative navigation filter as “good,” and explore what parameters in the Kalman filter have the most impact on the performance of the navigation system. Previous works had suggested that a good navigation filter would yield estimates of the along-track velocity and radial position that satisfy both the correlation and balance requirements. This paper presented detailed analysis and numerical simulations to show that the optimal approximate relative semimajor axis estimate from a Kalman filter does not satisfy these properties, for any combination of measurement and process noise, which is consistent with recent practical experience with CDGPS filters, even when the state estimates are very accurate. These results also highlighted the importance of developing good models of the process noise to improve the semimajor axis estimation performance.

Acknowledgment

This work was funded under Cooperative Agreements NCC5-729 and NCC5-724 through the NASA GSFC Formation Flying NASA Research Announcement.

References

Diego, CA, 2004, p. 6; also American Astronautical Society Paper AAS-04-175.


Safe Trajectories for Autonomous Rendezvous of Spacecraft

Louis Breger∗ and Jonathan P. How†
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

DOI: 10.2514/1.29590

Autonomous spacecraft rendezvous is an enabling technology for many future space missions, but anomalies in recent flight experiments suggest that safety considerations will play an important role in the success of future missions. This paper presents a method for online generation of safe, fuel-optimized rendezvous trajectories that guarantee collision avoidance for a large class of anomalous system behaviors. We examine the cost of imposing safety as a problem constraint and of additional constraints that guarantee infinite horizon passive collision avoidance while enabling future docking retries. Tradeoffs between passive and active approaches to safety are examined. A convex formulation of the collision avoidance algorithm is introduced and shown to provide much faster solutions with only a small additional fuel expense. Numerous examples using both rotating and nonrotating targets are presented to demonstrate the overall benefits of incorporating these safety constraints when compared to nominal trajectory design techniques.

I. Introduction

AUTONOMOUS spacecraft rendezvous is an enabling technology for many future space missions [1]. Autonomous rendezvous has been used for docking with Mir [2], and more recently on the Engineering Test Satellite VII (ETS-VII) [3] and Demonstration of Autonomous Rendezvous Technology (DART) [4,5] missions. However, anomalies occurred during both of these last two missions. In the case of ETS-VII, multiple anomalies caused entries into safe mode over the course of the mission, at least one of which resulted in a preprogrammed maneuver to move the spacecraft 2.5 km from its target. The anomaly in the DART mission is thought to have resulted in excess fuel expenditures and appears to have caused an on-orbit collision [5–7]. These recent experiences suggest that autonomous rendezvous and docking would greatly benefit from the inclusion of additional safeguards to protect the vehicles in the event of failures. Designing approach trajectories that guarantee collision avoidance for some common failures could simultaneously decrease the likelihood of catastrophic failures in which one, or both, of the spacecraft are damaged and increase the likelihood that future attempts at docking succeed. This paper introduces a method for generating fuel-optimized rendezvous trajectories online that are safe with respect to a large class of possible spacecraft anomalies.

Numerous methods of generating and analyzing rendezvous trajectories exist in the literature and encompass a wide range of rendezvous scenarios [8–13]. These papers consider rendezvous from many perspectives, often taking into account complicated collision avoidance constraints, nonlinear rotational dynamics, and fuel efficiency. Another perspective to be considered when designing trajectories is safe behavior [12,14–16]. Safety in the context of spacecraft rendezvous and docking is typically with respect to collision avoidance following some type of failure. Drifting or passive fault tolerance is a safety technique used in both manual (human-in-the-loop) and autonomous rendezvous missions [12]. For example, the approach in [16] creates trajectories which naturally tend to drift away from the target spacecraft in the absence of thrusting, but which are not fuel optimized. Alternately, [12,14] develop the safety ellipse method, in which a nearby orbit with a relative trajectory with minimal secular drift is established that allows safe long-term observation before docking; however, this approach is not fuel optimized and does not propose a specific docking path. A method proposed in [15] optimizes both safety and fuel using genetic algorithms. This approach treats safety as a goal rather than a constraint and thus, cannot assure that the resulting trajectory would be safe. Roger and McInnes [11] plan passively safe trajectories using potential functions, but their approach is computationally intensive and limited to static obstacles. Various types of safety have been considered in the design of autonomous spacecraft trajectories, but these focused on creating trajectories that are safe under nominal operating conditions (e.g., safety from adversaries, uncertain terrain) [17,18].

This paper defines a safe trajectory as an approach path that guarantees collision avoidance in the presence of a class of anomalous system behaviors. Similarly, a passive safe trajectory guarantees collision avoidance with no thrusting required for safety and an active safe trajectory requires that inputs be applied to keep the system safe in the event of a failure. Note that this definition of safety is more restrictive than guaranteeing nominal collision avoidance because it guarantees that no collisions will occur for a range of faults. In particular, for a passive safe trajectory, safety is guaranteed even if the chaser spacecraft cannot use thrusters, computers, or communications equipment. The rationale behind choosing a passive abort strategy is threefold: 1) passive abort can protect against a large set of possible system failures simultaneously; 2) an abort trajectory that does not require fuel use guarantees that remaining fuel will not be expended rapidly to increase spacecraft separation distance, thereby increasing the likelihood that future docking attempts can occur; and 3) passive abort guarantees thrusting will not be used in close proximity to the target during an anomaly, thereby eliminating the danger of plume impingement during an automatic safe-mode maneuver. Active safety is less restrictive than passive safety, but it requires that the types of any failures be identified in real time and that some components of the control system remain operational so that a sequence of control inputs can be applied.

The following sections review a method for generating fuel-optimized trajectories from linearized relative dynamics and develop a novel approach for guaranteeing those trajectories will be safe. Several examples of safe trajectories generated for docking with both rotating and nonrotating target spacecraft establish that adding safety constraints does not result in significantly increased fuel use. Next, we examine additional constraints to guarantee desirable infinite horizon passive collision avoidance and ease of future docking attempts. To address online implementation considerations, a convex formulation of the safety problem is introduced that trades
II. Online Trajectory Optimization for Autonomous Rendezvous and Docking

A trajectory generated through online optimization can be designed by choosing the system inputs that produce that trajectory. For a linear system, methods for incorporating and propagating the effects of inputs are well known. The trajectory optimization formulation in this section is presented in the context of linear time-invariant dynamics, but there is no inherent restriction in the formulation preventing the use of time-varying dynamics [19].

Given a chaser satellite whose state is \( \mathbf{x}_k \) at time \( k \), the linearized dynamics of the system can be written as

\[
\mathbf{x}_{k+1} = A_d \mathbf{x}_k + B_d \mathbf{u}_k
\]  

where \( A_d \) is the state transition matrix for a single time step, \( B_d \) is the discrete input matrix for a single time step, and \( \mathbf{u}_k \) is the input vector at step \( k \). Typically, in a rendezvous situation, spacecraft would be in sufficiently close proximity to enable the use of the Hill–Clohessy–Wiltshire (HCW) equations [20], but approaches such as [21] can be used for more widely separated situations. Examples in this paper will use the HCW equations and hence the state \( \mathbf{x} \) is defined as

\[
\mathbf{x} = [x \ y \ z \ v_x \ v_y \ v_z]^T
\]  

where \( x, y, z, v_x, v_y, \) and \( v_z \) are the positions and velocities of a chaser satellite in the radial, in-track, and cross-track axes, respectively, of a local–vertical/local–horizontal (LVLH) frame positioned on the center of gravity of a passive target vehicle. The input is defined as

\[
\mathbf{u} = [u_x \ u_y \ u_z]^T
\]  

where \( u_x, u_y, \) and \( u_z \) are the inputs of the chaser vehicle in the axes indicated by the subscripts in the LVLH frame.

Given an initial state \( \mathbf{x}_0 \), the state at any future step \( k \) is [22]

\[
\mathbf{x}_k = A^k_d \mathbf{x}_0 + \left[ A^{k-1}_d B_d \ A^{k-2}_d B_d \ \ldots \ A_d B_d \ B_d \right] \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_{k-1} \end{bmatrix}
\]  

where \( A^k_d \) is the discrete convolution matrix. Because the effects of the control on the states are readily expressed as linear combinations of the inputs, a linear optimization can be formed that optimizes the constrained control commands and constrains the states of the system. The cost function for this optimization will exclusively penalize fuel use. In an actual maneuver implementation, it may be preferable to optimize both the fuel use and the maneuver duration (see [23]); however, in this paper only fuel use will be considered to simplify presentation and cost comparisons. The cost of the optimization \( J \) is given by

\[
J = \sum_{i=0}^{N-1} \| \mathbf{u}_i \|_1
\]  

where the 1-norm cost is used to capture the expenditure of fuel used, which is proportional to acceleration and \( \Delta V \), from axial thrusters. The optimal cost is then given by

\[
J^* = \min_{\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}} \sum_{i=0}^{N-1} \| \mathbf{u}_i \|_1
\]  

At each step \( k \), it is possible to constrain the state at that time to lie inside a convex region

\[
\hat{A}_d \mathbf{x}_k \leq \hat{b}_k
\]  

where \( \hat{A}_d \) is a matrix and \( \hat{b}_k \) is a vector that together capture a set of linear constraints on the state. Note that the costs and constraints in Eqs. (6) and (8) show an example linear implementation of a trajectory optimization, but in general the same concepts will be presented hold for nonlinear costs and constraints as well. Alternately, the state \( \mathbf{x}_k \) could be constrained to lie outside a region through the use of binary variables [8]

\[
\hat{A}_d \mathbf{x}_k \geq \hat{b}_k + M \mathbf{y}_k
\]  

\[
\| \mathbf{y}_k \|_1 \leq m - 1
\]  

where \( \mathbf{y}_k \) is a vector whose elements are constrained to be 0 or 1, and \( M \) is a large number on the scale of values taken by elements of \( \mathbf{x} \). This “Big M” method of collision avoidance works by allowing, at most, all but one of the collision avoidance constraints to be relaxed. A constraint is relaxed when the binary variable associated with it is set to 1, thereby making the right-hand side of the inequality very large and guaranteeing constraint satisfaction. Because at least one constraint is always guaranteed to not be relaxed, collision avoidance is assured (e.g., knowing that one is outside of one side of a box is sufficient information to guarantee that one is not in the box).

The inputs at each time step can also be directly constrained using

\[
\mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}
\]  

where \( \mathbf{u}_{\min} \) and \( \mathbf{u}_{\max} \) are vector bounds on the values of \( \mathbf{u} \). Typically, the minimum thrust at all times would be \( -\mathbf{u}_{\max} \). A detailed described of the full matrix forms used in linear trajectory optimizations for space vehicles can be found in [8,19].

This section reviewed an approach for creating fuel-minimizing trajectories that satisfy time-varying position, velocity, and thrusting constraints. Applications of these constraint types can insure that a spacecraft remains inside a line-of-site cone, and arrives at a docking port position at a particular time with a particular speed range. In addition, the control authority available over the course of the trajectory can be varied according to desired pattern.

III. Safety Formulation

The trajectories generated by the constraints in Sec. II will satisfy docking requirements and use minimal fuel to arrive at a rendezvous location. However, as is typical of optimal paths, the trajectories will approach constraint boundaries and generally be sensitive to uncertain behavior. Richards [23] and Tillerson and How [24] describe computationally feasible methods of generating trajectories online that are robust to process and sensing noise expected under nominal operating conditions. That type of robustness to uncertainty is distinct from the definition of safety for off-nominal conditions considered herein. This section presents an approach for generating trajectories that are safe with respect to a class of system failures. Although it would be desirable to avoid collisions and successfully complete docking in the presence of any system failure, it is unlikely that such a scenario is possible. Instead, a large subset of all possible failures is used, including guidance system shutdowns, which encompasses thruster failures, computer anomalies, and loss of sensing. The response to these types of failures would be a guidance system shutdown in which the chaser vehicle would go into a safe mode with all its thrusters turned off. Safety to this class of failures is called passive abort safety, because any rendezvous can be aborted using no thrusting. Passive abort safety guarantees collision avoidance for any failure that can be identified and responded to by disabling thrusters before the spacecraft trajectory is affected. This type of safety does not include failures in which a thruster fails on (see Sec. VIII).

A consequence of passive abort is that if thrusters are disabled at any step \( T \), counted from the start of the plan, during the trajectory implementation, then the thrusters will remain failed until the last
state of the chaser spacecraft relative to the target spacecraft is precisely known. In practice, this relative state is only known to within the accuracy provided by the navigation system. Likewise, the propagation used in Eq. (12) is only as accurate as the linear dynamics used to formulate that equation, because the actual vehicle would be subject to nonlinear dynamics, and disturbances from effects such as drag, J2, separation distance, and eccentricity. Equation (12) can also be rewritten to enable time-varying dynamics or an additional vector of modeled disturbances can be added to the model without increasing the complexity of the resulting optimization [19]. This permits a more sophisticated dynamics model to be used, which could reduce some of the effects of the modeling error [25]. To account for navigation error, the constraints in Eq. (14) can be made robust by posing them multiple times for a representative sampling of possible initial states that cover the space of likely navigation errors. Breger et al. [26] introduce such an approach and an algorithm for minimizing the effect of robustness constraints on the size of the resulting optimization.

IV. Scenarios

The rendezvous and docking scenarios to be examined in this paper involves a chaser spacecraft maneuvering to achieve docking with a target spacecraft. Figure 1 shows a target spacecraft that lies at the center of a local frame. A line-of-sight (LOS) cone protrudes from the target spacecraft and it is required that the rendezvous remain within this LOS cone for vision-based sensing. At the interface between the LOS cone and the target is a docking port (rectangular platform). In the rotating case (Fig. 2) the axis of rotation is the long axis of the spacecraft and the rotation rate is orbital. The choice of rotation axis and rate was arbitrary and only enters the optimization through their effect on the time-varying constraints imposed for LOS requirements, docking, and safety. The LOS requirements are

\[
A_{\text{LOS}} x_k \leq b_{\text{LOS}} \quad \forall \ k = 1 \cdots N
\]

where \(A_{\text{LOS}}\) and \(b_{\text{LOS}}\) describe the states within the LOS cone at a step \(k\) in the planning horizon. The terminal constraint is

\[
A_{\text{Term}} x_N \leq b_{\text{Term}}
\]

where \(A_{\text{Term}}\) and \(b_{\text{Term}}\) describe the states the spacecraft must occupy at the end of the planning horizon to achieve safe docking. These constraints can be both on position (e.g., enter a region within reach of a grappling arm) and on velocity (e.g., dock within a velocity range that produces acceptable stress on the docking port). In addition, time-varying bounds are introduced on the maximum thrusting levels to ensure large thrusts are not planned for the period immediately before docking. The safety constraints in Eq. (14) are imposed for the last quarter of the planning horizon. In the examples, an orbit with frequency \(n = 0.001\) rad/s is used and is discretized into 20 steps and the set of inputs that can fail is \(T \in \{14 \cdots 19\}\). The planning horizon is a full orbit. The chaser spacecraft, modeled after the mission in [27], has a mass of 45 kg and a maximum acceleration of \(10^{-3}\) m/s² during the first 17 steps of the plan and \(10^{-5}\) m/s² for the last three steps to prevent trajectory solutions with large terminal
thrusts. In addition, the docking constraint specifies that the velocity of the spacecraft at the time of docking be less than 1 m/s. In summary, the safety algorithm used in this section minimizes the $l_1$-norm of fuel consumption subject to thrust magnitude constraints, line-of-sight constraints, rendezvous constraints, and safety constraints. This algorithm can be stated as the optimization

$$\min_{u_0, \ldots, u_N} \text{Eq. (6)}$$

s.t. Eq. (11) $\forall k \in \{0, \ldots, N - 1\}$

Eq. (15)

Eq. (16)

Eq. (14) $\forall T \in \mathcal{F}$  (17)

In these examples, the safety horizon is a full orbit. Any of the design parameters in the safety implementation can be easily adjusted and in practice one would likely conduct a simulation study or analysis [28] to find the best combination for minimizing fuel use and guaranteeing feasible solutions.

A. Case 1: Stationary-Target Satellite

An optimized trajectory with no safety constraints [i.e., Eq. (17) without constraints using Eq. (14)] for the stationary-target (i.e., not rotating or translating in the local frame) case is shown in Fig. 3. The initial trajectory to the docking port roughly corresponds to a two-impulse V-bar (in-track) [12] approach. The nominal trajectory is marked with $\bullet$ and the failure trajectories with $\times$. Failure trajectories, the paths followed by the spacecraft in the event of a guidance shutdown, are shown for the last five possible inputs. Several of the failure trajectories overlap, a condition which corresponds to the nominal input at a step already being set to zero thrust. All of the failure trajectories clearly impact the target spacecraft. Figure 4 shows the same rendezvous situation for trajectories generated with safety constraints. In this case, none of the failure trajectories impact the target spacecraft. As in the case without safety, several of the failure trajectories overlap. The fuel costs (measured in $\text{mm}/\text{s}$) of the trajectory with no safety guarantees and the trajectory with safety are 1.29 and 1.91 mm/s, respectively. Hence, in this case, imposing safety results in a 48% increase in fuel use. To put these numbers in context, an optimized approach constrained to follow a V-bar trajectory (strictly in-track) would use 37.7 mm/s of fuel.

An approximate method for overbounding the optimized numbers would be to consider an approach based on introducing in-track drift and arriving at the docking port after a full planning horizon, with no other constraints. In this case, the planning horizon is a full orbit, with no initial radial offset and no initial velocity; an initial thrust in the in-track direction will introduce a secular drift into the relative orbit. Over the course of an orbit this secular drift causes the in-track position to shift by

$$\Delta y = \Delta v_y \frac{6\pi}{n} \quad (18)$$

Fig. 1 Target spacecraft and docking configuration.

Fig. 2 Radial/in-track view of rotating target spacecraft and docking configuration.

Fig. 3 Nominal trajectory planning with no safety: constraint violations occur for trajectory failures. The nominal trajectory is marked with $\bullet$ and the failure trajectories with $\times$. The failure trajectories all result in collisions with the target spacecraft.

Fig. 4 Trajectory planning with safety: failed trajectories deviate around the target spacecraft, preventing collision. The nominal trajectory is marked with $\bullet$ and the failure trajectories with $\times$.
Lagrange multipliers, \( \lambda \), is a row vector that extracts the scalar radial component, and \( H \) is a row vector that extracts the scalar in-track component. The trajectory that minimizes the 2-norm of the input vectors and meets those simple constraints is given by the pseudoinverse solution [30] of

\[
A_Q \begin{bmatrix} u_0 \\ \vdots \\ u_{k-1} \end{bmatrix} = b_Q
\]

(19)

with

\[
A_Q = \begin{bmatrix} H_1 \Gamma_1 & 0 & \cdots & 0 \\ H_2 \Gamma_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_N \Gamma_{N-1} & 0 & \cdots & H_N \Gamma_N \end{bmatrix}, \quad b_Q = \begin{bmatrix} -H_1 A_1 x_0 \\ -H_2 A_2 x_0 \\ \vdots \\ -H_N A_N x_0 + y_{des} \end{bmatrix}
\]

(20)

where \( H_i \) is a row vector that extracts the scalar radial component, \( H_j \) is a row vector that extracts the scalar in-track component, and \( y_{des} \) is the desired in-track component at step \( N \). This form has \( N + 1 \) constraints and \( 3N \) input variables to choose. The trajectory that minimizes the 2-norm of the input vectors and meets those simple constraints is obtained by the pseudoinverse solution [30] of

\[
\begin{bmatrix} I_{3N} & A_Q^T \\ A_Q & 0_{N+1} \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{k-1} \\ z \end{bmatrix} = \begin{bmatrix} 0_{3N,1} \\ b_Q \end{bmatrix}
\]

(21)

where \( z \) is a vector of \( N + 1 \) Lagrange multipliers, \( I_p \) is a \( p \times p \) identity matrix, \( 0_{q \times q} \) is a \( q \times q \) matrix of zeros, and \( 0_{p \times 1} \) is a \( p \times 1 \) vector of zeros. The fuel cost of the trajectory found using this method is 39.1 mm/s, which is very close to the optimized cost of following a strict V-bar trajectory.

B. Case 2: Docking Port Perpendicular to Spin Axis

The rotating docking port case uses an identical formulation to the stationary case, however, the constraint regions are time varying. In particular, in the stationary case, \( A_k = A_{k+1} \forall k = 1 \cdots N \), but to formulate the rotating problem, the \( A_k \) and \( b_k \) matrices must be formed for each step of the planning horizon based on the rotation rate and, for more general cases, the motion of the target, the docking port, and the line-of-sight cone. One simple way to generate these constraints is to represent each side of an avoidance region as a plane that is specified by a sample of its constituent points. These points will remain in a plane through any reorientation of the original constraint. Thus, the rotated constraint side can be found by applying rotation matrices [31] to the points and then forming the equation of a new plane, which can then be used as an inequality constraint. The translation and rotation motion of the target spacecraft should be well characterized through observation or cooperation before starting a rendezvous maneuver, thereby allowing the prediction of its future trajectory to be used for forming constraints. All trajectory propagations of the target spacecraft used to create constraints are formed before the rendezvous maneuver is optimized. As a result, the propagation can be carried out using any method appropriate for the specific online implementation (e.g., simple linear propagation, high accuracy numerical integration). Robustness to uncertainty in target motion can be accommodated by guaranteeing that any optimized trajectory is valid for a range of representative target initial conditions [19].

For the rotation case examined in this section, the optimized trajectory no longer matches a two-impulse V-bar approach, but is instead forced to thrust regularly to stay within the rotating LOS cone. Figure 5 shows this optimized trajectory with no safety constraints. As in the stationary case, in the absence of safety constraints, the nominal trajectory collides with the target in the event of guidance shutdowns. An alternate form with safety constraints prevents collisions for failures occurring in the last quarter of the trajectory (Fig. 6). Note that in Fig. 6, the safe trajectory appears to pass through the target, but in actuality it avoids collision because of the rotational motion of the target. The fuel costs without and with safety constraints are 55.3 and 56.4 mm/s, respectively. In this case, the fuel cost of imposing safety as a constraint is only a 2% increase over the nominal cost. As in the nonrotating case, the increase in fuel needed to include the safety constraints is minimal and the advantage is guaranteed collision avoidance for passive abort in the last quarter of the nominal path.

C. Case 3: Stationary Docking Along a Radial Trajectory

Another example (safe trajectory in Fig. 7) using a stationary target with an initial radial offset required 104 mm/s of fuel with and without safety constraints. The safety constraints in this case were not necessary for creating a safe trajectory, because R-bar approaches tend to naturally drift away from the target satellite. However, the addition of the safety constraints would be appropriate for a radial approach if the target satellite has additional avoidance constraints (e.g., solar panels).
V. Probability of Collision

To judge the effectiveness of the safety algorithm introduced in Sec. III, define a probability of the collision metric $P_{\text{col}}$, which is the probability of a failure at any time step during a maneuver resulting in a collision between the target and chaser spacecraft. The probability of collision is given by

$$P_{\text{col}} = \sum_{i=1}^{N} P(\text{failure at } i \mid \text{no failure before } i)$$

where the probability $P(\text{failure occurs} \mid \text{failure at } i)$ is either 1 or 0 and is evaluated by examining the trajectory followed if thrusters are disabled at step $i$ and checking for future collisions. Assuming that the probability of a failure at any step in the trajectory is $f$, then

$$P(\text{failure at } i \mid \text{no failure before } i) = (1-f)^{i-1}f$$

Using the metric $P_{\text{col}}$, the effectiveness of the safety approach was investigated by creating a series of safe trajectories starting from different initial conditions near the target. The initial condition positions were chosen to create a range of nearby starting points. The velocity vector for each position was chosen according to the conditions in [14] to create a safety ellipse. This creates a situation where each rendezvous trajectory begins from a safe, invariant orbit within range of a final approach rendezvous trajectory.

Figures 8 and 9 show the values of $P_{\text{col}}$ for full-orbit optimized final approach trajectories, discretized into $N=20$ steps. The trajectories are generated using $\mathcal{F} = \emptyset$ (no safety constraints) and $\mathcal{F} = \{9, \ldots, 19\}$ (guaranteed safe for the last 10 steps of the trajectory), respectively, where $f = 0.001$. These plots show that without safety, the probability of collision for a given rendezvous trajectory tends to fall between 0.005 and 0.015. However, the addition of safety for half of the trajectory brings the collision probability for most of the trajectories below 0.001. The same optimizations were performed and analyzed for a range of other $\mathcal{F}$ ranges and the results are summarized in Fig. 10. The dashed line indicates an overbound, $P_{\text{col}}$, for the maximum possible probability of collision, which is the case where every failure during the course of the trajectory when safety is not guaranteed (i.e., steps not in $\mathcal{F}$) would result in a collision, which is given by

$$\sum_{i=0}^{N-1} P(\text{failure at } i \mid \text{no failure before } i)$$

The line marked with ◆ shows the largest probability encountered in the optimized trajectories for all initial conditions considered. This is equivalent to finding the maximum height ($z$ value) in a plot of the type in Fig. 9 for each different set of $\mathcal{F}$ used to create Fig. 10. The minimum (line marked by ○) shows that in each case, there were some initial conditions that did not result in collision, regardless of the steps in $\mathcal{F}$. In those cases, the fuel-optimal rendezvous trajectory is safe. The average $P_{\text{col}}$ (solid line), equivalent to averaging the probability heights over an area of the type in Fig. 9, followed a similar trend to the largest $P_{\text{col}}$, but was significantly lower. This indicates that although some initial conditions are particularly prone to collision, on average the collision probabilities are significantly improved by safety and in no case has the addition of safety made collisions more likely than in the fuel-optimal case ($\mathcal{F} = \emptyset$). Furthermore, for this particular case, the trends indicate that guaranteeing more than the last five steps safe does not significantly decrease the probability of a collision. This conclusion would be valuable from a mission planning perspective, because each
additional plan step that is guaranteed safe represents a tradeoff in which computation time and nominal fuel use potentially increase.

Equation (24) indicates that the overbound $\bar{P}_{\text{col}}$ decreases with increasing length of the safe region (i.e., fewer steps in $\mathcal{F}$). For the purposes of worst-case safety guarantees, the overbound could be used as an analytic rule of thumb for mission design studies.

VI. Invariant Formulation

The safety formulation introduced in Sec. III only guarantees passive collision avoidance until the end of the safety horizon. In previous examples, the safety horizon has been fixed at one orbit. Figure 11 shows a stationary-target case where a collision would occur soon after the end of a one orbit safety horizon. If the safety horizon is extended to multiple orbits, the resulting failure trajectories will tend to either drift away from target spacecraft or create invariant orbits that neither drift toward nor away from the target spacecraft. Drifting away from the target orbit is preferable to collision; however, it means that fuel will need to be expended to bring the chaser near the target for any future docking attempts. Furthermore, the longer the controllers wait to cancel the drift, the farther apart the two spacecraft will become, thereby creating an additional timing consideration during an anomalous event. It is preferable for the chaser to drift into an invariant orbit that is near the target, but can never, under the assumptions of Keplerian dynamics, collide. The preference for invariant failure orbits can be captured by constraining a state in the failure trajectory at some step $k$ to be the same one full orbit after $k$ using a linear state transition matrix to propagate the state forward. This constraint is written as

$$x_{FT_s} = A^k_{FT} x_{FT_t} \text { for } k \geq T$$

where $N_o$ is the number of steps in an orbit. By imposing this constraint for all possible failure times in $\mathcal{F}$, all failure orbits are guaranteed to be invariant with respect to the target. The safety algorithm in Eq. (17) is only altered by the addition of the new constraints in Eq. (25).

Figure 12 shows the same rendezvous problem from Fig. 11, but with additional invariance constraints on failures occurring in the last quarter of the rendezvous trajectory. Imposing invariance constraints yields circular trajectories relative to the target, which are traversed once per orbit with no fuel expenditure. Some of the six failure trajectories shown in Fig. 12 overlap each other where the optimized trajectory did not require fuel inputs. The safe trajectory with no invariance constraint used 3.4 mm/s of fuel and the safe trajectory with invariance used 7.1 mm/s of fuel. In this case, the invariance constraints have roughly doubled fuel requirements. However, this tradeoff may be beneficial when there is a danger of collision after the safety horizon or when the fuel requirements for canceling drift after a failure are taken into account. The cost of using an optimized rendezvous trajectory with no safety constraints is 3.1 mm/s (versus 3.4 mm/s with safety), making safety roughly 10% more expensive. The cost of using a strict V-bar approach (59.2 mm/s) is roughly 18 times more expensive than the optimized safe approach and 8.6 times more expensive than the optimized safe invariance approach. Thus, when compared to the cost of using a standard nonoptimized maneuver, the fuel premium for using safety with invariance is small.

The examples using invariance constraints presented in this section used Hill’s dynamics [20], meaning that the particular type of invariance achieved is effectively equivalent to constraining the relative semimajor axes of the target and chaser spacecraft to be zero. Alternate equations of motion exist which model additional orbital perturbations. For example, the dynamics presented in [32] could be used for rendezvous and docking trajectory generation to create invariant orbits that account for the effects of differential $J_2$. Likewise, the effects of differential drag can also be taken into account using the method in [19].

VII. Convex Safety Formulation

The safety constraints introduced in Sec. III guaranteed that the chaser states would not collide with the target in the event of a failure. The collision avoidance in those constraints is accomplished using binary variables to capture the nonconvexity of the problem. The problem with binaries was formulated as a mixed-integer linear program (MILP) and posed to a commercial solver. Solving a MILP can be a computationally intensive task and the time required to solve tends to grow very quickly with the number of discrete variables in the problem [33]. The trajectory shown in Fig. 4 required 8.92 s to solve on a 3-GHz computer. That problem had a 20-step safety horizon, an avoidance region with eight sides, and six inputs that were safe (i.e., the last quarter of the trajectory) in the event of guidance shutdown. Each avoidance region side requires a binary variable at each step of the safety horizon and those constraints are included 6 times, each propagating forward from a different failed thruster step. Thus, implementing collision avoidance over the safety horizon for that simple example required 960 binary variables. Solving the same trajectory for a two orbit safety horizon required 24.6 s. Using the same horizon duration with a finer discretization step would further increase the required computation time. It is likely that online implementations would need to be solved with limited computer resources and that a nonconvex implementation may be impractical for implementations requiring short discretization steps.

An alternative to the nonconvex formulation is to use a more restrictive form of collision avoidance that is convex. Instead of requiring the chaser to remain outside an avoidance region, the
failure trajectories are instead constrained to remain inside a region that is known to not contain the target. This is similar to the type of convex passively safe trajectory examined for rotating satellite capture in [16]; however, the approach in this paper explicitly minimizes fuel use. Figure 13 shows an example of the optimized trajectory for the rendezvous problem in Fig. 4, but instead solved as a LP using the convex safety constraints

\[ H_k x_{FT_k} \geq y_{\text{min}} \quad \forall \ k \in \{T + 1, \ldots, N + S\} \] (26)

where \( y_{\text{min}} \) is the maximum in-track position of the target spacecraft. The nonconvex case results in failure trajectories that are not permitted in the convex case and as a result, there is a fuel penalty for imposing convexity. Using the convex formulation in Eq. (26) does not require altering the basic safety algorithm in Eq. (17). Instead, only the target geometry, given by the set \( T \) in Eq. (14), needs to be altered to ensure that its complement is convex. The nonconvex case requires 1.5 mm/s of \( \Delta V \) and the convex case requires 3.7 mm/s. The more restrictive area in which failure trajectories can lie caused the required fuel to increase by more than a factor of 2. However, the amount of time required to compute the convex trajectory was only 0.06 s, a decrease from the nonconvex case by a factor of 150. In cases where it is impractical to dedicate significant computational resources to planning, it may be desirable to trade the fuel optimality of the more general MILP formulation for the speed of the LP formulation. In addition, the convex solution is often similar to the invariant given in Eq. (25). For the example in Fig. 13, the trajectory cost when invariance and convexity are imposed is 4.2 mm/s, which is the same as the cost of using invariance in the fully nonconvex problem.

VIII. Active Safety

An alternative to passive safety is active safety, in which a set of thruster inputs is applied to ensure rendezvous safety. The active response is a set of input sequences that is used instead of passive safety. The safe input sequence can be designed a priori (e.g., thrust in-track, thrust radially) and chosen in real time or optimized at the time the nominal rendezvous maneuver is optimized. In either case, the safe inputs are known at all times during the maneuver and no additional optimization is required in the event of a failure. The advantages of active safety over passive safety are significant: by allowing thrusting in the event of a failure, a significantly larger portion of the nominal trajectory can be guaranteed safe and the fuel costs of guaranteeing the safety of the nominal trajectory are reduced. Passively safe trajectories can be considered a subset of active safe trajectories in which the active input sequence has no thrusting. The primary limitation of active safety is that it provides safety guarantees for a smaller set of possible system malfunctions than passive safety. In the case of passive safety, any anomaly in which the thrusters can be disabled can be made safe. Safety guarantees resulting from an active safety trajectory require that some thrusters continue to work properly and in the correct directions in the event of a failure. An extension at the end of this section will show how active safety can be modified to provide safety guarantees for single thruster failures.

To create an active safety constraint, the optimization from Sec. III is altered to allow the possibility of using a safe input sequence by introducing an additional discrete convolution matrix. Rewriting Eq. (12) for a predetermined safe input sequence

\[ x_{FT_k} = \begin{bmatrix} A_{d,T}^{-1}B_d & A_{d,T}^{-2}B_d & \cdots & A_{d,T}^{-1}B_d & 0 & \cdots & A_dB_d & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{k-1} \end{bmatrix} + A_d^k x_0 \] (27)

where \( T < k < N, k - T < N, N \) is the number of steps in the nominal plan, and \( N \) is the number of steps in the safe input sequence. If \( k \leq T \), then \( x_{FT_k} = x_k \), because no potential failure could have occurred at that time. Equation (27) can be written more generally as

\[ x_{FT_k} = \begin{cases} \Gamma_k S(T, k) U_k + A_d^k x_0 + \Gamma_{k-T} S(k - T, N) V_k, & T \leq k \leq N, T < k - T \leq N \\ A_d^{k-N} \Gamma_N S(T, N) U_N + A_d^k x_0 + \Gamma_{k-T} S(k - T, N) V_k, & k > N, k - T < N \\ \Gamma_k S(T, k) U_k + A_d^k x_0 + A_d^{k-N} \Gamma_N V_N, & k < N, k - T < N \\ A_d^{k-N} \Gamma_N S(T, N) U_N + A_d^k x_0 + A_d^{k-N} \Gamma_N V_N, & k > N, k - T > N \end{cases} \] (28)

where \( k \) is the time step that the failure trajectory is propagated forward to, \( S(q, N) = \text{diag}(I_{3m}, 0_{(N-k)} \cdot I) \), \( I \) is an \( n \times n \) identity matrix, \( 0_n \) is an \( n \times n \) matrix of zeros, the decision variables for the nominal input are the vector \( U_k = [u_0, \ldots, u_{k-1}] \) and the predetermined safe input sequence \( V_k = [v_0, \ldots, v_{k-1}] \). The possible ranges in Eq. (28) correspond to the steps before the nominal plan has ended and before the end of the safe input sequence (\( k \leq N, k \leq k - T \)), the times after the nominal plan has ended and before the end of the safe input sequence (\( k > N, k \leq k - T \)), times before the
nominal plan has ended and after the end of the safe input sequence \(k \leq N, k > k - T\), and the times after the both the nominal plan and the safe input sequence have ended \(k > N, k > k - T\). All four cases must be considered to allow for safe input sequences that are longer or shorter than the nominal plan length. Active safety can be guaranteed by introducing the set of constraints

\[
\ldots x_{FT} \neq T_k \quad \forall k \in \{T + 1, \ldots, N + S\}
\]

The set of constraints in Eq. (29) is applied for each step \(T\) at which safety should be guaranteed in the event of a failure.

An alternate approach to active safety where the safe input sequence is optimized online can be implemented by moving the safe input sequence \(V_{N1}\) into the decision vector of Eq. (28) such that the safe input sequence \(V_{N1}\) is optimized at the same time as the nominal rendezvous trajectory. The weighting on the fuel inputs in \(V_{N1}\) relative to those in \(U_{N}\) is made small to minimize the fuel required for the more likely nominal case. Active safety uses the same safety algorithm in Eq. (17), but with Eq. (14) replaced by Eq. (29) using the active failure trajectory given by Eq. (28) for a priori known safe input sequences or Eq. (30) for safe input sequences optimized online.

The implementation of an active safe trajectory would be similar to that of a safe trajectory. Before entering the trajectory, the spacecraft is assumed to be in a nominal state (i.e., all systems are functioning correctly). If a fault has not yet occurred, the spacecraft follows the nominal trajectory, which is given by \(U_{N}\). If a fault occurs during a step that has been guaranteed to be safe in the event of that fault, then the spacecraft begins using the safe input sequence. For the duration of the safe input sequence, the chaser and target spacecraft are guaranteed to not collide. If the invariance constraints in Sec. VI are used, safety can be guaranteed for any time horizon over which the dynamics are valid.

\[
x_{FT} = \begin{cases}
  [\Gamma S(T, k) \Gamma_{k-T} S(k - T, N)]^{-1} U_{N} \quad & T \leq k \leq N, T < k - T \leq N_s \\
  [A_{S}^{T-N} \Gamma_{N} S(T, N) \Gamma_{k-T} S(k - T, N)]^{-1} U_{N} \quad & k > N, k - T \leq N_s \\
  [\Gamma S(T, k) \Gamma_{k-T} S(k - T, N)]^{-1} U_{N} \quad & k \leq N, k - T > N_s \\
  [A_{S}^{T-N} \Gamma_{N} S(T, N) \Gamma_{k-T} S(k - T, N)]^{-1} U_{N} \quad & k > N, k - T > N_s
\end{cases}
\]

A. Examples

A stochastic analysis of the type performed in Sec. V was conducted using active safety, guaranteeing safety for the last three-fourths of the nominal trajectory \((F = [9, \ldots, 19])\). The results indicated that the average collision probability for failures accounted for by active safety was reduced to \(1.96 \times 10^{-3}\) from \(0.0057\) for the optimal unsafe case and \(4 \times 10^{-4}\) for the passive safety case \((F = [4, \ldots, 19])\). The differences between the active safety approaches are demonstrated in Figs. 14–17. Figure 14 shows an active safe rendezvous trajectory beginning from a safety ellipse holding orbit. In this case, the safe input sequence \(V\) has been arbitrarily chosen to be an orbit of constant thrusting at \(10^{-6}\) m/s\(^2\) in the \(-x\) direction of an LVLH frame centered on target. The last three-quarters of the rendezvous trajectory have been guaranteed through constraints to be actively safe. In the figure, the nominal rendezvous trajectory (line marked with •) shows the planned rendezvous maneuver which will be followed if no failures occur. Each portion of the trajectory marked with △ shows a possible path followed by the chaser in the event that the safe input sequence is used. Constraints guarantee safe collision avoidance for the entire red portion of the trajectory; however, no safety guarantees exist for the trajectory after the safe input sequence is enacted. The trajectories marked by × show how the path drifts after the end of each safe trajectory. In several cases, the drifting path would result in a collision at some time in the future. To ensure collision avoidance, Fig. 16 shows an active safe trajectory optimized from the same initial conditions, but with the invariance constraint from Eq. (25) imposed. In this case, a failure at any step in the last three-quarters of the nominal trajectory would result in the chaser spacecraft entering a safe, invariant trajectory near the target spacecraft. Figures 16 and 17 show the optimized active safe trajectories using the constraints in Eq. (30) without and with invariance constraints, respectively.

Table 1 compares various approaches for creating rendezvous trajectories with and without safety using the same initial conditions as the examples in Figs. 14–17. The first row refers to the fuel cost (mm/s) of implementing the nominal rendezvous trajectory. The second row gives the cost of implementing a full safe input sequence (mm/s). The last row gives the probability of collision for the trajectory using the method introduced in Sec. V. The columns compare the fuel-optimal path with no safety to the passive safety path and paths using active safety. The active safety columns labeled a priori use the predefined safe input sequence approach in Eq. (28) and the columns labeled optimized use the approach in Eq. (30). The columns marked invariant also use the invariance constraints in Eq. (25). It is notable that for the example in the table, the probability of collision for the fuel-optimal trajectory (i.e., no safety) is 0.012, but the addition of passive or active safety to the problem causes the probability to drop to zero. Note that this probability is predicated on the assumption that a failure is identified within a time step of its

![Fig. 14 Rendezvous trajectories using active safety.](image-url)
occurrence and that the thrusters can be turned off (for passive safety) or used nominally (for active safety).

Passive safety requires more fuel for rendezvous than the case without safety; however, active safety with an optimized safe input sequence has the same cost as the fuel-optimal case. In the cases where invariance is imposed as a constraint, the fuel cost using active optimized safety is lower than the passive invariance case, but not as low as the optimal trajectory. Thus, for these initial conditions, both the nominal trajectory and the safe abort trajectory must be shaped to achieve active invariance. The cost of safety for the nominal and passive safety trajectories is zero, because those cases do not consider safety and do not require thrusting for failures, respectively. In each active safety case, the safety cost is very small compared to the cost of the nominal trajectory, indicating that it should be possible to implement active safety on a space mission without significantly increasing the $\Delta V$ budget.

B. Active Safety for Thruster Failures

The active safety approach in Eq. (30) can be modified to guarantee safety for cases of individual thruster failure by optimizing multiple safe input sequences. Each safe input sequence is constrained to only use a single thruster direction, or alternately, a single thruster assuming that thrusters act through the center of gravity. This guarantees that if only one thruster fails, another safe trajectory which does not use the failed thruster still exists. Thus, in a system with at least two thrusters, any single thruster failure to the off state will be in the set of possible system failures covered by active safety. Likewise, if thrusters in the system can be used to cancel each other (e.g., a system with axial thrusters), then this active safety extension can also be used in the presence of thruster-on failures. In that case, the thruster opposite that which failed can be used to cancel erroneous thrusting, while a thruster in another direction can be used to enact a preplanned safe input sequence.
Modifying Eq. (30) to include multiple safe input sequences yields

\[
x_{RTi} = \begin{bmatrix}
\Gamma_i S(T) & \Gamma_i S(T - k) H_i & 0 & 0 \\
\Gamma_i S(T) & 0 & \Gamma_i S(T - k) H_i & 0 \\
A_{d_i}^{k} \Gamma_i S(T) & 0 & 0 & 0 \\
A_{d_i}^{k} \Gamma_i S(T) & 0 & 0 & 0 \\
\end{bmatrix} \begin{bmatrix}
U_k \\
V_k \\
V_k' \\
V_k'' \\
\end{bmatrix} + A_{d_i}^{k} x_{0_i}, \quad k \leq N, k - T \leq N_s \\
\begin{bmatrix}
\Gamma_i S(T) & 0 & 0 & 0 \\
\Gamma_i S(T) & 0 & 0 & 0 \\
A_{d_i}^{k} \Gamma_i S(T) & 0 & 0 & 0 \\
A_{d_i}^{k} \Gamma_i S(T) & 0 & 0 & 0 \\
\end{bmatrix} \begin{bmatrix}
U_k \\
V_k \\
V_k' \\
V_k'' \\
\end{bmatrix} + A_{d_i}^{k} x_{0_i}, \quad k > N, k - T \leq N_s \\
\begin{bmatrix}
\Gamma_i S(T) & 0 & 0 & 0 \\
\Gamma_i S(T) & 0 & 0 & 0 \\
A_{d_i}^{k} \Gamma_i S(T) & 0 & 0 & 0 \\
A_{d_i}^{k} \Gamma_i S(T) & 0 & 0 & 0 \\
\end{bmatrix} \begin{bmatrix}
U_k \\
V_k \\
V_k' \\
V_k'' \\
\end{bmatrix} + A_{d_i}^{k} x_{0_i}, \quad k > N, k - T > N_s \\
\end{equation}

where \( V_{x_i} \) is the safe input sequence of only \( x \)-direction inputs, \( V_{y_i} \) is the safe input sequence of only \( y \)-direction inputs, and \( H_i \) and \( H_s \) are matrices that extract only elements of \( \Gamma \) pertaining to \( u_x \) and \( u_y \) inputs, respectively. The active safety algorithm remains the same, but the failure trajectory used in formulating Eq. (29) must be propagated using Eq. (30) instead of Eqs. (28) and (30).

Figure 18 shows an example trajectory using the multisolution active safety form in Eq. (31) to solve the safe rendezvous problem for the initial conditions used in Table 1. The left side of the figure shows the nominal rendezvous trajectory and the safe trajectories that would be used in the event of a failure in the \( \pm y \) direction thrust (resulting from using \( V_{y_i} \)). The right side shows the same nominal trajectory, but the safe trajectories shown correspond to \( V_{x_i} \). In this case, a single optimization has produced two sets of safe input sequences, either valid at any time step with guaranteed safety. The safe input sequence solutions are

\[
V_{x_i} = \begin{bmatrix} -4.56 \times 10^{-6} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad V_{y_i} = \begin{bmatrix} -3.06 \times 10^{-6} \\ 0 \\ 0 \\ 0 \end{bmatrix}
\]

The algorithm for using passive safety only requires that thrusters be disabled in the event of a failure and active safety only requires that a predetermined safe input sequence be used. The implementation algorithm for the modified active safety formulation in this section requires an additional input from the spacecraft fault detection and isolation system which indicates the type of fault. In the case of thruster failure, this would also need to include which thruster failed and the nature of the failure. This additional information enables the active safety implementation to choose the appropriate safe input sequence to use.

**IX. Conclusions**

Safety in autonomous spacecraft rendezvous trajectory design allows abort with guaranteed collision avoidance for a class of anomalous system behaviors. This paper introduced several online optimization formulations that guarantee passive and active safety and demonstrated in numerous simulations that the additional fuel costs are comparatively small, particularly relative to commonly considered suboptimal trajectories. Additional restrictions to guarantee failure trajectories that minimize drift and guarantee long-horizon passive collision avoidance were shown to require fuel use on the same order of magnitude as optimized solutions that do not include safety guarantees. Approximate analytic methods for creating upper and lower bounds on the expected fuel use for several mission types yielded accurate estimates compared to optimized fuel costs. A convex formulation of the safety problem was introduced...
which uses approximately twice as much fuel, but more than 150 times less computation time than the nonconvex formulation. An active safety approach was developed and shown to be capable of achieving the same fuel costs as trajectories without safety while still guaranteeing collision-free escape trajectories for a large class of potential anomalies, including single thruster failures. The safety algorithms presented provide a fuel-efficient, computationally feasible framework for designing safe-mode procedures for multispacecraft missions.

Acknowledgments

This work was funded under Cooperative Agreement NCCS-729 through the Goddard Space Flight Center Formation Flying NASA Research Announcement. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References


Control of Spacecraft in Proximity Orbits

by

Louis Scott Breger

Bachelor of Science Aeronautics and Astronautics
Massachusetts Institute of Technology, 2002

Master of Science Aeronautics and Astronautics
Massachusetts Institute of Technology, 2004

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2007

© Louis Scott Breger, MMVII. All rights reserved.

The author hereby grants to MIT permission to reproduce and
distribute publicly paper and electronic copies of this thesis document
in whole or in part.

Author ..........................................................
Department of Aeronautics and Astronautics
June 8, 2007

Certified by ..........................................................
Jonathan P. How
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ......................................................
Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students
Control of Spacecraft in Proximity Orbits

by

Louis Scott Breger

Submitted to the Department of Aeronautics and Astronautics on June 8, 2007, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

Formation flying of spacecraft and autonomous rendezvous and docking of spacecraft are two missions in which satellites operate in close proximity and their relative trajectories are critically important. Both classes of missions rely on accurate dynamics models for fuel minimization and observance of strict constraints for preventing collisions and achieving mission objectives. This thesis presents improvements to spacecraft dynamics modeling, orbit initialization procedures, and failsafe trajectory design that improve the feasibility and chances of success for future proximity operations. This includes the derivation of a new set of relative linearized orbital dynamics incorporating the effects of Earth’s oblateness. These dynamics are embedded in a model predictive controller, enabling LP-based MPC formulations for large baseline formations in highly elliptic orbits. An initialization algorithm is developed that uses the new dynamics to optimize multiple objectives (drift and fuel usage minimization, geometry) over science-relevant time frames, improving previous $J_2$-invariant initialization techniques which only considered infinite-horizon secular drift. The trajectory planning algorithm is used to design spacecraft rendezvous paths that observe realistic constraints on thruster usage and approach path. The paths are fuel-optimized and further constrained to be safe (i.e., avoid collisions) in the presence of many possible system failures, an enhancement over previous guaranteed-safe rendezvous methods, which did not minimize fuel use. The fuel costs of imposing safety as a constraint on trajectory design are determined to be low compared to standard approaches and a stochastic analysis demonstrates that both active and passive forms of the safe rendezvous algorithm substantially decrease the likelihood of system failures resulting in collisions. The effectiveness of the new controller/dynamics combination is demonstrated in high fidelity multi-week simulations. An optimized safe rendezvous trajectory was demonstrated on a hardware testbed aboard the International Space Station.

Thesis Supervisor: Jonathan P. How
Title: Associate Professor of Aeronautics and Astronautics
Acknowledgments
I would like to thank everyone who has assisted me in my academic career. My parents and sister have supported me in every way and taught me the most important principles of graduate work long before my arrival at MIT. A complete list of everyone who’s had a hand in this thesis would include my teachers through MIT, high school, and even earlier. I’ve been fortunate to have had access to an exceptional education and many good people to steer me along the way. In particular, I’d like to thank my first control systems instructors: Winston Markey, John Deyst, and Emilio Frazzoli. The semester-long introductory class they taught in automatic control in the Fall of 2000 changed the course of my life and I am grateful for having had the opportunity to study under them. Likewise, Lorraine Fesq helped me immeasurably in my quest to enter graduate school and find work afterward by providing me with many of my most treasured job experiences. Terry Alfriend’s generous donations of time, expertise, and renowned intuition have enabled much of this research. The SPHERES team (in particular, Simon Nolet, Nick Hoff, and Chris Mandy) deserve many thanks for the time and effort they put into helping me get experiments working on their hardware and on ISS. My fellow labmates (notably Yoshiaki Kuwata, Arthur Richards, and Luca Bertuccelli) have always been around to bounce ideas off, study with, and even fly airplanes alongside; their companionship will be greatly missed. Kathryn Fischer’s help navigating the administrative side of MIT has been indispensable. Finally, Jonathan How, who has been my graduate and research advisor for the past five years, has taught me far far more than I ever thought I had to learn. Jon’s patience, wisdom, funding, and humor were necessary conditions for producing the contributions that follow. This thesis is dedicated to my parents, for whom the term acknowledgement seems inadequate. Their encouragement and example are my principal inspirations. This work was funded under Cooperative Agreement NCC5-729 through the NASA GSFC Formation Flying NASA Research Announcement. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration.
2.4 Model Predictive Control Using GVEs  50
    2.4.1 Error-Box Constraints Using Relative Orbital Elements  53
    2.4.2 Formation Flying: Coordination Using GVEs  55
2.5 Comparison to Other GVE-based Impulsive Control Schemes  57
2.6 Formation Maintenance on MMS-like Mission  60
2.7 Summary  63

3 Fuel-optimized Semi-$J_2$-invariant Initial Conditions  65
    3.1 Formulation  66
    3.2 Results  69
    3.3 Summary  75

4 Formation Flying Simulations  77
    4.1 Introduction  77
    4.2 Mission Description  77
    4.3 Simulation Controller Configuration  79
        4.3.1 Parameters Examined  80
    4.4 Simulation Results  82
    4.5 Summary  86

5 Safe Trajectories for Autonomous
   Rendezvous of Spacecraft  105
    5.1 Introduction  105
    5.2 Online Trajectory Optimization for Autonomous Rendezvous and Docking  107
    5.3 Safety Formulation  110
    5.4 Scenarios  112
        5.4.1 Case 1: Stationary Target Satellite  114
        5.4.2 Case 2: Docking Port Perpendicular to Spin Axis  118
    5.5 Probability of Collision  119
    5.6 Invariant Formulation  123
5.7 Convex Safety Formulation ........................................... 126
5.8 Active Safety .......................................................... 129
  5.8.1 Examples .......................................................... 133
  5.8.2 Active Safety for Thruster Failures ............................. 136
  5.8.3 Mitigating Impact of Process Noise and Navigation Error .. 140
5.9 Summary ............................................................... 142

6 Safe Docking Demonstrations on SPHERES .......................... 145
  6.1 SPHERES Background ................................................ 145
  6.2 Safe Autonomous Rendezvous and Docking on the SPHERES Testbed 146
  6.3 Flight Experiment Results .......................................... 151
  6.4 Flat Table Experiment Results .................................... 157
  6.5 Summary ............................................................... 157

7 Conclusions and Future Work .......................................... 161
  7.1 Conclusions .......................................................... 161
    7.1.1 Linearized Relative Dynamics ................................. 161
    7.1.2 Optimized Initialization of Semi-Invariant Orbits .......... 162
    7.1.3 Control of Spacecraft Formations Using MPC ............... 162
    7.1.4 Safety in Autonomous Rendezvous and Docking ............. 163
    7.1.5 Safe Autonomous Rendezvous Demonstration on Orbit ....... 164
  7.2 Future Extensions .................................................. 164
List of Figures

2-1 Effect of Orbital Element Perturbations on the $\Delta B_{\text{true}}$ Matrix for a LEO Orbit .............................................. 40
2-2 Effect of Orbital Element Perturbations on the $\Delta B_{\text{true}}$ Matrix for a HEO Orbit .............................................. 40
2-3 Difference between integrated and approximated $\Gamma$ for different discretization times using the LEO orbit in Eq. (2.14) ............... 49
2-4 Difference between integrated and approximated $\Gamma$ for different discretization times using the HEO orbit in Eq. (2.16) ............... 50
2-5 Example of a plan generated to MPC with $J_2$-modified GVEs (lines indicate relative state error) ........................................ 52
2-6 Error in open-loop trajectory following using the HEO orbit example. Lines indicate the difference between planned and implemented trajectories in a fully nonlinear simulation. ................................. 54
2-7 Forming and maintaining a 1000 km (at apogee) tetrahedron formation in a highly eccentric orbit ($e \approx 0.8$) in the presence of $J_2$ .... 61
2-8 Forming and maintaining a 1000 km (at apogee) tetrahedron formation in a highly eccentric orbit ($e \approx 0.8$) in the presence of $J_2$ .... 62
3-1 Illustration of Initialization Approach .................................................. 68
3-2 Expected drift and fuel cost of a range of optimized initial conditions in a LEO orbit ............................................................... 69
3-3 Expected drift and geometry cost of a range of optimized initial conditions in a LEO orbit ............................................................... 70
3-4 Expected drift due to differential $J_2$ effects and fuel cost of a range of optimized initial conditions in a HEO orbit.  ........................................ 73
3-5 Expected drift due to differential $J_2$ effects and geometry cost of a range of optimized initial conditions in a HEO orbit.  ......................... 73
3-6 Relative satellite separations for a vehicle initialized using analytic invariance and a vehicle using optimized invariance. Trajectories generated using realistic nonlinear simulation including the effects $J_2$.  ......................... 74
3-7 Relative satellite separations for a vehicle initialized using analytic invariance and a vehicle using optimized invariance. Trajectories generated using a nonlinear simulation that did not incorporate the effects $J_2$.  ........................................ 74
4-1 Concept for MELCO formation flying mission [113]  ......................... 78
4-2 Summary of Station-keeping Results from Table 4.1  ......................... 84
4-3 Simulation #1: Stationkeeping in a passive aperture formation (50 m) 87
4-4 Simulation #1: Stationkeeping in a passive aperture formation (500 m) ........................................ 88
4-5 Simulation #1: Stationkeeping in a passive aperture formation (5000 m) ........................................ 89
4-6 Simulation #2: Stationkeeping in a passive aperture formation (50 m) 90
4-7 Simulation #2: Stationkeeping in a passive aperture formation (500 m) ........................................ 91
4-8 Simulation #2: Stationkeeping in a passive aperture formation (5000 m) ........................................ 92
4-9 Simulation #3: Stationkeeping in a passive aperture formation (50 m) 93
4-10 Simulation #3: Stationkeeping in a passive aperture formation (500 m) ........................................ 94
4-11 Simulation #3: Stationkeeping in a passive aperture formation (5000 m) ........................................ 95
4-12 Simulation #4: Stationkeeping in a passive aperture formation (50 m) 96
4-13 Simulation #4: Stationkeeping in a passive aperture formation (500 m)
4-14 Simulation #4: Stationkeeping in a passive aperture formation (5000 m)
4-15 Simulation #5: Stationkeeping in a passive aperture formation (50 m)
4-16 Simulation #5: Stationkeeping in a passive aperture formation (500 m)
4-17 Simulation #5: Stationkeeping in a passive aperture formation (5000 m)
4-18 Simulation #6: Stationkeeping in a passive aperture formation (50 m)
4-19 Simulation #6: Stationkeeping in a passive aperture formation (500 m)
4-20 Simulation #6: Stationkeeping in a passive aperture formation (5000 m)

5-1 Target spacecraft and docking configuration
5-2 Radial/in-track view of rotating target spacecraft and docking configuration
5-3 Nominal trajectory planning with no safety: constraint violations occur for trajectory failures.
5-4 Trajectory planning with safety: failed trajectories deviate around the target spacecraft, preventing collision.
5-5 Nominal trajectory planning with no safety in the rotating case: constraint violations occur for trajectory failures.
5-6 Trajectory planning with safety in the rotating case: failed trajectories deviate around the target spacecraft, preventing collision
5-7 Trajectory planning with safety in the fast rotating (3/2 orbital speed) case.
5-8 Probability of a collision occurring for a range of initial conditions with $F = \emptyset$ (No safety guarantees).
5-9 Probability of a collision occurring for a range of initial conditions with $\mathcal{F} = \{9, \ldots, 19\}$ (latter half of trajectory guaranteed safe) ... 124

5-10 Probability of collision occurring for various values of $\mathcal{F}$ ....... 125

5-11 Case where end of safety horizon is followed by a collision. ....... 127

5-12 Use of invariance constraints guarantees infinite horizon passive collision avoidance and prevents failure trajectories from drifting away from the target. .................................................. 127

5-13 Collision avoidance for failure trajectories using convex constraints indicated by arrows. .................................................. 129

5-14 Active safety demonstrated for a range of initial conditions with $\mathcal{F} = \{4, \ldots, 19\}$ (latter 3/4 of trajectory guaranteed safe) ......... 134

5-15 Rendezvous trajectories using active safety. ......................... 138

5-16 Rendezvous trajectories using active safety with invariance constraints. 139

5-17 Rendezvous trajectories using optimized active safety. .......... 139

5-18 Rendezvous trajectories using optimized active safety with invariance constraints. .................................................. 141

5-19 Rendezvous trajectories for using active safety optimized for two possible thruster failure directions. .......................... 141

5-20 Safe rendezvous trajectory with robustness to initial condition velocity uncertainty .................................................. 142

6-1 SPHERES Microsatellite in air carriage on flat table testbed ....... 146

6-2 Planned safe trajectory for stationary rendezvous: $x$-$y$ view. Note that trajectory $z$ offset is zero at all times ............... 149

6-3 SPHERES Microsatellite in air carriage on flat table testbed: Perspective view .................................................. 149

6-4 Planned safe trajectory for rotating rendezvous: $x$-$y$ view. Note that trajectory $z$ offset is zero at all times ..................... 150
6-5 Planned safe trajectory for rotating rendezvous: testbed: Perspective view
6-6 SPHERES satellites docked together aboard the International Space Station using a safe optimized rendezvous trajectory.
6-7 Position trajectory followed during nominal docking ISS experiment: x-y view
6-8 Position time-series for z axis during nominal docking ISS experiment
6-9 Velocity time-series view during nominal docking ISS experiment
6-10 Velocity trajectory followed during nominal docking ISS experiment: x-y view
6-11 Position trajectory followed during failed docking ISS experiment: x-y view
6-12 Position time-series for z axis during failed docking ISS experiment
6-13 Velocity time-series view during failed docking ISS experiment
6-14 Velocity trajectory followed during failed docking ISS experiment: x-y view
6-15 Position trajectory followed during failed docking flat table experiment: x-y view
6-16 Position time-series for z axis during failed docking flat table experiment
6-17 Velocity time-series view during failed docking flat table experiment
6-18 Velocity trajectory followed during failed docking flat table experiment: x-y view
7-1 Fuel Required to Maintain an In-track Formation
7-2 Fuel Required to Maintain a Passive Aperture Formation
7-3 Maneuver costs: plans created accounting for eccentricity and $J_2$ effects
7-4 Maneuver costs: plans created accounting for eccentricity with no $J_2$ effects modeled
7-5 Maneuver costs: plans created accounting for eccentricity with no $J_2$ effects modeled \hfill 173

7-6 Maneuver costs: plans created not accounting for eccentricity with no $J_2$ effects modeled \hfill 174
List of Tables

1.1 Relative orbital dynamics indexed by regime of validity [115] . . . . 25

2.1 Average durations (in seconds) required to compute $\Gamma$ matrices using various methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

4.1 Fuel use results for formation flying simulations . . . . . . . . . . . . 83

5.1 Comparison of various types of safe rendezvous trajectories. . . . . . 135

7.1 Table of maneuvers planned using combinations of $J_2$ and eccentricity 170
Chapter 1

Introduction

Many future space missions will require autonomous proximity operations in which the knowledge and control of the relative state between space vehicles is critically important [1, 54]. For example, formation flying satellites operating in close proximity to accomplish coupled goals will require high levels of on-orbit autonomy and coordinated control [65]. Rendezvous and docking missions are also inherently concerned with controlling the reduction of the distance between spacecraft. Both types of spacecraft proximity missions share common characteristics and control requirements that include: similar proposed sensing technologies [1] (CDGPS, inter-satellite ranging), critical dependence on fuel minimization to ensure mission feasibility [84], the need to prevent relative drift between vehicles [73], use of relative orbital dynamics for the control design [65], concerns of collision avoidance between vehicles [61], and complicated multi-vehicle safe mode considerations [69, 79]. This thesis develops several new control technologies, analyzes their performance, and demonstrates their potential for improving the feasibility and safety of future spacecraft proximity operations.

Satellite formation flying missions will use coordinated observations between space vehicles to increase the resolution of the science data or achieve faster ground-track repeats [1]. For example, formation flying will be critical for creating large sparse-aperture optical and X-ray telescopes for space science and synthetic aperture radars for earth mapping. As discussed in [26], formation flying combines many component
technologies, such as distributed relative navigation, autonomous control, and distributed fault-protection. This thesis contributes to formation flying control systems in two significant ways: 1) a new linear dynamics model is introduced that extends the range of missions that can be controlled using linear control formulations; and 2) an optimization-based method for finding initial conditions that balances the need to reduce relative drift between satellites against the goals of minimizing fuel use and maintaining desired geometries. The new dynamics are embedded in a model predictive controller and demonstrated in realistic simulation environments.

Autonomous spacecraft rendezvous is an enabling technology for many future space missions [54]. Autonomous rendezvous has been used for docking with Mir [55], and more recently on the ETS-VII [56] and DART [57, 60] missions. However, anomalies occurred during both of these last two missions. In the case of ETS-VII, multiple anomalies caused entries into safe mode over the course of the mission, at least one of which resulted in a preprogrammed maneuver to move the spacecraft 2.5 km from its target. The anomaly in the DART mission is thought to have resulted in excess fuel expenditures and appears to have caused an on-orbit collision [58–60]. These recent experiences suggest that autonomous rendezvous and docking would greatly benefit from the inclusion of additional safeguards to protect the vehicles in the event of failures. Designing approach trajectories that guarantee collision avoidance for some common failures could simultaneously decrease the likelihood of catastrophic failures in which one, or both, of the spacecraft are damaged and increase the likelihood that future attempts at docking succeed. This thesis introduces a method for generating fuel-optimized rendezvous trajectories online that are safe with respect to a large class of possible spacecraft anomalies and demonstrates such a trajectory on a hardware testbed aboard the International Space Station.
1.1 Background

1.1.1 Formation Flying Control Systems

Formation flying spacecraft pose several control challenges beyond the problem of controlling a monolithic spacecraft or a constellation [5–7, 64]. In a typical single-spacecraft mission, the term control would refer to maintaining and altering the attitude of the spacecraft, whereas guidance would encompass the maintenance and manipulation of the trajectory on the scale of an orbit. After launch and initial correcting maneuvers, adjusting a spacecraft’s orbit would be an occasional activity planned from the ground. A constellation of spacecraft is operated much the same way [22, 23], because the constituent spacecraft operate in widely spaced orbits, with short-term decoupled performance objectives. A formation of spacecraft is defined by the need for inter-satellite control cooperation [65]. The satellites in a formation are typically represented as sharing a common reference orbit, that is, being close enough in terms of their position and velocity in a central body frame that their long-term, large-scale motion can be modeled using the dynamics of a single orbit. This proximity, while typical for rendezvous missions, is uncommon for satellite missions where there is an expectation for long-term collision-free operation. Formation flying is expected to require a level of autonomous onboard guidance that in most applications would be classified as automatic control [3, 8, 9, 65].

Many formation control approaches have been presented in the literature [6, 19, 20, 49, 65, 84, 91, 92, 94, 100, 104]. These papers cover a variety of approaches, including PD, LQR, LMI, nonlinear, Lyapunov, impulsive, RRT, and model predictive. Typically, it is assumed that a formation is initialized to a stable orbit and deviations caused by disturbances such as differential drag and/or differential $J_2$ must be corrected. Some approaches, such as Lyapunov controllers and PD controllers [92], require that control be applied continuously, a strategy both prone to high fuel use and difficult to implement when thrusting requires attitude adjustment. Other approaches, such as the impulsive thrusting scheme introduced in Ref. [93], require spacecraft to thrust at previously specified times and directions in the orbit, ensuring...
many potential maneuvers will not be fuel-optimal.

Model Predictive Control (MPC) can be used to generate optimized plans that satisfy performance constraints \([24–26, 49, 73, 84, 101]\). MPC using linear programming (LP) has a number of other advantages for spacecraft formation flying: it easily incorporates realistic constraints on thrusting and control performance; it generates plans that closely approximate fuel-optimal “bang-off-bang” solutions rather than the continuous thrusting plans that inevitably arise from LQR, \(H_\infty\), and Lyapunov controllers; and it allows for piecewise-linear cost functions, such as the 1-norm of fuel use.

### 1.1.2 Linearized Relative Orbital Dynamics

Optimization-based controllers make explicit use of the system dynamics. Because of the advantages of linear optimization (i.e., fast solution times, global optimality), it is preferable to use linear relative dynamics in the model predictive controller. Linear models have the advantage that they can easily exploit the superposition principle to predict the effects of future inputs.

A variety of sets of linear dynamics for relative orbit propagation have been examined in the literature and are summarized in Table 1.1. When spacecraft are in very close proximity (meters) their relative motion is often modeled as a double integrator. More widely separated formations in circular orbits (usually less than 1 km in LEO \([115]\)) can use the Hill-Clohessy-Wiltshire (HCW) equations \([75, 76]\). For very large formations in circular orbits, there are a number of modifications in the literature that can be used to improve the accuracy of Hill’s equations \([45, 117, 118]\). For propagation of relative elliptical orbits, Lawden’s equations are valid for any eccentricity \([16–18]\). However, like Hill’s equations, they degrade quickly with separation distance. Note that both Hill’s and Lawden’s equations have been modified in the literature \([88, 116]\) to include the relative effects of Earth’s oblateness, however, these approaches are still only valid for formations with short baselines.

An alternative to planning in Cartesian frames is using orbital elements, which have been shown to not degrade as rapidly with separation distance \([74]\). In orbital
elements, the relative dynamics can be propagated using Gauss’ Variation Equations (GVEs). A state transition matrix capable of propagating spacecraft with large separations in elliptical orbits and incorporate the effects of $J_2$ is presented in [119]. This thesis builds on the work in [119] to create a discrete input effect matrix that incorporates the effects of the same range of disturbances and use the combined linear time-varying dynamics in a model predictive controller.

1.1.3 Formation Flying Initial Conditions

One of the principal requirements of a spacecraft formation is that the component spacecraft do not drift apart from one another [14, 65]. In a fully Keplerian orbit, the only source of drift over multiple orbits is a difference between spacecraft periods, which is equivalent to a difference in spacecraft semimajor axes [15]. The presence of the relative disturbances between spacecraft (e.g., relative drag, $J_2$) can also lead to drift in a formation. An alternative to expending regular control energy to counteract drift is to choose formation initial conditions that reduce relative drift between spacecraft.

Several approaches for creating $J_2$ invariant relative orbits have recently been proposed in the literature [20, 66, 110]. Different classes of “invariant” orbits have been introduced: those that are truly invariant over time, orbits that retain the same mean period over time, orbits that are invariant except for argument of perigee drift, and orbits that are invariant except for right ascension drift. In the case of full invariance conditions, where the formation returns to an identical relative state

| Table 1.1: Relative orbital dynamics indexed by regime of validity [115] |
|---------------------------------|-----------------|----------------|-----------------|-----------------|
| $e = 0$                         | $0 < e < 1$     | $e = 0$         | $0 < e < 1$     |
| no $J_2$                        | no $J_2$        | with $J_2$      | with $J_2$      |
| Linearized Dynamics             | Hill’s [75]     | Lawden [16]     | Schweighart [116]| Chrétien [88] |
|                                | Mitchell [118]  | Alfriend [119]  |                 |                 |
|                                | Alfriend [45]   |                 |                 |                 |
every orbit, the set of relative orbits that satisfy the conditions is very small and the
gometry of those orbits is highly restricted [20]. Hence, it is more common for a
$J_2$-invariant orbit to only be invariant in a reduced set of dimensions for which it is
possible to analytically cancel the relative effects of $J_2$. In all of the aforementioned
invariance cases, the drift being minimized is secular variation in the mean orbital
elements.

1.1.4 Safety in Autonomous Rendezvous and Docking

Numerous methods for generating rendezvous trajectories exist in the literature and
encompass a wide range of rendezvous scenarios [61–63, 67, 79]. Those papers consider
rendezvous from many perspectives, often taking into account complicated collision
avoidance constraints, nonlinear rotational dynamics, and fuel efficiency. Another
perspective to be considered when designing trajectories is safe behavior [67–70].
Safety in the context of spacecraft rendezvous and docking is typically with respect to
collision avoidance following some type of failure. The approach in Ref. [70] creates
trajectories which naturally tend to drift away from the target spacecraft in the
absence of thrusting. This method can guarantee safety for thruster failures, but
is not fuel-optimized and does not apply to more complicated docking situations in
which those trajectories cannot be used for nominal rendezvous.

Alternately, Refs. [67] and [68] develop the safety circle method, in which a nearby
orbit with a relative invariant trajectory is established that allows safe long-term ob-
servation before docking, however this approach is not fuel optimized and does not
propose a specific docking path. A method proposed in Ref. [69] optimizes both
safety and fuel using genetic algorithms. This approach treats safety as a goal rather
than a constraint and thus, cannot assure that the resulting trajectory would be safe.
Ref. [79] plans safe trajectories using potential functions, but the approach is comput-
ationally intensive and limited to static obstacles. Various types of safety have been
considered in the design of UAV trajectories, but these focused on creating trajecto-
ries that are safe under nominal operating conditions (e.g., safety from adversaries,
uncertain terrain)[71, 72].
1.2 Thesis Overview

This thesis develops and validates technologies that improve the state of the art in control for formation flying spacecraft and for the autonomous rendezvous of spacecraft. The chapters and their contributions are:

Chapter 2 develops, validates, and analyzes a set of time-varying linearized relative spacecraft dynamics that advances the state of the art by incorporating three dominant orbital effects not previously accommodated together: 1) Orbit eccentricity; 2) nonlinearity due to large separation distances between spacecraft; and 3) Earth oblateness. All three are expected to be present in the planned MMS mission [86]. Continuous- and discrete-time versions of the new dynamics are presented and linearization assumptions for each are evaluated. The dynamics are embedded in an LP-based model-predictive control system and demonstrated controlling a four spacecraft formation in a highly elliptic orbit in the presence of realistic disturbances and navigation uncertainty.

Chapter 3 discusses the use of the dynamics presented in Chapter 2 in a linear optimization-based initialization algorithm that produces fuel-minimizing invariant orbits. This approach improves existing techniques for producing invariant orbits by explicitly considering multiple objectives in its cost function. The costs minimized are: 1) Cartesian, mean, and/or osculating drift over arbitrary time frames; 2) fuel costs associated with maneuvering to the desired initial conditions; and 3) the distance of the initial conditions from a desired formation geometry (e.g., a shape appropriate for observation or science data collection). Existing techniques rely either on analytic conditions to prevent mean drift over an infinite horizon with no notion of fuel or geometry cost; or on large nonlinear optimization techniques that are ill-suited to onboard deployment. This chapter investigates the ranges of solutions available for large ranges of objective weights and compares those solutions to semi-invariant conditions available through analytic techniques.

Chapter 4 uses the controller/dynamics combination developed in Chapter 2 to examine a realistic formation flying mission scenario and evaluate the effectiveness of
control parameter settings. The mission examined has three satellites and maneuvers between multiple formation configurations (1 km in-track, 50 m passive aperture, 500 m passive aperture, 5 km passive aperture) in which station-keeping is performed using 10% baselines for hard error box constraints. Each simulation covers a multi-week period to demonstrate formation stability and to determine steady-state fuel use. The effects of constraints on passive operation during science data collection on fuel use and performance constraint satisfaction in the presence of navigation error are investigated. Variations of optimization terminal conditions and error box relaxations are also examined.

Chapter 5 introduces a new approach to guaranteeing safety against failures in autonomous rendezvous and docking maneuver generation for spacecraft. The maneuver generation algorithm uses the same LP-based optimization as the model predictive controller in Chapter 2 to minimize fuel, but uses additional linear constraints to ensure safety. No other guaranteed-safe rendezvous methods in the literature also minimize fuel use. The approach in this chapter has the added advantage of being valid for safe docking with general polygonal target shapes experiencing arbitrary, but known, attitude motion under any relative linear dynamics. The fuel cost associated with imposing safety as a constraint is investigated and the value of adding safety is established through stochastic analysis. In addition, a modification to the safe trajectory formulation is examined that, through use of the invariance concept in Chapter 3, enables infinite horizon passive safe collision-avoidance guarantees. A convex formulation of the safety constraint is developed and analyzed in terms of fuel cost and computation trades. Also, an active form of safety is developed and evaluated that greatly expands the space of safe rendezvous maneuvers by allowing powered abort trajectories.

Chapter 6 applies the safe rendezvous generation techniques developed in Chapter 5 to use on a hardware testbed aboard the International Space Station. Both nominal and stochastic passive-abort cases are examined. Additional testing is conducted using a similar terrestrial testbed.

Chapter 7 summarizes the main contributions of the thesis to the state of the art
in formation flying control and autonomous rendezvous and docking of spacecraft. These contributions are to linearized relative dynamics, initialization techniques, and safety in autonomous rendezvous and docking.
Chapter 2

GVE-based Dynamics and Control for MPC

This chapter presents several modeling and control extensions that would enhance the efficiency of many formation flying missions. In particular, a new linear time-varying form of the equations of relative motion is developed from Gauss' Variational Equations. These new equations of motion are further extended to account for the effects of $J_2$, and the linearizing assumptions are shown to be consistent with typical formation flying scenarios. It is then shown how these models can be used to control general formation configurations in an embedded on-line, optimization-based, model predictive controller (MPC). A convex, linear approach for initializing fuel-optimized, partially $J_2$ invariant orbits is developed and compared to analytic approaches. All control methods are validated using a commercial numerical propagator. The simulation results illustrate that formation flying using this MPC with $J_2$-modified GVEs requires fuel use that is comparable to using unmodified GVEs in simulations that do not include the $J_2$ effects.
Nomenclature

\[ \begin{align*}
    a &= \text{semimajor axis} \\
    e &= \text{eccentricity} \\
    i &= \text{inclination} \\
    \Omega &= \text{right ascension of the ascending node} \\
    \omega &= \text{argument of perigee} \\
    M &= \text{mean motion} \\
    p &= \text{semilatus rectum} \\
    b &= \text{semiminor axis} \\
    h &= \text{angular momentum} \\
    \theta &= \text{argument of latitude} \\
    r &= \text{magnitude of radius vector} \\
    n &= \text{mean motion}
\end{align*} \]

2.1 Background

Formation control objectives typically focus on controlling the relative states of the spacecraft, the dynamics of which can be captured using variants of Hill’s and Lawden’s equations for LEO missions [84]. However, both of these approaches linearize the nonlinear relative spacecraft motions about a reference orbit, which is only valid for small separation distances of the satellites in the formation relative to the reference orbit radius. For larger separations, these equations of motion can no longer be used to cancel relative drift rates (initialization) or to accurately predict the effect of inputs (control) [85]. For example, the four spacecraft of the planned MMS mission [86] will be placed in a tetrahedron-shaped relative configuration with sides ranging between 10–1000 km at apogee, which far exceeds the separations for which Hill’s and Lawden’s models are valid for a full HEO orbit, even with the correction terms introduced in Ref. [87]. Furthermore, these models do not accurately capture the effects of Earth’s oblateness, which Ref. [20] showed can lead to very inefficient control designs. This chapter develops a new linearized modeling approach that is valid for widely-spaced formations in highly elliptic orbits, accurately captures the effects of the Earth’s gravity, and can be embedded in an optimization-based controller.
that is suitable for real-time calculations.

The relative dynamics used in this chapter are based on a form of Gauss’ Variational Equations (GVEs) that have been modified to include the effects of $J_2$. GVEs are convenient for specifying and controlling widely separated formations because they are linearized about orbital elements, which are expressed in a curvilinear frame in which large rectilinear distances can be captured by small element perturbations [89]. This bypasses the linearization error created by representing the entire formation in a single rectilinear frame, which was the approach used in Ref. [84]. The use of GVE dynamics as opposed to Hill’s dynamics incurs the cost of computation associated with the use of multiple sets of time-varying equations of motion. Specifying a formation’s relative geometry in terms of differential orbital elements is an exact approach that does not degrade for large spacecraft separations. However, the advantage of using GVEs for control could be reproduced by using Lawden’s equations of motion in a different LVLH frame for each spacecraft in the formation while still using orbital element differences to represent the formation relative geometry. Given that a nonlinear transformation and rotation is required to switch between an LVLH frame and orbital element differences, and that GVEs are already linearized in an orbital element frame, it is both simpler and computationally more efficient to use orbital element differences to specify the formation configuration and GVEs for control.

Many formation control approaches have used GVEs for nonlinear, continuous control [90–92] and also for impulsive control [93, 94]. This chapter introduces a control law that generally does not fire continuously and, more importantly, makes explicit its objective to minimize fuel use, which is measured in $\Delta V$ in this chapter. The control approach optimizes the effects of arbitrarily many inputs over a chosen planning horizon. Plans are regularly re-optimized, forming a closed-loop system [95]. By extending previous planning approaches [84, 96] to use GVEs, we can optimize the plans for spacecraft in widely-separated, highly elliptic orbits. Results are presented to show that the GVE-based planning system is more fuel-efficient than the four-impulse method in Ref. [93]. In addition, control optimized online has the advantage of being capable of handling many types of constraints, such as limited thrust capability,
sensor noise robustness, and error box maintenance [84]. We also extend the virtual center approach to formation flying in Ref. [97] to GVEs and present a decentralized implementation of that algorithm.

A limitation of the orbital element approach in Ref. [96] is that it does not account for the effects of the $J_2$ disturbance, which impacted the closed-loop performance in full nonlinear simulations. This chapter extends the use of the relative orbital elements in Ref. [96] to the $J_2$-modified relative state transition matrix in Ref. [98] and develops and evaluates several approaches for including the effects of thruster inputs. The resulting $J_2$-modified GVEs are used to form a set of linear parameter-varying dynamics that can be embedded in an optimization-based control system. The combination creates a controller that retains the advantages of the GVE-based controller in Ref. [96], but uses a more accurate dynamics model, thereby improving plan tracking and fuel efficiency. In particular, simulations are presented to show that the new controller in the presence of $J_2$ disturbances requires comparable levels of fuel to the approach in Ref. [96] when no $J_2$ disturbances are simulated in the model.

### 2.2 Relative Orbital Elements and Linearization

Validity

Gauss’ Variational Equations (GVEs) are derived in Ref. [99] and are reproduced here for reference

\[
\frac{d}{dt} \begin{pmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ M \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ n \end{pmatrix} + \begin{pmatrix} \frac{2a^2 e \sin f}{h} & \frac{2a^2 p}{v_h} & 0 \\ \frac{p \sin f}{h} & \frac{(p+r) \cos f + re}{h} & 0 \\ 0 & 0 & \frac{r \cos \theta}{h} \\ \frac{p \cos f}{he} & \frac{(p+r) \sin f}{he} & -\frac{r \sin \theta \cos i}{h \sin i} \\ \frac{b(p \cos f - 2re)}{ahe} & -\frac{b(p+r) \sin f}{ahe} & 0 \end{pmatrix} \begin{pmatrix} u_r \\ u_\theta \\ u_h \end{pmatrix} \tag{2.1}
\]
where the state vector elements are $a$ (semimajor axis), $e$ (eccentricity), $i$ (inclination), $\Omega$ (right ascension of the ascending node), $\omega$ (argument of periapse), and $M$ (mean motion). The other terms in the variational expression are $p$ (semi-latus rectum), $b$ (semiminor axis), $h$ (angular momentum), $\theta$ (argument of latitude), $r$ (magnitude of radius vector), and $n$ (mean motion). All units are in radians, except for semi-major axis and radius (meters), angular momentum (kilogram · meters$^2$ per second), mean motion (1/seconds), and eccentricity (dimensionless). The input acceleration components $u_r$, $u_\theta$, and $u_h$ are in the radial, in-track, and cross-track directions, respectively, of an LVLH frame centered on the satellite and have units of meters per second$^2$. Although the traditional Keplerian form of the orbital elements is used in this chapter for conceptual clarity, later uses of transformations from Refs. [98] and [100] require a conversion to the nonsingular form described in those references. The form of the GVEs can be more compactly expressed as

$$
\dot{\mathbf{e}} = A(\mathbf{e}) + B(\mathbf{e})\mathbf{u}
$$

(2.2)

where $\mathbf{e}$ is the state vector in Eq. (2.1), $B(\mathbf{e})$ is the input effect matrix, $\mathbf{u}$ is the vector of thrust inputs in the radial, in-track, and cross-track directions, and $A(\mathbf{e}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{\mu/a^3} \end{pmatrix}^T$, where $\mu$ is the gravitational parameter.

In a formation, the orbital element state of the $i$th satellite is denoted $\mathbf{e}_i$. The states of the vehicles in the formation can be specified by relative orbital elements by subtracting the state of an arbitrarily chosen spacecraft in the formation, which is designated as $\mathbf{e}_1$

$$
\delta\mathbf{e}_i = \mathbf{e}_i - \mathbf{e}_1
$$

(2.3)

For a desired orbit geometry, a set of desired relative elements, $\delta\mathbf{e}_{di}$ will specify the desired state $\mathbf{e}_{di}$ of each spacecraft in the formation$^1$.

$$
\mathbf{e}_{di} = \mathbf{e}_1 + \delta\mathbf{e}_{di}
$$

(2.4)

$^1$Approaches for choosing and coordinating the desired spacecraft states will be addressed in Sections 2.4 and Chapter 3.
The state error for the \( i \)th spacecraft in the formation, \( \zeta_i \), is then defined as

\[
\zeta_i = e_i - e_{di} = \delta e_i - \delta e_{di}
\]

(2.5)

Note that the definition of state error given in Eq. (2.5) is independent of the choice of which spacecraft state is represented by \( e_1 \). The form of Gauss’ Variational Equations in Eq. (2.1) is for perturbations of orbital elements. To reformulate these equations for perturbations of relative orbital elements \([100]\), the GVEs for \( e_i \) and \( e_{di} \) are combined

\[
\dot{\zeta}_i = \dot{e}_i - \dot{e}_{di} = A(e_i) - A(e_{di}) + B(e_i)u_i
\]

(2.6)

where the term \( B(e_{di})u_{di} \) has been excluded, since thrusting does affect the desired state of the spacecraft. The unforced dynamics can be linearized by introducing the first-order approximation \([100]^2\)

\[
A(e) - A(e_d) \approx \frac{\partial A}{\partial e} \bigg|_{e_d} (e - e_d) = \frac{\partial A}{\partial e} \bigg|_{e_d} \zeta \equiv A^*(e_d)\zeta
\]

(2.7)

where the matrix \( A^*(e_d) \) is all zeros except for the lower-leftmost element, which is \(-3n/2a\), where the sparsity of \( A^* \) arises from the sparsity of the \( A \) function in Eq. (2.2). With this approximation, the differential GVE expression in Eq. (2.6) can be rewritten as

\[
\dot{\zeta} = A^*(e_d)\zeta + B(e)u = A^*(e_d)\zeta + B(e_d + \zeta)u
\]

(2.8)

In this case the control of the relative error state, \( \zeta \), is nonlinear, because the control effect matrix \( B \) is a function of the state. Ref. [100] accounts for this nonlinearity in a continuous nonlinear control law that was shown to be asymptotically stable. The control approach developed in this section uses linearized dynamics to predict the

\(^2\) The subscript \( i \) is henceforth omitted for notational simplicity.
effect of future control inputs. Linearizing the matrix \( B \) in Eq. (2.8) yields

\[
\dot{\zeta} \simeq A^*(e_d)\zeta + \left( B(e_d) + \frac{\partial B}{\partial e} \right) u = A^*(e_d)\zeta + B(e_d)u + [B^*(e_d)]\zeta u
\] (2.9)

where the term \( B^*(e_d) \) is a third rank tensor and the quantity \( B^*(e_d)\zeta \) is a matrix with the same dimensions as \( B(e_d) \). For convenience, define

\[
\Delta B(e_d, \zeta) \equiv B^*(e_d)\zeta
\] (2.10)

resulting in the new state equation

\[
\dot{\zeta} = A^*(e_d)\zeta + (B(e_d) + \Delta B(e_d, \zeta)) u
\] (2.11)

Note that if \( \Delta B \) is much smaller than \( B(e_d) \), then the first-order term can safely be ignored, yielding the approximate linearized dynamics

\[
\dot{\zeta} = A^*(e_d)\zeta + B(e_d)u
\] (2.12)

which can be controlled by any one of a variety of linear control techniques, including the model predictive controller discussed in Section 2.4.

The critical requirement for linear control and planning is that the term \( \Delta B \) has a much smaller influence on the state dynamics than the term \( B(e_d) \). However, \( \Delta B \) is a linear function of the state error \( \zeta \), which can be arbitrarily large. The amount of acceptable error due to linearization will be a function of the mission scenario, but the linearization assumption will typically only be valid for small values of the state error. It is reasonable to expect that the values of state error will be small, because the linearization is only in separation between a spacecraft and its desired orbit. For a given desired orbit, a bound can be established numerically that indicates the state separation from an orbit where the dynamics linearization is valid. This section examines several example orbits that are representative of space missions that might occur in Low Earth Orbits and Highly Elliptical orbits. In each case, this range
of acceptable error is found to be large enough to accommodate expected mission performance requirements.

The magnitude of the acceptable error can be computed by comparing the induced norm of the difference between the control influence matrix at its desired state, \( B(e_d) \) and at the actual position of the spacecraft, \( B(e) \). In Eq. (2.10), the first order approximation of this term was defined as \( \Delta B \). In the following examples, \( \Delta B_{\text{true}} \), which is defined as

\[
\Delta B(e_d, \zeta)_{\text{true}} \equiv B(e) - B(e_d) = B(e_d + \zeta) - B(e_d)
\] (2.13)

and will be calculated numerically. The cut-off point of acceptable linearization error is when the norm of \( \Delta B \) exceeds some (possibly mission dependent) fraction of the norm of \( B(e_d) \). To investigate this cut-off point, the following examples consider many random values of \( \zeta \) in the set \( \|\zeta\|_2 = r \) and calculate \( \Delta B_{\text{true}} \). The \( \Delta B_{\text{true}} \) with the largest 2-norm will be used to test the validity of the linearization for a given \( r \). This procedure is repeated for multiple \( r \) to find the largest \( \|\zeta\|_2 \) for which the linearization is considered valid. Other methods of examining the linearization error of \( B \) are possible, but the approach used in this chapter was chosen because of its ease of implementation and consistent results for particular mission types.

**Example: Low Earth Orbit** – An example low Earth orbit is

\[
e_d = \begin{pmatrix} 1.08182072 & 0.005000000 & 0.610865238 & 2\pi & \pi & 3.82376588 \end{pmatrix}^T
\] (2.14)

where the first element, semimajor axis, is normalized by the Earth’s radius, making
the orbital element vector dimensionless. The matrix corresponding $B(e_d)$ is

$$B(e_d) = \begin{pmatrix}
-5.6794478 & 1808.6011 & 0 \\
-0.000082308780 & -0.00020502572 & 0 \\
0 & 0 & 0.00010304404 \\
0 & 0 & 0.00014406293 \\
0.020528419 & -0.032987976 & -0.00011800944 \\
-0.020792326 & 0.032987564 & 0
\end{pmatrix}$$

(2.15)

where $\|B(e_d)\|_2 = 1808.61$. The effect of perturbing $e_d$ for a given norm bound on $\zeta$ is shown in Figure 2-1. The plots show that an arbitrary linearization validity cutoff of 0.01, i.e., $\|\Delta B(e_d, \zeta)_{true}\|_2 \leq 0.01\|B(e_d)_{true}\|_2$, can be achieved by ensuring that $\|\zeta\|_2 \leq 8.16 \times 10^{-3}$. This bound on $\zeta$ allows for orbital element perturbations that equate to rectilinear distances on the order of 25 kilometers and velocities on the order of 40 m/s. Typical error box sizes for LEO formation flying missions are 10-100 meters in size [4], decidedly inside the linearization range of the LEO orbit examined.

**Example: Highly Elliptical Earth Orbit** – One motivation for using GVEs as the linearized dynamics in a planner is recent interest in widely spaced, highly elliptical orbits [86]. An orbit of this type is

$$e_d = \begin{pmatrix}
6.59989032 & 0.818181000 & 0.174532925 & 2\pi & 0 & \pi
\end{pmatrix}^T$$

(2.16)

with

$$B(e_d) = \begin{pmatrix}
4.767920 \times 10^{-12} & 8651.830 & 0 \\
2.288208 \times 10^{-20} & -0.0003736926 & 0 \\
0 & 0 & -0.001027650 \\
0 & 0 & 7.247461 \times 10^{-19} \\
0.0002283680 & 1.817849 \times 10^{-19} & -7.137356 \times 10^{-19} \\
-0.001313020 & -1.45192 \times 10^{-19} & 0
\end{pmatrix}$$

(2.17)
Fig. 2-1: Effect of Orbital Element Perturbations on the $\Delta B_{\text{true}}$ Matrix for a LEO Orbit

Fig. 2-2: Effect of Orbital Element Perturbations on the $\Delta B_{\text{true}}$ Matrix for a HEO Orbit
Repeating the same procedure used for the LEO case, it is determined from Figure 2-2 that an arbitrary 1% linearization validity cutoff can be achieved provided that \( \| \zeta \|_2 \leq 3.66 \times 10^{-3} \). In this case, the bound on \( \| \zeta \|_2 \) corresponds to rectilinear distances of approximately 50 kilometers and velocities of 2 meters per second. As in the LEO case, these distances are far larger than expected error box sizes. Since any planned trajectory would be expected to remain inside an error box at all times, the range of state errors in which the linearization is valid will not be exceeded. Unlike the LEO case, error boxes for widely-separated missions, such as MMS, may be much larger than 10 meters to a side, even approaching kilometers. The 1% cutoff ensures that error boxes of up to 5% of the distance between MMS satellites (1000 km during the most widely spaced phase of the mission) are acceptable [86].

Validating the linearization for additional reference orbits is a straightforward computational exercise. For example, repeating the validation process for the LEO orbits used in Chapter 3 and the HEO orbits used in the simulations in Section 2.6 yielded valid ranges of separation that were far larger than the expected error box sizes.

### 2.3 \( J_2 \)-Modified GVEs and Linearization Validity

Just as the GVEs in Eq. (2.2) express the motion of a Keplerian orbit, the equations of motion of the mean orbital element state vector \( e_m \) describes the average motion of an orbit influenced by Earth oblateness effects and are given by

\[
\dot{e}_m = \bar{A}(e_m) + \frac{\partial e_m}{\partial u} u \tag{2.18}
\]

where \( \bar{A} \) is explicitly a function of the mean state and implicitly a function of \( J_2 \), see Ref. [100]. Although Eqs. (2.2) and (2.18) appear similar, there are some important differences. In particular, Eq. (2.2) describes the motion of a spacecraft’s osculating orbit and is the form of the classical GVEs. Section 2.2 established that it is valid and effective to linearize the GVEs and use them for model predictive control. However,
the GVEs incorporate neither the absolute nor the relative effects of $J_2$ on a satellite’s orbit. Conversely, Eq. (2.18) describes the motion of an orbit in a set of mean orbital elements, where the secular effects of $J_2$ are incorporated and harmonics are removed. This form of the dynamics is useful for controlling the secular drift between satellites in a formation, but does not describe the physical motion and has limited applicability for missions with high precision relative state constraints. Furthermore, Eq. (2.18) is nonlinear in terms of the relative state, which accurately captures the system dynamics, but complicates the optimization of the control inputs. The following shows that, by utilizing the linearized propagation and rotation matrices developed in Ref. [98], a linearized form of the equations of relative motion in Eq. (2.18) can be derived that incorporates the osculating effects of $J_2$, is linear parameter varying, and is valid for large spacecraft separations and reference orbit eccentricities.

The control influence matrix for mean element motion is derived using the transformation matrices between the mean and osculating motion. The following identity is used to define these transformations,

$$\frac{\partial \mathbf{e}_m}{\partial \mathbf{u}} = \left( \frac{\partial \mathbf{e}_m}{\partial \mathbf{e}} \right) \left( \frac{\partial \mathbf{e}}{\partial \mathbf{u}} \right)$$

(2.19)

From the appendix of Ref. [20], the relation between the mean orbital element state vector and the osculating orbital element state vector can be written as $\mathbf{e}_m = f(\mathbf{e})$, so that

$$\dot{\mathbf{e}}_m = \frac{\partial f(\mathbf{e})}{\partial \mathbf{e}} \dot{\mathbf{e}} \quad \Rightarrow \quad \frac{\partial \mathbf{e}_m}{\partial \mathbf{e}} = \frac{\partial f(\mathbf{e})}{\partial \mathbf{e}}$$

(2.20)

Substituting Eq. (2.20) and the $B$ matrix from Eq. (2.2) into Eq. (2.19) gives

$$\frac{\partial \mathbf{e}_m}{\partial \mathbf{u}} = \frac{\partial f(\mathbf{e})}{\partial \mathbf{e}} B(\mathbf{e})$$

(2.21)

which yields the equations of motion of the mean orbit in terms of the osculating orbital state vector $\mathbf{e}$ (the mean elements may be considered a function of the osculating
elements) and an input vector \( \mathbf{u} \) as

\[
\dot{e}_m = \bar{A}(e_m) + \frac{\partial f(e)}{\partial e} B(e) \mathbf{u} \quad (2.22)
\]

The actual mean orbit \( e_m \) is now defined in terms of a desired mean orbit \( e_{md} \) and a vector offset \( \zeta_m \)

\[
e_m = e_{md} + \zeta_m \quad (2.23)
\]

Rearranging this expression and applying Eq. (2.18) gives

\[
\dot{e}_m - \dot{e}_{md} = \dot{\zeta}_m = \bar{A}(e_m) - \bar{A}(e_{md}) + \frac{\partial \dot{e}_m}{\partial \mathbf{u}} \mathbf{u} \quad (2.24)
\]

where the term \( \frac{\partial \dot{e}_{md}}{\partial \mathbf{u}} \mathbf{u} \) is omitted because the desired orbit is fixed and not subject to thrusting. Similar to the previous section, the following linearization approximation can be made [100]

\[
\bar{A}(e_m) - \bar{A}(e_{md}) \approx \frac{\partial \bar{A}}{\partial e_m} \bigg|_{e_{md}} \zeta_m \equiv \bar{A}^*(e_{md}) \zeta_m \quad (2.25)
\]

which is then used to find the equations of motion of the mean element offset \( \zeta_m \)

\[
\dot{\zeta}_m = \bar{A}^*(e_{md}) \zeta_m + \frac{\partial \dot{e}_m}{\partial \mathbf{u}} \mathbf{u} \quad (2.26)
\]

where the terms of the matrix function \( \bar{A}^* \) are given in Ref. [100]. Equation (2.26) provides a linear description of the motion of the relative mean orbital elements. However, the mean orbit describes where the spacecraft is in an average sense, whereas the osculating orbits specifies the actual position of spacecraft. Thus, to maximize the ability of the planner to exploit natural dynamics and operate with tight performance constraints, it is preferable to plan in terms of the osculating orbit. The approach in this chapter uses a hybrid of the osculating and mean to capture both the effects of \( J_2 \) and plan in a way that accounts for the actual motion of the spacecraft. Having developed the relative dynamics in terms of the mean elements, we now convert to an osculating state.
Using the notation in Eq. (2.5), formation relative dynamics can be specified in terms of the osculating orbit \( e \), an osculating desired orbit \( e_d \), and an osculating orbital offset \( \zeta \) between them. The mean elements are expressed as functions of the osculating elements by rearranging the state error form in Eq. (2.23). This is used to create a relative state and a linearized rotation matrix for transitioning between the mean and osculating equations of relative motion.

Given that \( e_m = f(e) \) and \( e_{md} = f(e_d) \), then using Eq. (2.5), Eq. (2.23) can be rewritten as

\[
\zeta_m = f(e) - f(e_d) \approx \frac{\partial f(e)}{\partial e} \bigg|_{e_d} \zeta
\]  

(2.27)

by utilizing the same linearization approach in Ref. [100]. Defining the matrix function \( D \) (available in Ref. [98]),

\[
D(e_d) \equiv \frac{\partial f(e)}{\partial e} \bigg|_{e_d}
\]

(2.28)

and substituting into Eq. (2.21) and then into Eq. (2.26) yields

\[
\dot{\zeta}_m = \bar{A}^*(f(e_d))\zeta_m + D(e)B(e)u
\]

(2.29)

This form of the relative equations of motion is nonlinear in terms of the osculating absolute state \( e \). Making the linearizing assumption (accuracy of the linear approximations is discussed later in this section)

\[
D(e)B(e) = D(e_d + \zeta)B(e_d + \zeta) \approx D(e_d)B(e_d)
\]

(2.30)

allows the relative equations of motion to be rewritten as

\[
\dot{\zeta}_m = \bar{A}^*(f(e_d))\zeta_m + D(e_d)B(e_d)u
\]

(2.31)

which has a desired osculating orbit \( e_d \) and is linear in terms of the relative mean state \( \zeta_m \). The equations of motion in Eq. (2.31) are still not suited to control of the osculating relative orbit in the presence of \( J_2 \), because they describe the derivative of the mean state. The following section derives a form of discrete dynamics that use a
relative osculating orbit as their state.

### 2.3.1 Extension to Discrete Time

To use Eq. (2.31) in an optimization-based controller of the type used in Ref. [96], it must first be discretized. Reference [98] introduces the state transition matrix $\bar{\Phi}$, which is the discrete form of the continuous matrix $\bar{A}^*(f(e_d))$, and is defined such that

$$\zeta_m(t_1) = \bar{\Phi}^*(e_md(t_0), t_1, t_0)\zeta_m(t_0)$$  \hspace{1cm} (2.32)

where $t_0$ and $t_1$ are the times of the initial and final states, respectively, and are provided as arguments to the state vectors. The analytic definition of the matrix $\bar{\Phi}^*$ (an implicit function of $J_2$ and a highly nonlinear function of the mean absolute elements) is included in the appendices of Ref. [98].

The dynamics in Eq. (2.31) can be formulated exclusively in terms of the osculating state. Using Eq. (2.27), define $D^{-1}$ as

$$D^{-1}(e_d) \equiv \left. \frac{\partial e}{\partial f(e)} \right|_{e_d}$$  \hspace{1cm} (2.33)

Substituting Eqs. (2.28) and (2.33) into Eq. (2.32) yields

$$\zeta(t_1) \approx D^{-1}(e_d(t_1))\bar{\Phi}^*(e_md(t_0), t_1, t_0)D(e_d(t_0))\zeta(t_0)$$  \hspace{1cm} (2.34)

The analogous discrete form of the control influence matrix $B$ on the osculating state is then given by

$$\Gamma(e_d(t_0), t_1, t_0) = \int_{t_0}^{t_1} D^{-1}(e_d(t_1))\bar{\Phi}^*(e_md(\tau), t_1, \tau)D(e_d(\tau))B(e_d(\tau))d\tau$$  \hspace{1cm} (2.35)

Thus, combining Eqs. (2.32) and (2.35) yields the discrete time equations of motion

$$\zeta(t_1) \approx D^{-1}(e_d(t_1))\bar{\Phi}^*(e_md(t_0), t_1, t_0)D(e_d(t_0))\zeta(t_0) + \Gamma(e_d(t_0), t_1, t_0)u$$  \hspace{1cm} (2.36)
which are the linear parameter-varying discrete equations of motion for a relative osculating orbit in the presence of $J_2$.

### 2.3.2 Validity of the Linearization Approximations

Reference [87] showed that the approximation $B(e_d) \approx B(e_d + \zeta)$, which is used to derive Eq. (2.36), is a sufficiently close approximation for levels of state error, $\zeta$, that would normally be expected in spacecraft formation flying missions. In order to use Eq. (2.36) for linear control, it must also be shown that the linearized rotation and transition combination $D^{-1}(e_d(t_1))\Phi^*(e_{md}(t_0), t_1, t_0)D(e_d(t_0))$ remains a close approximation for expected values of $\zeta$. Ref. [98] showed this matrix has low linearization error for a wide range of reference orbit eccentricities and spacecraft formation baselines in excess of 10 km. By specifying the chief orbit in Ref. [98] as the desired spacecraft state, the transition matrices then allow a maximum state error of 10 km, which is much larger than the error that would be tolerated in most proposed spacecraft formation flying missions.

### 2.3.3 Calculating the $\Gamma$ matrix

The discrete control effect matrix is defined as a matrix integral in Eq. (2.35). One way to calculate this matrix is by computing its derivative and numerically integrating. However, in practice this is a computationally intensive approach that may not be consistent with real-time controller implementation. A number of alternate approaches exist. This subsection discusses several of those techniques and compares their accuracy and computation times.

**Continuous Integration** The continuous integration method for getting the discrete input matrix $\Gamma$ from time $t_0$ to time $t_1$ is

$$
\Gamma_{true} = \int_{t_0}^{t_1} \left\{ D^{-1}(e_d(t_1))\Phi^*(e_{md}(\tau), t_1, \tau)D(e_d(\tau))M(e_d(\tau)) \begin{bmatrix} 0_3 \\ I_3 \end{bmatrix} \right\} d\tau \quad (2.37)
$$
where

\[ M \mathbf{x} = \delta e_{\text{osc}} \quad (2.38) \]

and the analytic form of \( M \) can be found in Ref. [100]. The vector \( \mathbf{x} \) is in the LVLH coordinate system and has the form

\[ \mathbf{x} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T \quad (2.39) \]

where the positions \( x, y, \) and \( z \) are in meters and the velocities \( \dot{x}, \dot{y}, \) and \( \dot{z} \) are in meters per second. This approach should use no additional linearization assumptions beyond those in Ref. [98], since the inputs and their coupling effects are incorporated continuously. While the outputs of the integration in Eq. (2.37) are very accurate, the approach itself requires significant computational effort (see Section 2.3.3).

**Discretized Integration**  An approach to integrating \( \Gamma \) that is not as computationally intensive as the numerical integration is to approximate the integral in Eq. (2.37) discretely. This discrete approach introduces an additional timestep \( \Delta t \) which is the duration of each discrete term in the new approximate \( \Gamma \). Conceptually, this approach is treating \( \Gamma \) as a series of smaller input effects, each based on a small time-invariant assumption. The discretized integration is

\[
\Gamma_{\text{disc}} = \sum_{i=1}^{n} D^{-1}(\mathbf{e}_d(t_1)) \Phi^*(\mathbf{e}_{md}(t_0+i\Delta t), t_1, t_0+i\Delta t) D(\mathbf{e}_d(t_0+i\Delta t)) M(\mathbf{e}_d(t_0+i\Delta t)) \left[ \frac{(\Delta t)^2}{2} I_3 \right] \Delta t I_3
\quad (2.40)
\]

where \((n+1)\Delta t = t_1 - t_0\). Here, a number of small double integrator assumptions are made, each one assuming that inputs of \( \Delta t \) seconds can safely ignore coupling effects.

**Rectilinear Dynamics Discretization**  Another approach to finding \( \Gamma \) is to use the discrete input matrix from a set of rectilinear equations of motion and rotate it into relative orbital elements

\[
\Gamma_{\text{rect}} = M(\mathbf{e}_d(t_0)) \Gamma_{\text{LVLH}}
\quad (2.41)
\]
where $\Gamma_{LVLH}$ is the discrete input matrix for a set of rectilinear equations of motion (e.g., inputs to a double integrator system, Hill’s equations, or Lawden’s equations). In Section 2.3.3, an LVLH-based version of the approximated $\Gamma$ matrix for Lawden’s equations [101] is evaluated.

**GVE-based Discretization** An alternate approach to computing $\Gamma$ is to use the continuous GVEs by taking the matrix exponential of the continuous matrix $A^*$ from Eq. (2.7)

$$
\Gamma_{GVE} = e^{A^*(t_1-t_0)}B(e_d(t_0))
$$

(2.42)

where $B$ is the GVE matrix. This approach assumes the effects of $J_2$ on the input matrix are negligible.

**Validating $\Gamma$** The continuous integration computation of $\Gamma$ in Eq. (2.37) should not require any additional validation beyond the verification that the linearization assumptions in the component matrices $D$, $\Phi^*$, and $M$ are valid. To compare the matrices calculated using Eqs. (2.42), (2.40), and (2.41) their norm can be divided by that of the matrix generated using Eq. (2.37) ($\Gamma_{true}$) to find the normalized error

$$
\epsilon_{disc} = \frac{\|\Gamma_{true} - \Gamma_{approx}\|_2}{\|\Gamma_{true}\|_2}
$$

(2.43)

where $\Gamma_{approx}$ is the $\Gamma$ matrix computed using one of the approximate methods. If the error $\epsilon$ is kept sufficiently low (typical cutoff might be 0.01), then the approximate method would be considered valid. Figure 2-3 shows the values of $\epsilon$ computed for the three approximate methods (the continuous integration method is taken as the true $\Gamma$). Timestep increments are used in Figure 2-3, but an alternate validation method using true anomaly would be appropriate if steps of true anomaly are being used for plan implementation [25].

In Figure 2-3, “GVEs w/o $J_2$” refers to the $\epsilon$ for $\Gamma_{GVE}$, “Discrete Approximation (n=50)” refers to the $\epsilon$ for $\Gamma_{disc}$, and “Lawden LTV Method” refers to the $\epsilon$ for $\Gamma_{rect}$. The methods using Eqs. (2.41) and (2.42) are significantly more accurate than the
discrete approximation. This difference can be corrected by refining the discretization timestep, however, Table 2.1 shows that the discrete method using \( n = 50 \) already requires more computation time to evaluate. Hence, in the LEO orbit examined, the GVE- and Lawden-based approximations are both faster to compute and more accurate than the discrete approximation method for all discretization times. Although the methods in Eqs. (2.41) and (2.42) are marginally less accurate than the continuous integration, they are, respectively, approximately 25 and 625 times as fast to compute.

Figure 2-4 shows how the evaluation of \( \Gamma \) using Eq. (2.42) degrades as the discretization time step is increased for the highly eccentric orbit \((e \approx 0.8)\) case examined in Section 2.2. In the figure, \( \Delta \Gamma \) refers to the difference between the \( \Gamma \)'s calculated using Eqs. (2.40), (2.41), and (2.42), respectively. For each time step, a series of \( \Delta \Gamma \) matrices are evaluated and the matrix with the largest induced 2-norm is used to rep-
Fig. 2-4: Difference between integrated and approximated $\Gamma$ for different discretization times using the HEO orbit in Eq. (2.16).

represent the discretization error. Figure 2-4 indicates that the 86 second time step used in the simulations in Section 2.6 is associated with just over 2% error between the $\Gamma$ matrices for the GVE-based calculation method. As the timestep grows larger, the discrete approach to computing $\Gamma$ becomes marginally better than the other methods, however it is still undesirable given that it is more than 250 times slower to compute than the method in Eq. (2.42).

2.4 Model Predictive Control Using GVEs

Reference [84] showed that given a valid set of linearized dynamics and a desired trajectory, a model predictive controller for a spacecraft formation can be designed that allows for arbitrarily many convex terminal and intermediate state conditions, as well as sensor noise robustness requirements. This controller is implemented on
Table 2.1: Average durations (in seconds) required to compute $\Gamma$ matrices using various methods

<table>
<thead>
<tr>
<th>Calculation Method</th>
<th>Average Computation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma_{\text{true}}$</td>
<td>1.02 sec</td>
</tr>
<tr>
<td>$\Gamma_{\text{disc}} (n=50)$</td>
<td>0.42 sec</td>
</tr>
<tr>
<td>$\Gamma_{\text{rect}}$</td>
<td>0.041 sec</td>
</tr>
<tr>
<td>$\Gamma_{\text{GVE}}$</td>
<td>0.0016 sec</td>
</tr>
</tbody>
</table>

Each spacecraft in the formation and it is using a linear programming formulation. The general form of the optimization performed by the controller is

$$\min \|U\|_1 \text{ subject to } AU \leq b$$  \hspace{1cm} (2.44)

where the matrix $A$ and the vector $b$ are formed based on the input dynamics and problem constraints and $U$ is a vector of potential control inputs vectors

$$U = \begin{bmatrix} u_i(1)^T & u_i(2)^T & \cdots & u_i(n-1)^T & u_i(n)^T \end{bmatrix}^T$$  \hspace{1cm} (2.45)

where each vector $u_i(k)^T$ is the input for spacecraft $i$ at step $k$ for an $n$ step plan.

In order to use the linearized GVE-based dynamics developed in Eq. (2.12) in the MPC formulation, the dynamics can be discretized using a zero order hold assumption according to the procedure described in Ref. [102]. To use the linearized $J_2$-modified GVE-based dynamics developed in Eq. (2.31) in the MPC formulation, the discrete dynamics in Eq. (2.36) are used. Solutions to the optimization posed in Eq. (2.44) usually take the form of classical “bang-off-bang” optimal control laws. Figure 2-5 shows a typical plan to correct a small orbital element error. Note that although only two elements begin with errors, the optimized solution requires some elements to deviate from their desired states in order to minimize overall fuel use.

Solving the optimization in Eq. (2.44) with 1000 discretization steps and a terminal constraint has always required less than 0.05 seconds on a 3 GHz computer. Formulating the matrices used in the optimization has always taken under 10 seconds, far less than the 86 second discretization time step. The time required to formulate
the problem will increase as the discretization step is made smaller and additional constraints are added. Although the computation numbers are very small, a more complicated formulation could still be implemented in a real-time system by specifying that thrusting not begin for several time steps into the plan. This will result in a plan that does not require action until some specified time in the future when it is certain that the formulation and optimization will have been completed.

Figure 2-6 shows the error between a planned trajectory using the HEO orbit and the actual implemented trajectory when a trajectory is implemented without replanning (i.e., open-loop). The norm of the final error vector is $\|\zeta\|_2 \approx 1.236 \times 10^{-5}$, which is significantly below $3.66 \times 10^{-3}$, the maximum norm of acceptable linearization error for this orbit, which was determined in Section 2.2. Thus, it is valid to use the same linearized dynamics and controller to create a new plan from this terminal
position. Repeatedly implementing new plans from a given initial error within the 
valid linearization range has always yielded terminal errors that were smaller and, 
hence, valid as initial conditions for replanning.

2.4.1 Error-Box Constraints Using Relative Orbital Elements

Several approaches have been developed to specify formation-flying mission perfor-
mance constraints. Generally, the goal of formation-flying control is to keep the 
formation from “drifting apart” and to maintain some relative geometry. This re-
quirement has been translated into maintaining orbits that have the same period 
and specifying desired relative states for spacecraft to follow. Both goals can be ac-
complished simultaneously by specifying relative desired points that have identical 
periodicity. Then to ensure that the spacecraft do not drift and that the formation 
geometry is maintained adequately, the control objective is to keep the spacecraft 
within some region around its desired point. This region is defined as a dead-band in 
Ref. [103] and similarly as an error box in Ref. [84].

Maintaining a spacecraft within an error box has several advantages over tracking 
a desired point: it does not require fuel be used to correct minor deviations from the 
desired orbit, it better captures mission constraints which typically only require satel-
lites to be in desired positions within some acceptable error, and it allows “breathing 
room” for the controller to account for modeling errors. In addition, the method 
of planning based on GVEs proposed in Section 2.4 relies on the validity of the lin-
earization analyzed in Section 2.3, which degrades as the difference between the actual 
orbital element state and the orbital element state that has been linearized increases. 
If the error box used for a particular mission is smaller than the linearity range, which 
it typically would be (see Section 2.2), the constraint that the spacecraft remain in 
the box provides an additional means of verifying that the linearity assumptions will 
be satisfied.

Several approaches can be taken to create an error box. Position error boxes are 
demonstrated in Ref. [84], which is a convenient bounding mechanism for a formation 
flying mission because it coincides well with science requirements on the accuracy of
Fig. 2-6: Error in open-loop trajectory following using the HEO orbit example. Lines indicate the difference between planned and implemented trajectories in a fully nonlinear simulation.
the formation geometry shape. When the formation geometry is specified in orbital elements, it is most convenient to use a six dimensional error box with bounds on each of the state elements. This approach, while simple and convenient for enforcing acceptable relative drift levels, does not map well into the position error box constraints typical of previous performance specifications. To transition between LVLH error states, \( x \), and relative orbital element error states, \( \zeta \), a first order rotation matrix \( M(e_d) \) is used (see Eq. (2.38)). It is possible to enforce relative position and relative velocity error box constraints using the \( M(e_d) \) matrix by formulating the optimization problem in Eq. (2.44) with constraints at every step \( k \) where it is desired that the spacecraft remain inside an error box

\[
x_{\text{min}} \leq M^{-1}(e_d)\zeta \leq x_{\text{max}}
\]  

where \( x_{\text{max}} \) and \( x_{\text{min}} \) denote opposing corners of the error box. To exclusively enforce a partial state error box (e.g., a position box), \( M(e_d) \) can be premultiplied by an additional matrix \( H \) in Eq. (2.46) to only retain the desired components of the state.

### 2.4.2 Formation Flying: Coordination Using GVEs

The model predictive controller described in Section 2.4 is designed to be decentralized, with a fully independent controller being run on each spacecraft. The controller designs trajectories that will keep a spacecraft \( i \) inside an error box centered about the spacecraft’s desired orbit, \( e_{di} \). In Section 2.2, the desired orbits are defined with respect to the actual orbit of an arbitrary satellite in the formation, \( e_1 \), using differential orbital element vectors, \( \delta e_{di} \), in the same manner used in Ref. [100]. In a system where initial conditions are chosen infrequently, it may be desirable to introduce additional coordination into the formation. When spacecraft each track desired states with no coordination, the control task is referred to as formation-keeping [84]. Alternatively, formation flying occurs when the spacecraft controllers collaborate to achieve formation-wide fuel minimization. This coordination can be achieved by calculating a central point that minimizes the overall weighted state error of each spacecraft in
the formation. Approaches to implementing closed-loop coordination of this type are presented in Refs. [97] and [104]. The virtual center approach in Ref. [97] is a centralized calculation of the error-minimizing center based on fuel-weighting and derived from measurements available through carrier-phase differential GPS (CDGPS) relative navigation of the type described in Ref. [105]. An equivalent approach can be used to find an error-minimizing reference orbit for a formation described in differential orbital elements.

Measurements from a CDGPS relative navigation system are assumed to be in the form of relative LVLH states [105], $\mathbf{x}_i$ (see Eq. (2.39)), for each satellite in the formation. The measurements will be relative to an arbitrary absolute satellite state, $\mathbf{e}_1$, in the formation, which is assumed to be at the origin of the LVLH frame. In addition to relative states, the GPS sensors on each satellite can be expected to compute a less accurate estimate of the spacecraft’s absolute state. Given an estimate of the absolute state in Earth Centered Inertial (ECI) coordinates, $\mathbf{X}_{\text{ECI}1}$ and the relative states $\mathbf{x}_i$, the differential states $\delta \mathbf{e}_i$ in Eq. (2.3) can be computed in several ways. The matrix $M(\mathbf{e}_1)$ in Eq. (2.38) could be computed and used to create a first order approximation of the relative differential element states. However, an exact conversion can be calculated by forming estimates of the absolute states of each of the satellites based on their relative measurements

$$\mathbf{X}_{\text{ECI}i} = \mathbf{X}_{\text{ECI}1} + \mathbf{x}_i$$  \hspace{1cm} (2.47)

The absolute states $\mathbf{X}_{\text{ECI}i}$ can be converted to Keplerian orbital elements, $\mathbf{e}_i$, of each satellite using a well-known procedure described in Ref. [106]. The relative measurements are then recovered in terms of differential orbital elements, $\delta \mathbf{e}_i$, using Eq. (2.3). Desired differential elements, $\delta \mathbf{e}_{dci}$, are then specified with respect to an unknown virtual center state, $\delta \mathbf{e}_c$, which is specified with respect to the absolute state $\mathbf{e}_1$. The error of spacecraft $i$ with respect to the virtual center, $\zeta_{ci}$, is given by [97]

$$\mathbf{e}_i - \delta \mathbf{e}_{dci} - \delta \mathbf{e}_c = \zeta_{ci}$$  \hspace{1cm} (2.48)
which can be placed in the standard least squares form

\[ b_i - C_i \delta e_c = \zeta_{ci}, \]

where \( b_i = e_i - \delta e_{dci}, \) \( C_i \) is a 6 × 6 identity matrix, and \( \delta e_c \) denotes the location of the virtual center with respect to \( e_1 \) in differential orbital elements. By concatenating the \( b_i, C_i, \) and \( \zeta_{ci} \) vectors for each spacecraft, the statement of error for the entire formation is written

\[ b - C \delta e_c = \zeta, \]

where \( b = [ b_1 \ldots b_n ]^T, \) \( C = [ C_1 \ldots C_n ]^T, \) and \( \zeta = [ \zeta_{c1} \ldots \zeta_{cn} ]^T. \) The solution that minimizes the error vectors globally in a weighted least squares sense is

\[ \delta e_c = (C^T W C)^{-1} C^T W b \quad (2.49) \]

where \( W \) is a weighting matrix that can be used to bias the center location according to the fuel-use rates of different satellites in the formation, as well as to weight orbital elements individually based upon the amount of control required to alter them (obtainable from the GVEs for \( e_1 \)). This calculation can be decentralized and reduces the following iterative form[96]

\[ \delta e_{ci} = b_i, \quad \delta e_{c_{i+1}} = \delta e_{c_{i-1}} + \frac{w_i}{\bar{w}_i + w_i} (b_i - \delta e_{c_{i-1}}) \quad (2.50) \]

where \( w_i \) is the weight of the \( i^{th} \) estimate, and \( \bar{w}_i = w_1 + w_2 + \ldots + w_i. \) In this formulation, a spacecraft \( i \) must pass its current state estimate, \( \delta e_{ci}, \) and the scalar \( \bar{w}_i \) to the next spacecraft for a new estimate of the optimal center position to be formed. Using this method, the error-minimizing fuel-weighted virtual center can be known in one full cycle around a formation.

### 2.5 Comparison to Other GVE-based Impulsive Control Schemes

The optimized controller developed in Section 2.4 can be applied to a range of spacecraft control problems. This section uses that controller for the specific problem of correcting state error over a finite horizon in order to compare its performance and ca-
pabilities with other methods. Gauss’ Variational Equations (GVEs) have been used
to design many Lyapunov and fixed impulse control systems [90, 91, 93, 94, 99]. Sev-
eral research groups have proposed control laws for formation-flying spacecraft that
use GVEs to design impulsive thrusting maneuvers for orbit correction. A method
of producing optimized impulsive plans for very-low eccentricity orbits is presented
in Ref. [107], but this approach does not extend to the higher eccentricities required
for MMS missions. Another method based on GVEs [94] allows optimized planning
for low Earth orbits, but only permits optimization over a single impulsive thrust,
guaranteeing that the solution will be sub-optimal in many cases. In addition, this
approach is only derived for correcting errors in semimajor axis, eccentricity, and in-
clination. Another approach to using GVEs for formation control is to derive a contin-
uous proportional-derivative controller satisfying a Lyapunov equation [90–92, 100].
Control algorithms of this type have been shown to be asymptotically stable in most
cases [92], but belong to a class of control systems that fire continuously. Continuous
firing is generally not desirable for space missions because it is often disruptive to the
science mission; it typically must be coupled with attitude maneuvers, and it expends
fuel (nonreplenishable aboard a spacecraft) continuously.

The method of formation control in Ref. [94] is based on GVEs and uses a single
corrective thrust computed using a nonlinear optimization. Although this method is
guaranteed to find the optimal single-thrust correction for an arbitrary time period,
it is not guaranteed (or likely) to find the optimal multiple-thrust correction. In
addition, this approach is restricted to use in low Earth orbits and is only designed to
correct errors in semimajor axis, eccentricity, and inclination. An approach presented
in Ref. [108] uses a pseudo-inverse to the GVE control effect matrix to calculate a
single corrective impulse. This approach is not guaranteed to be fuel-optimal for any
cases and is not accurate for correcting position errors.

Ref. [93] describes a controller which uses four impulses over the course of an orbit
to correct arbitrary orbital element perturbations. Because of its more general appli-
cability, this section will compare that approach to the MPC controller presented in
Section 2.4. Both methods are designed to drive the elements of a state error ζ to zero
over a fixed time interval. The four-impulse approach has not been presented in the context of performance criteria (e.g., trajectory or terminal error boxes, robustness to disturbances) or constraints (e.g., maximum thrust level), so the comparisons in this section will use an MPC controller formulation that minimizes fuel use while driving the error state to zero in a fixed time and has no other constraints. In addition, for the purposes of comparison, no $J_2$ effects are used in either planning approach.

The algorithm in Ref. [93] can be summarized in four steps to be taken over the course of an orbit. When the argument of latitude, $\theta$, is 0 or $\pi$ radians, implement a velocity change (impulsive thrust), $\Delta v_{h_i} = [h/(r \cos \theta)] \Delta i$, in the cross-track direction of an LVLH frame centered on the spacecraft to cancel the inclination error component of $\zeta$. When the argument of latitude, $\theta$, is $\pi/2$ radians, implement a velocity change, $\Delta v_{h\Omega} = [h \sin i/(r \sin \theta)] \Delta \Omega$ in the cross-track direction to cancel the ascending node error. At perigee and apogee, implement $\Delta v_{r_p}$ and $\Delta v_{r_a}$, respectively, in the radial direction to cancel the argument of perigee and mean anomaly errors

\begin{align*}
\Delta v_{r_p} &= -\frac{na}{4} \left( \frac{(1+e)^2}{\eta} \left( \Delta \omega + \Delta \Omega \cos i \right) + \Delta M \right) \tag{2.51} \\
\Delta v_{r_a} &= \frac{na}{4} \left( \frac{(1-e)^2}{\eta} \left( \Delta \omega + \Delta \Omega \cos i \right) + \Delta M \right) \tag{2.52}
\end{align*}

Also at perigee implement $\Delta v_{\theta_p}$ and at apogee implement $\Delta v_{\theta_a}$ in the in-track direction, to cancel the semimajor axis and eccentricity errors

\begin{align*}
\Delta v_{\theta_p} &= \frac{nan\eta}{4} \left( \frac{\Delta a}{a} + \frac{\Delta e}{1+e} \right) \tag{2.53} \\
\Delta v_{\theta_a} &= \frac{nan\eta}{4} \left( \frac{\Delta a}{a} - \frac{\Delta e}{1-e} \right) \tag{2.54}
\end{align*}

Using the notation and the HEO reference orbit from Section 2.2, the following example compares the MPC method with the control approach reviewed in this section. Note that in comparison to the MPC approach, the four impulse method is very simple to implement. However, the two approaches have different rates of fuel use for
identical tasks. For the state error

\[ \zeta = \begin{pmatrix} 10^{-9} & 10^{-7} & 10^{-7} & 10^{-7} & 10^{-7} & 10^{-7} \end{pmatrix}^T \]  

(2.55)

the 4-impulse method requires 1.42 mm/s of fuel to correct the state error over the course of an orbit and the MPC method requires 0.549 mm/s of fuel. In this example, the model predictive controller was given a full orbit time horizon. However, the same control objective could have been achieved in less time, but using more fuel.

A series of 1000 orbital element state error vectors, \( \zeta \), were generated in which each perturbed element was a random number between \( \pm 10^{-6} \). For each of the error vectors, both control methods were used to generate plans for eliminating the error. On average, the MPC maneuvers only required 51% of the fuel required by the 4-impulse maneuver. Further controller comparisons are presented in Section 2.6.

### 2.6 Formation Maintenance on MMS-like Mission

The control system described in Section 2.4 was demonstrated on a segment of a mission similar to MMS, which is comprised of four spacecraft that create a regular tetrahedron geometry once per orbit near apogee to perform science observations. The orbits of the four spacecraft are widely separated and highly elliptical, presenting a challenge for many optimal formation specification and control approaches in the literature [25, 109]. Using the tetrahedron initial-condition optimization approach in Ref. [96] and the model predictive approach in Section 2.4, the four spacecraft were controlled in a fully nonlinear simulation with Earth oblateness effects, atmospheric drag effects, and other realistic disturbances using a commercial orbit propagator [112]. The control objective in this simulation is to achieve a set of tetrahedron initial conditions once per day near the formation orbit apogee. The reference orbit for formation is the highly eccentric orbit used in Example 2 of Section 2.2.

In order to implement the MPC scheme in Section 2.4 using the dynamics developed in Section 2.3, the approximate rectilinear method of calculating \( \Gamma \) is used. The
Fig. 2-7: Forming and maintaining a 1000 km (at apogee) tetrahedron formation in a highly eccentric orbit ($e \approx 0.8$) in the presence of $J_2$

dynamics matrices $\Gamma$, $\Phi$, and $D$ are all functions of the desired orbital elements, which are parameter-varying. To obtain accurate trajectories for the absolute desired orbital elements for the formation, they are integrated numerically with $J_2$ disturbance effects included and then used to generate the linear propagation matrices used in the optimization. The time step used in the simulation is 86 seconds, providing approximately 1000 discretization points in each daylong orbit. Constant time step duration was chosen to approximate typical flight computer operation, but plans with varying time steps can also be designed by planning in increments of true anomaly, and Ref. [25] discusses how these can be implemented. The planning horizon length for this simulation is one full day.

Figure 2-7 shows the rate at which fuel was used over the course of one week of formation flying. The formation fuel use rate converges to approximately 12.1 mm/s per day ($\approx 1$ orbit) for each satellite. Note that all previous simulations had to be
Fig. 2-8: Forming and maintaining a 1000 km (at apogee) tetrahedron formation in a highly eccentric orbit \((e \approx 0.8)\) in the presence of \(J_2\).

limited to the case where the \(J_2\) effects were disabled in order to ensure that the formation remained stable. In that case, for the same configuration, but without the effects of \(J_2\) included in the controller dynamics or the simulation dynamics, the results showed an average \(\Delta V\) of 11.5 mm/s per satellite per orbit. These nearly equivalent fuel use rates for simulations with and without \(J_2\) indicate that linearized modeling of \(J_2\) effects in a controller of the type presented in Section 2.4 is sufficient to prevent the disturbances from dominating fuel usage. The state error for one of the spacecraft in the formation is seen being driven to the origin in Figure 2.6. Trajectories followed during this simulation fall within the range of acceptable state error determined for the linearizing assumptions used in Section 2.3.
2.7 Summary

A variant of Gauss’ Variational Equations that incorporates the effects of $J_2$ was used to derive a set of linearized relative dynamics of orbital motion and extend previous work on planning-based controllers. This choice of a linear parameter varying (LPV) dynamics model to design the controller allows a compromise between simple, but inaccurate, linear models (e.g., Hill’s equations) and high fidelity, but often difficult to control, nonlinear models. In particular, by accounting for $J_2$ disturbances in the dynamics, the planning controller can exploit these dynamics for improved fuel efficiency. The linearization assumptions used in the approach were shown to be valid for typical spacecraft error box sizes. The LPV model was used in a model predictive controller (MPC) and the combination was shown to be more fuel-efficient than a previously published technique. The overall controller ($J_2$-modified GVE-based dynamics embedded in the MPC controller) was also used to specify and control a large (1000 km sides at apogee) tetrahedron-shaped formation in an MMS-like orbit for a period of twenty days using a commercial propagator with realistic disturbances. The results showed that the controller is reliable and that formation flying using this MPC with $J_2$-modified GVEs requires fuel use that is comparable to using unmodified GVEs in simulations that do not include the $J_2$ effects.
Chapter 3

Fuel-optimized Semi-$J_2$-invariant Initial Conditions

Section 2.4 presented a model predictive controller that can be used to create optimized plans for relative orbit control in the presence of $J_2$ disturbances. In a spacecraft formation, it is critically important both to conserve fuel when maneuvering and to maneuver to a state that will, over time, conserve fuel. The latter is an initial condition (IC) problem, the specifications of which will depend on the unique requirements of a particular mission. However, in any spacecraft formation, a primary goal will be to prevent the vehicles from drifting apart, since that will typically end the mission. If the spacecraft in a formation tend to drift apart, then periodic maintenance maneuvers will be required to restore the formation. Initial conditions are called invariant if they eliminate drift, thereby allowing spacecraft to maintain their relative orbits without expending fuel. In the context of this thesis, invariance does not necessarily imply any form of relative boundedness at other points along the reference orbit, but this could be addressed by including additional error box constraints to the optimization developed in this section.

For purely Keplerian orbits, invariance translates into a requirement that all spacecraft in a formation have the same semimajor axis. For example, this requirement was solved for analytically using Lawden’s equations of motion in Ref. [109]. However, Earth oblateness effects ($J_2$) make orbits based on the Keplerian invariance solution
drift apart. In fact, when the effects of $J_2$ are considered, very few perfectly invariant orbits exist. Hence, it is more common for a $J_2$ “invariant” orbit to instead be truly invariant only in several dimensions where it is possible to cancel the relative effects of $J_2$. Analytic conditions based on this partial invariance have been introduced [20, 110]. The following presents an alternate approach that uses the dynamics in Section 2.2 and Ref. [98] in a convex linear optimization to find initial conditions that balance the objective of not drifting in the presence of relative $J_2$ effects against the objectives of minimizing the fuel use required to achieve these initial conditions and retaining a specified geometry for the formation.

### 3.1 Formulation

To begin, specify that orbits are invariant if their relative orbital offset, $\delta e$, in Eq. (2.5) remains unchanged over a period of time, so that $\delta e(t_1) \equiv \delta e(t_2)$, where $t_2 - t_1$ is the duration of interest (typically an integer number of orbits). Then, using the state transition matrix from Eq. (2.34) gives the constraint

$$ \delta e(t_1) = \delta e(t_2) = D^{-1}(e(t_2)) \Phi^*(e(t_1), t_2, t_1) D(e(t_1)) \delta e(t_1) $$

(3.1)

Defining the matrix function $\Phi^*_{Dk} \equiv D^{-1}(e(t_{k+1})) \Phi^*(e(t_k), t_{k+1}, t_k) D(e(t_k))$ gives the invariance condition

$$ \delta e(t_1) = \Phi^*_{D1} \delta e(t_1) \to (\Phi^*_{D1} - I) \delta e(t_1) = 0 $$

(3.2)

where $I$ is a $6 \times 6$ identity matrix. As mentioned above, the resulting geometry of the no-drift (complete invariance) condition is too restrictive for many missions, but partially invariant conditions can be obtained by minimizing the weighted norm of the invariance condition

$$ \min_{\delta e(t_1)} \| W_d (\Phi^*_{D1} - I) \delta e(t_1) \| $$

(3.3)
where the weighting matrix $W_d$ is introduced to extract states of interest to penalize particular types of drift. Note that if $W_d$ is the matrix $M(e(t))$ (see Eq. (2.37)), which rotates the differential osculating elements $\delta e$ into an LVLH frame, then the elements of the LVLH state can be directly penalized (e.g., extracting only position states could penalize meters of drift). This enables the drift formulation to penalize the distance from the desired geometry in a Cartesian frame, as opposed to just using orbital elements. Penalizing true separation distance finds initial conditions that will maintain the formation shape, an important consideration for missions that require specific geometric configurations [86].

The overall problem statement then is, given a spacecraft at offset $\delta e(t_0)$, design a control input sequence $U(\tau)$, $\tau \in [t_0, t_1]$ that generates a set of initial conditions at $t_1$ that balances the trade-off between the ensuing drift by time $t_2$, the fuel cost of achieving these initial conditions, and the extent to which the formation geometry is maintained. The proposed approach is illustrated in Figure 3-1. The two diagrams in Figure 3-1 show local frame views of relative orbital motion. One satellite is fixed at the origin and the other (represented as a circle) is shown at several different times. On the left, the satellite is pictured at some initial time $t_0$ and again after some maneuver at time $t_1$. The open loop propagation of the satellite from $t_0$ to $t_1$ that would occur absent the maneuver is also shown (hatched circle). The right figure shows the drift that would occur starting from each state at $t_1$. The case state at $t_1$ after a maneuver experiences less drift than the open-loop case, but it required more fuel. Also, the dashed lines show the geometry cost of moving the satellite away from the desired initial conditions. Thus, the three times in the problem are $t_0$, when the maneuver is initiated and fuel is penalized, $t_1$ when the geometry cost is penalized, and $t_2$ when drift is penalized. The semi-invariant initial condition optimization cost function is

$$
C^* = \min_U \{ Q_d \| W_d(\Phi_{D1}^* - I)(\Phi_{D0}^*\delta e(t_0) + \hat{\Gamma}U)\| + Q_x\| W_x\hat{\Gamma}U\| + Q_u\| U\| \} \quad (3.4)
$$

where $C^*$ is the optimal cost, $W_x$ is a weighting matrix to specify the type of geometry.
penalty, $Q_u$ is a weighting on fuel minimization, $Q_x$ is a weighting on desired formation geometry, and $Q_d$ is a weighting on drift. Using Eq. (3.1), $\delta e(t_1) = \Phi D_0 \delta e(t_0) + \hat{\Gamma} U$, where $\hat{\Gamma}$ is a row of convolved $\Gamma$ matrices (see Eq. (2.35)) that propagate the effects of a vector of inputs ($U$) (see Eq. (2.45)) at each time step of the maneuver [84]

$$
\begin{bmatrix}
\Phi(n, 1) \hat{\Gamma}(0) & \Phi(n, 2) \hat{\Gamma}(1) & \cdots & \Phi(n, n - 1) \hat{\Gamma}(n - 2) & \hat{\Gamma}(n - 1)
\end{bmatrix}
$$

(3.5)

where $\Phi(k, j) \equiv D^{-1}(e(kt_s))\Phi^*(e, kt_s, jt_s)D(e(jt_s))$, $\hat{\Gamma}(k) \equiv \Gamma(e, (k + 1)t_s, kt_s)$, and $t_s$ is the discretization time step. The cost function uses the initial state of each spacecraft in the formation as the desired geometry, so the geometry weighting penalizes deviations from the open-loop state propagation. Note that a simple modification to the cost function could separate the initial geometry from the desired geometry. The optimization in Eq. (5.6) can be easily implemented as a linear program if 1-norms are used, permitting efficient, fast online solutions [111]. As expected, a sufficiently high weighting on invariance results in a minimizing control input $U^*$ where $\hat{\Gamma} U^* = -\Phi D_0 \delta e(t_0)$. Alternately, a sufficiently high $Q_x$ (with an identity matrix for $W_x$) results in $\hat{\Gamma} U^* = [0 0 0 0 0 0]^T$, because the inputs will all be zero in order to maintain the original geometry.
Fig. 3-2: Expected drift and fuel cost of a range of optimized initial conditions in a LEO orbit. The lines between marked points indicate the least drift that can be attained for the indicated amount of fuel. In this example the fuel weighting is 1, the geometry weighting is \(10^{-18}\), and the drift weighting is allowed to vary widely. The ■ represents the solution based on the period-matching condition in Case 1 of Ref. [110]. The ♦ represents the solution based on the \(J_2\) invariant orbit with perigee drift in Ref. [20].

### 3.2 Results

Figure 3-2 shows drift rates and fuel costs for a series of initial conditions generated by the optimization method with a half orbit planning horizon as \(Q_d\) is changed. For this example, the 1-norm is used to penalize both drift and fuel use and both \(W_d\) and \(W_x\) are set to the position rows of \(M\) in order to only penalize position drift and geometry separation. With a very low \(Q_d\), \(Q_u\) will dominate, resulting in no control use. The zero drift point corresponds to high fuel use, because it necessitates driving the spacecraft to nearly the same orbits. A range of possible optimized initial conditions lie in between those extrema. The high-drift, unmodified initial conditions lie very close to the vertical drift axis, but drop to just over 0.5 m of drift with a minimum of fuel use. Further drift reductions are possible, but at greater fuel cost. It
Fig. 3-3: Expected drift and geometry cost of a range of optimized initial conditions in a LEO orbit. The lines between marked points indicate the least drift that can be attained for the indicated amount of separation from the desired orbital offset. The ■ represents the solution based on the period-matching condition in Case 1 of Ref. [110]. The ♦ represents the solution based on the $J_2$ invariant orbit with perigee drift in Ref. [20].

is readily apparent from the graph that using additional $\Delta V$ will produce diminishing returns in terms of reducing drift.

Initial conditions generated by other $J_2$-invariant conditions should lie either on or above the optimized result. The ■ in Fig. 3-2 represents the initial condition based on the $J_2$ invariance condition in Case 1 of Ref. [110] that requires no mean period drift. The point is nearly optimal for this example, but this is not guaranteed to be the case for other problems. The ♦ indicates the drift that occurs when using semi-invariant initial conditions that allow perigee drift [20]. In both specific cases, the partial invariance conditions allow for a range of possible initial conditions. Both analytic cases occur near the same drift levels as the initial conditions found by the optimizing approach. This indicates the optimized ICs may, in those cases, be meeting the same invariance criteria, while simultaneously finding ICs that minimize
fuel use. The optimization-based approach enables the identification of a range of fuel-optimized initial conditions that can be used to better meet the requirements of a specific mission.

Figure 3-3 shows a number of optimizations of the same orbit and desired offset, however, in this case, both $Q_x$ and $Q_u$ are made significant while $Q_d$ is varied. The figure shows the cost associated with changing the formation geometry ($||W_x \hat{\Gamma} U||_1$) versus the fuel cost ($||U||_1$). When $Q_x$ is very high relative to $Q_d$, the optimized ICs, $\delta e(t_1)$, are equivalent to the open-loop propagation of $\delta e(t_0)$ (i.e., no fuel is used). This corresponds to $12\text{m}$ of drift over an orbit. As $Q_d$ is increased, the optimized ICs are farther from $\Phi_{D0}^* \delta e(t_0)$, but the resulting drift is lower because $Q_d$ has a greater effect on the solution. The solution that achieves $0.5\text{m}$ of drift with almost no geometry cost represents a compromise between the desired formation geometry and the drift resulting from the effects of relative $J_2$. At the cost of slightly repositioning the formation (velocity changes are not penalized in the $W_d$ used for this example), the drift over an orbit has been reduced by $4\text{m}$. When invariance dominates (the lower-right corner of the figure), the optimized initial conditions cancel almost all of the orbital offset $\delta e$, indicating that geometry goals have been ignored.

Figures 3-4 and 3-5 show the effect of independently varying the fuel and geometry weights for an MMS-like HEO orbit

\[
e_A(t_0) = \begin{bmatrix} 6.6 & 0.82 & 0.17 & 6.28 & 6.28 & 3.14 \end{bmatrix}^T
\]  (3.6)

with the desired geometry

\[
\zeta(t_0) = \begin{bmatrix}
-4.00000 \times 10^{-8} \\
-4.98086 \times 10^{-7} \\
-8.50000 \times 10^{-7} \\
2.30000 \times 10^{-7} \\
-7.33646 \times 10^{-7} \\
3.72778 \times 10^{-6}
\end{bmatrix}
\]  (3.7)
As in the LEO case, the unmodified initial conditions experience a significant amount of drift, which can be greatly reduced using comparatively small fuel expenditures. Likewise, small (centimeter) changes in the formation geometry can also decrease the drift. In this case, more intermediate optimized geometry steps exist, allowing for additional choices between precisely achieving the desired shape and drifting out of that shape over the course of the next orbit.

Figure 3-6 shows an example in which a spacecraft trajectory is simulated using a commercially available fully nonlinear propagator that accounts for the effects of many realistic disturbances, including $J_2$, drag, and solar pressure. The reference orbit has a semimajor of 1.08 Earth radii, an eccentricity of 0.03, and an inclination of 45 degrees. The two line show two possible relative trajectories for a spacecraft initialized in a 2 km in-track formation. The solid line shows initial conditions that are chosen by creating a 2 km separation in-track separation in LVLH coordinates and then adjusting the velocity to eliminate differential semimajor axis. The dashed line shows the trajectory which results from using the initial conditions given as outputs from the optimization method presented in this chapter when the fuel penalty is zero and the geometry and drift penalties are weighted equally. In this case, the optimized initial conditions begin 12 m away from the desired 2 km mark, well within the 100 m error box that would be used for a 1 km (at apogee) baseline formation. The drift reduction for this tradeoff was on the order of 400 m per day. In the simulation shown in Figure 3-7, all realistic disturbances are modeled except $J_2$ and the same initial conditions produce significantly different results. In this case, SMA-corrected trajectory (solid line) creates a path with almost no drift and the $J_2$-corrected optimized trajectory now produces drift. This drift is caused by the $J_2$ correcting SMA offsets that the optimizer produces, which become a source of drift in a simulation without $J_2$ perturbations.
Fig. 3-4: Expected drift due to differential $J_2$ effects and fuel cost of a range of optimized initial conditions in a HEO orbit. The lines between marked points indicate the least drift that can be attained for the indicated amount of fuel.

Fig. 3-5: Expected drift due to differential $J_2$ effects and geometry cost of a range of optimized initial conditions in a HEO orbit. The lines between marked points indicate the least drift that can be attained for the indicated amount of separation from the desired orbital offset.
Fig. 3-6: Relative satellite separations for a vehicle initialized using analytic invariance and a vehicle using optimized invariance. Trajectories generated using realistic nonlinear simulation including the effects $J_2$.

Fig. 3-7: Relative satellite separations for a vehicle initialized using analytic invariance and a vehicle using optimized invariance. Trajectories generated using a nonlinear simulation that did not incorporate the effects $J_2$. 

not
3.3 Summary

This chapter introduced a new approach to creating invariant relative orbits. The method uses linear optimization to find initial conditions that minimize drift, but also maintain a desired geometry and minimize the fuel use required to attain the initial conditions. This approach represents a hybrid between trajectory planning and formation design that balances the common approach of finding ideal drift-minimizing initial conditions against the practical need to not waste fuel in an effort to reduce fuel use. Multiple definitions of drift over arbitrary time-frames are considered including mean drift, osculating drift, and Cartesian drift. The range of solutions produced by the new technique was examined for several reference orbits and compared to analytic approaches.
Chapter 4

Formation Flying Simulations

4.1 Introduction

Using the dynamics introduced in Chapter 2 inside a model predictive controller, it is possible to demonstrate closed-loop control subject to constraints for widely separated missions with coupled control/performance objectives and multiple large formation reconfigurations.

4.2 Mission Description

A series of nonlinear simulations of portions of the Mitsubishi Electric Company (MELCO) reference mission [113] (see Figure 4-1) were performed using the FreeFlyer™ orbit simulator [112] in order to demonstrate the effectiveness of the controller/dynamics combinations in Chapter 2 for controlling a realistic satellite formation through all planned stages of operation. A commercial orbit propagator, FreeFlyer™, was used to propagate orbits in an absolute frame for both satellites. Propagation included the effects of many realistic disturbances, including drag, lift, solar radiation pressure, and $J_2$. The propagator interacts with the controller through a MATLAB™ interface. During each propagation step, the control algorithm is queried. If the controller is currently implementing a thrusting plan, the thrusts corresponding to the current position in the plan are converted into appropriate orbital element offsets and re-
turned to the propagator. After each plan is completely implemented, a new plan is created.

The simulation presented in the following sections involves a three satellite formation. The reference orbit, represented by a virtual satellite with properties similar to the average of the fleet, has a semi-major axis of 6900 km, inclination 45°, and eccentricity 0.003. Realistic disturbances (drag, \( J_2 \), solar radiation pressure, Sun/Moon effects) were included in all simulations.

Each satellite is modeled as on the current specifications for the MELCO formation flying mission [113]. Each satellite has a mass of 900 kg and a ballistic coefficient of 0.4. The satellite thrusters are restricted to provide a maximum of 2 Newton of force over a 10.8 second time step. Lawden’s time-varying equations are used to determine the desired state for each spacecraft, however Hill’s equations are used in LP problem. This is consistent with the observation in Ref. [84] that slightly eccentric orbits (0.0005 < \( e \) < 0.01) require eccentricity-invariant initial conditions, but not time-varying dynamics. Ten minutes of each orbit are reserved for observations: during this time position constraints are enforced, but no thrusting is permitted.

The MELCO formation flying mission consists of two different formation shapes: 1) in-track separation formations and 2) in-track/cross-track passive aperture forma-

\[ Fig. 4-1: \text{Concept for MELCO formation flying mission [113]} \]
tions (triangular). Over the course of a mission, the formation is designed to achieve four different configurations in the following order:

1. In-track formation with 1 km (at apogee) of separation between spacecraft (represents deployment configuration)

2. Passive aperture formation with 50 m baseline

3. Passive aperture formation with 500 m baseline

4. Passive aperture formation with 5 km baseline

For each type of formation at each baseline, one or more 18 day simulations were conducted to determine the average fuel usage. The mission requirements [113] specify that each formation size must use an error box that is 10% of the baseline.

### 4.3 Simulation Controller Configuration

The planning controllers used in the simulations in this Chapter use a modified form of the basic problem statement in Chapter 2. Each spacecraft designs its control individually and coordination between spacecraft is accomplished through the design of the formation desired states at the time the formation is initialized. Error box constraints of the type in Ref. [84] are added at every 6 time steps to ensure that satellites achieve their performance objectives, while reducing the computational burden of imposing them at every time step. In addition, fuel inputs are permitted every 6 time steps for station-keeping (in which spacecraft are tasked to remain in formation) and every time step for formation maneuvers (those periods during which the formation is being switched from one configuration to another). Mission requirements [113] specify that a ten minute time window at apogee should be reserved during formation-keeping maneuvers in which no thruster inputs are permitted.

An always-feasible formulation was used for the simulation controller, guaranteeing that a plan is always returned, even if no feasible solution exists which satisfies the error box constraints. The always-feasible formulation is based on the constraint
enlarging approach presented in Ref. [84], but with a modification allowing several degrees on constraint violation. If no feasible solution is possible for the desired error box, the approach in Ref. [84] enlarges the error box as little as possible until a feasible solution is found. However, the resulting plan may end with the spacecraft outside the nominal box, which would guarantee that the next plan would use as much fuel as necessary to return the spacecraft to the error box on the first step of the plan or would also require an expansion of the nominal box. The always-feasible formulation used in this section also enlarges the error box until a feasible plan is found, but includes another soft constraint which prefers that the planned trajectory still end inside the nominal error box. This additional constraint is also implemented using with an additional high cost-penalty variable that is used to ensure that it is only relaxed in the event that no feasible solution exists that enlarges the error box and ends inside the nominal error box. The modified always-feasible formulation allows the controller to prefer plans that make future optimizations initially feasible.

4.3.1 Parameters Examined

An initial simulation of the mission in Sec. 4.2 using the parameters and specifications in Sec. 4.3 was conducted. The results of that simulation indicated that levels of CDGPS noise expected were excessive for the station-keeping in a 50 m passive aperture formation with $5 \text{ m} \times 5 \text{ m} \times 5 \text{ m}$ error box, the most tightly constricting phase of the mission. The controller was unable to keep the satellite constrained to its error box under those conditions and experienced errors that grew over time. In order to investigate this phenomenon and find a stable configuration for the controller that met as many mission requirements as possible, a number of variations of the basic controller setup in Sec. 4.3 have been implemented and tested in realistic simulations. Section 4.4 presents the results of those simulations and evaluates the effectiveness of the parameter variations in terms of effectiveness at preventing error box violations and average fuel use. The control parameters considered are:

Thrusting During Observations
The MELCO mission specifications indicate that passive aperture radar observations cannot occur when any spacecraft in the formation is thrusting. As a result, the basic configuration in Sec. 4.3 prevents thrusting during a 10 minute period at the apogee of every orbit that the formation is in the passive aperture configuration. However, the MELCO mission reference orbit is 95 minutes, so the effect of the observation thrusting prohibition is to reduce the overall control authority of any plan by more than 10%. Although, this requirement is hard constraint for the mission, it is included as a parameter in the simulation study so that its effects formation flying mission performance can be judged.

**Error Boxes Relaxed When Not Observing**

One of the original concepts for the MELCO formation flying mission [113] specified that the formation should be in a passive aperture during periods when the formation taking observations and transition to a widely separated during periods when observations are not needed. The maneuver generation analysis in Appendix A indicates that there is generally no fuel advantage to formation-keeping in an in-track formation versus a passive aperture formation and that the cost of maneuvering into and out-of holding configurations is unnecessary.

Given that the tight error box requirements are derived from the needs of the distributed observation instrument, a reasonable modification of the mission requirements is to enlarge the error boxes during the 85 minutes of each orbit in which no observations are taking place and only enforce tight performance constraints when they are strictly needed. This constraint is implemented by doubling the error box size during the time steps outside the 10 minute period reserved for observations at apogee.

**Dynamically Motivated Error Box Shapes**

Because the 50 m passive aperture is the most fuel-intensive of the formation configurations, a $5 \times 10 \times 5$ meter (radial/in-track/cross-track) error box is examined as an alternative to the 10% requirement for that portion of the mission. The $5 \times 5 \times 5$ meter error box is sufficiently small that the navigation errors [73]
strongly influence the closed-loop behavior. Enlarging the in-track dimension allows a slightly more natural relative elliptical motion in the radial/cross-track plane (typically a $1 \times 2$ ellipse). This observation is derived from the form of Hill’s equations, in which the coefficients of the harmonic terms for the in-track axis are exactly twice the value of the coefficients of the harmonics terms for the radial axis.

**Invariance Terminal Condition**

A plan that is optimized to guarantee that a spacecraft remains inside an error box over some future horizon will accomplish that goal, but will not provide any guarantees for the future behavior of the spacecraft. Occasionally, a situation may arise where the spacecraft would approach its constraint boundary in the near future after the end of a plan. This situation is prevented from resulting in a constraint violation by the creation and implementation of a new plan. However, if the new plan is forced to react quickly to a potential constraint violation in the near term, the only feasible solution may require a great deal of fuel. In an effort to reduce the need for vehicles to make short term “emergency” corrections, an invariant set terminal condition was examined. In this case, the condition guarantees that after a plan is ended, the spacecraft will naturally (i.e., with no thrust inputs) remain inside its box for a full orbit and return to its state at the time the plan ended. Within the time span that the dynamics can accurately propagate the states, this terminal constraint guarantees perpetual collision avoidance in the absence of state knowledge uncertainty. This constraint is imposed using the nominal estimate of the spacecraft state, because a implementation is generally infeasible.

**4.4 Simulation Results**

Multiple simulations were performed to study the MELCO formation flying mission and the effects of the control system parameters introduced in Sec. 4.3. These simulations are described in Table 4.1.
Table 4.1: Fuel use results for formation flying simulations. Fuel costs for station keeping (SK) are given in mm/s/orbit/satellite and fuel costs for maneuvers (Mvr) are given in m/s for the entire formation.

<table>
<thead>
<tr>
<th></th>
<th>50 m PA: 5 x 5 x 5 m Error Box</th>
<th>50 m PA: 5 x 10 x 5 m Error Box</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sim 1</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sim 2</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Sim 3</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sim 4</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sim 5</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Sim 6</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In Figs. 4-3-4-20, the vast majority of the trajectories remain inside the error boxes. Several instances of error box constraint violations occurred as a result of the always-feasible formulation, but were quickly corrected. Several general trends are visible: in most cases the 50 m passive aperture requires the most fuel to maintain and the 500 m passive aperture requires the least fuel to maintain. Also, enlarging the in-track dimension of error box tended to reduce fuel use a great deal in comparable simulation configurations (i.e., Sim 1 & Sim 4 and Sim 3 & Sim 6).

Interestingly, the least restrictive error box (5 km) did not result in the lowest fuel use. This phenomenon is likely due to the reduced likelihood of encountering constraints in a single orbit planning horizon for the larger error box. As a result, it is possible for a spacecraft to enter into a large, high speed relative orbit that may end near an error box constraint. In those situations, future trajectory optimizations would need to take immediate corrective action to avoid constraint boundaries. This hypothesis is supported by significant reductions in fuel use for the 5 km passive aperture between Sim 1 & Sim 3 and Sim 4 & Sim 6, even with fewer thrusting times available in Sim 3 and Sim 6 due to constraints against thrusting during observations. In both cases, the only difference between the simulations is the inclusion of the terminal invariance constraint and a restriction against thrusting during observations.
The restriction against thrusting tends to increase the fuel use, as in the Sim 4 & Sim 5 pair. In Sim 4, thrusting is permitted at all times in the orbit, whereas in Sim 5 10% of the orbit is reserved for observations. In the 50 m and 5 km configurations, the 10% restriction increased fuel use by more than an order of magnitude for those simulations. It is likely the that difference between the fuel use numbers for the 500 m configuration is insignificant due to random elements in the simulations.

The addition of the terminal invariance constraint in Sim 6 lowers the fuel consumption for the 5 km configuration, but raises it for the 50 m and 500 m baselines compared to Sim 4. At the 5 km level, the reduction is greater than a factor of 3. Figures 4-14 and 4-20 show the error box motion for all three satellites in the formation for station-keeping in the 5 km formation for Sims 4 and 6, respectively. While Figure 4-14 shows a considerable amount of motion throughout the entire error box, Figure 4-20, shows that the spacecraft actively controlled in the center of the error box. These results are counterintuitive for several reasons: 1) the addition of constraints to an optimization would typically be associated with increased fuel cost; and
2) expending fuel to keep a spacecraft in a small box would usually be more expensive than allowing the same spacecraft to stay in a large box. The fact that the fuel costs were reduced is not in conflict with the fact that individual optimizations should have increased cost; the fuel costs measured are the steady state closed-loop levels of consumption, as opposed to expected maneuver costs. Also, note that the fuel use in Sim 4 remained high for a period after the maneuver to the 5 km configuration before settling into a steady state, whereas the Sim 5 fuel use settled almost immediately. This is likely because the terminal invariance condition forces the trajectories to enter closed ellipses. In Sim 4, where no invariance condition was specified, the spacecraft did not start in a closed ellipse, but did eventually enter one and not exit for the duration of the simulation. This is because even without requiring a closed-ellipse, the optimization recognizes that there is no need to expend fuel to change a trajectory which will result in no constraint violations. An almost identical pattern is visible between Sim 1 (which does not impose terminal invariance) and Sim 3 (which uses terminal invariance).

Terminal invariance raises the fuel cost between Sims 1 & 3 and Sims 4 & 6 for the 50 m and 500 m configurations. At the 50 m level, this effect is complicated by the fact that the increase is small and Sims 3 and 6 both also restrict thrusting at apogee. Sims 1 and 4 were found to not be feasible with thrust restrictions. Sim 3 was feasible most of the time, but Figure 4-9 shows that there were a number of instances in which error box violations occurred. It should be noted that Sim 3 was the only one of the simulations that used the original MELCO mission control specifications and succeeded in remaining stable. All of the other simulations used modifications that enlarged the error boxes at some or all times. Sim 6 used a 5×10×5 meter error box for the 50 m configuration and did not require any error box violations. At the 500 m level, the addition of the invariance constraint causes an almost 2 order of magnitude fuel consumption increase. It appears that this is because almost no fuel is used without invariance as constraint, but the resulting trajectories are naturally invariant (see Figs. 4-4, 4-7, 4-13, and 4-16) but with slight semimajor axis mismatches which cause the ellipses to travel inside the error box. The trajectories
with invariance (see Figs. 4-19 and 4-19) take on similar shapes but expend fuel to cancel any real or perceived drift introduced through navigation error each with each successive optimization.

Overall, it appears that the most consistent combination of successful constraint satisfaction and low fuel came from the simulations using the invariance constraints (Sims 3 and 6). These simulations used the observation thrusting restriction and the specified error box size for Sim 3 and a slightly larger error box for the 50 m configuration in Sim 6. Thus, the 500 m and 5 km configurations were the same in Sims 3 and 6 and, as would be expected, they have nearly identical fuel use. The only difference is at the 50 m level, where a 5 m increase in the in-track error box size decreases the fuel use by more than a factor of 4.

The fuel numbers for the maneuvers between formation types are included for completeness, but cannot be used to draw conclusions because of the stochastic nature of the simulations. Many additional simulations would need to be run and averages examined. This fact does not reduce the validity of the conclusions regarding the fuel use for station-keeping, because in those situations the fuel use tends to reach a steady state, as is evidenced by the linear (fixed slope) rates of fuel use in the $\Delta V$ plots in this section.

### 4.5 Summary

Multiple high fidelity simulations of the MELCO formation flying mission were performed successfully. These simulations demonstrated, with the most rigorous tools available, that the MELCO mission can be feasibly controlled and made to meet all of its performance constraints using the dynamics and model predictive control formulation introduced in Chapter 2. The simulations indicated that the fuel cost of reserving passive observation time can be significant. Also, it appears that the fewest constraint violations and most consistently low fuel usage occur when a terminal invariance constraint is added to the control formulation. It was demonstrated that small increases in error box size can result in large fuel savings.
(a) Fuel Use  
(b) 3D LVLH Trajectory  
(c) LVLH Trajectory: in-track / radial  
(d) LVLH Trajectory: in-track / cross-track

**Fig. 4-3:** Simulation #1: Stationkeeping in a passive aperture formation (50 m)
Fig. 4-4: Simulation #1: Stationkeeping in a passive aperture formation (500 m)
Fig. 4-5: Simulation #1: Stationkeeping in a passive aperture formation (5000 m)
Fig. 4-6: Simulation #2: Stationkeeping in a passive aperture formation (50 m)
Fig. 4-7: Simulation #2: Stationkeeping in a passive aperture formation (500 m)
Fig. 4-8: Simulation #2: Stationkeeping in a passive aperture formation (5000 m)
Fig. 4-9: Simulation #3: Stationkeeping in a passive aperture formation (50 m)
Fig. 4-10: Simulation #3: Stationkeeping in a passive aperture formation (500 m)
Fig. 4-11: Simulation #3: Stationkeeping in a passive aperture formation (5000 m)
Fig. 4-12: Simulation #4: Stationkeeping in a passive aperture formation (50 m)
Fig. 4-13: Simulation #4: Stationkeeping in a passive aperture formation (500 m)
Fig. 4-14: Simulation #4: Stationkeeping in a passive aperture formation (5000 m)
Fig. 4-15: Simulation #5: Stationkeeping in a passive aperture formation (50 m)
Fig. 4-16: Simulation #5: Stationkeeping in a passive aperture formation (500 m)
Fig. 4-17: Simulation #5: Stationkeeping in a passive aperture formation (5000 m)
Fig. 4-18: Simulation #6: Stationkeeping in a passive aperture formation (50 m)
Fig. 4-19: Simulation #6: Stationkeeping in a passive aperture formation (500 m)
Fig. 4-20: Simulation #6: Stationkeeping in a passive aperture formation (5000 m)
Chapter 5

Safe Trajectories for Autonomous Rendezvous of Spacecraft

Autonomous spacecraft rendezvous is an enabling technology for many future space missions, but anomalies in recent flight experiments suggest that safety considerations will play an important role in the success of future missions. This chapter presents a method for online generation of safe, fuel-optimized rendezvous trajectories that guarantee collision avoidance for a large class of anomalous system behaviors. Next, the chapter examines the cost of imposing safety as a problem constraint and of additional constraints that guarantee infinite horizon passive collision avoidance while enabling future docking retries. Tradeoffs between passive and active approaches to safety are examined. A convex formulation of the collision avoidance algorithm is introduced and shown to provide much faster solutions with only a small additional fuel expense. Numerous examples using both rotating and non-rotating targets are presented to demonstrate the overall benefits of incorporating these safety constraints when compared to nominal trajectory design techniques.

5.1 Introduction

This chapter introduces a method for generating fuel-optimized rendezvous trajectories online that are safe with respect to a large class of possible spacecraft anomalies.
A safe trajectory is defined as an approach path that guarantees collision avoidance in the presence of a class of anomalous system behaviors. Similarly, a passive safe trajectory guarantees collision avoidance with no thrusting required for safety and an active safe trajectory requires that inputs be applied to keep the system safe in the event of a failure. Note that this definition of safety is more restrictive than guaranteeing nominal collision avoidance because it guarantees that no collisions will occur for a range of faults. In particular, for a passive safe trajectory, safety is guaranteed even if the chaser spacecraft cannot use thrusters, computers, or communications equipment. The rationale behind choosing a passive abort strategy is threefold: (a) passive abort can protect against a large set of possible system failures simultaneously; (b) an abort trajectory that does not require fuel use guarantees that remaining fuel will not be expended rapidly to increase spacecraft separation distance, thereby increasing the likelihood that future docking attempts can occur; and (c) passive abort guarantees thrusting will not be used in close proximity to the target during an anomaly, thereby eliminating the danger of plume impingement during an automatic safe-mode maneuver. Active safety is less restrictive than passive safety, but it requires that the types of any failures be identified in real-time and that some components of the control system remain operational so that a sequence of control inputs can be applied.

The following sections review a method for generating fuel-optimized trajectories from linearized relative dynamics and develop a novel approach for guaranteeing those trajectories will be safe. Several examples of safe trajectories generated for docking with both rotating and non-rotating target spacecraft establish that adding safety constraints does not result in significantly increased fuel use. Next, we examine additional constraints to guarantee desirable infinite horizon passive collision avoidance and ease of future docking attempts. To address online implementation considerations, a convex formulation of the safety problem is introduced that trades some performance for large computation reductions. An active form of safety is then considered as a means of reducing fuel costs while still remaining safe for a large set of possible failure modes.
5.2 Online Trajectory Optimization for Autonomous Rendezvous and Docking

A trajectory generated through online optimization can be designed by choosing the system inputs that produce that trajectory. For a linear system, methods for incorporating and propagating the effects of inputs are well-known. The trajectory optimization formulation in this section is presented in the context of linear time-invariant dynamics, but there is no inherent restriction in the formulation preventing the use of time-varying dynamics [84]. Given a chaser satellite whose state is $x_k$ at time $k$, the linearized dynamics of the system can be written as

$$x_{k+1} = A_d x_k + B_d u_k$$

(5.1)

where $A_d$ is the state transition matrix for a single time step, $B_d$ is the discrete input matrix for a single time step, and $u_k$ is the input vector at step $k$. Typically, in a rendezvous situation, spacecraft would be in sufficiently close proximity to enable the use of the Hill-Clohessy-Wiltshire (HCW) equations [75], but GVE-based approaches [74] can be used for more widely separated situations. Examples in this chapter will use the HCW equations and hence the state $x$ is defined as

$$x = \begin{bmatrix} x & y & z & v_x & v_y & v_z \end{bmatrix}^T$$

(5.2)

where $x$, $y$, $z$, $v_x$, $v_y$, and $v_z$ are the positions and velocities of a chaser satellite in the radial, in-track, and cross-track axes, respectively, of an LVLH frame positioned on the center of gravity of a passive target vehicle. The input is defined as

$$u = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$$

(5.3)

where $u_x$, $u_y$, and $u_z$ are the inputs of the chaser vehicle in the axes indicated by the subscripts in the LVLH frame.
Given an initial state $x_0$, the state at any future step $k$ is [102]

$$x_k = A_d^k x_0 + \begin{bmatrix} A_d^{k-1} B_d & A_d^{k-2} B_d & \ldots & A_d B_d & B_d \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{k-1} \end{bmatrix}$$

$$= A_d^k x_0 + \Gamma_k \begin{bmatrix} u_0 \\ \vdots \\ u_{k-1} \end{bmatrix}$$

(5.4)

(5.5)

where $\Gamma_k$ is the discrete convolution matrix. Since the effects of the control on the states are readily expressed as linear combinations of the inputs, a linear optimization can be formed that optimizes the constrained control commands and constrains the states of the system. The cost function for this optimization will exclusively penalize fuel use. In an actual maneuver implementation, it may be preferable to optimize both the fuel use and the maneuver duration (see Ref. [77]), however in this chapter only fuel use will be considered to simplify presentation and cost comparisons. The cost of the optimization $J$ is given by

$$J = \sum_{i=0}^{N-1} \|u_i\|_1$$

(5.6)

where the 1-norm cost is used to capture the expenditure of fuel used, which is proportional to acceleration and $\Delta V$, from axial thrusters. The optimal cost is then given by

$$J^* = \min_{u_0, \ldots, u_{N-1}} \sum_{i=0}^{N-1} \|u_i\|_1$$

(5.7)

At each step $k$, it is possible to constrain the state at that time to lie inside a convex region

$$A_k x_k \leq b_k$$

(5.8)

where $A_k$ is a matrix and $b_k$ is a vector that together capture a set of linear constraints on the state. Note that the costs and constraints in Eqs. 5.6 and 5.8 show an example
linear implementation of a trajectory optimization, but in general the same concepts that will be presented hold for nonlinear costs and constraints as well. Alternately, the state $x_k$ could be constrained to lie outside a region through the use of binary variables [61]

$$A_k x_k \leq b_k + My_k$$

$$\|y_k\|_1 \leq m - 1$$

where $y_k$ is a vector whose elements are constrained to be 0 or 1, and $M$ is a large number on the scale of values taken by elements of $x$. This “Big M” method of collision avoidance works by allowing, at most, all but one of the collision avoidance constraints to be relaxed. A constraint is relaxed when the binary variable associated with it is set to 1, thereby making the right-hand side of the inequality very large and guaranteeing constraint satisfaction. Since at least one constraint is always guaranteed to not be relaxed, collision avoidance is assured (e.g., knowing that one is outside of one side of a box is sufficient information to guarantee that one is not in the box).

The inputs at each time step can also be directly constrained using

$$u_{\min_k} \leq u_k \leq u_{\max_k}$$

where $u_{\min_k}$ and $u_{\max_k}$ are vector bounds on the values of $u_k$. Typically, the minimum thrust at all times would be $-u_{\max_k}$. A detailed description of the full matrix forms used in linear trajectory optimizations for space vehicles can be found in Refs. [84] and [61].

This section has reviewed an approach for creating fuel-minimizing trajectories that satisfy time-varying position, velocity, and thrusting constraints. Applications of these constraint types can insure that a spacecraft remains inside a line-of-site cone, and arrives at a docking port position at a particular time with a particular speed range. In addition, the control authority available over the course of the trajectory can be varied according to desired pattern.
5.3 Safety Formulation

The trajectories generated by the constraints in Section 5.2 will satisfy docking requirements and use minimal fuel to arrive at a rendezvous location. However, as is typical of optimal paths, the trajectories will approach constraint boundaries and generally be sensitive to uncertain behavior. Refs. [73] and [77] describe computationally feasible methods of generating trajectories online that are robust to process and sensing noise expected under nominal operating conditions. That type of robustness to uncertainty is distinct from the definition of safety for off-nominal conditions considered herein. This section presents an approach for generating trajectories that are safe with respect to a class of system failures. While it would be desirable to avoid collisions and successfully complete docking in the presence of any system failure, it is unlikely that such a scenario is possible. Instead, a large subset of all possible failures is used, including guidance system shutdowns, which encompasses thruster failures, computer anomalies, and loss of sensing. The response to these types of failures would be a guidance system shutdown in which the chaser vehicle would go into a safe mode with all its thrusters turned off. Safety to this class of failures is called *passive abort safety*, because any rendezvous can be aborted using no thrusting. Passive abort safety guarantees collision avoidance for any failure that can be identified and responded to by disabling thrusters before the spacecraft trajectory is affected. This type of safety does not include failures in which a thruster fails on (see Section 5.8).

A consequence of passive abort is that if thrusters are disabled at any step $T$, counted from the start of the plan, during the trajectory implementation, then the thrusters will remain failed until the last step $N$ of the plan. Clearly choosing $N - T$ to be small (*i.e.*, only constraining steps toward the end of the horizon to be safe) puts fewer constraints on the trajectory optimization than a large value, but it assumes that more of the plan will be successfully implemented. Conversely, a large $N - T$ is a more conservative approach to safety and more tightly constrains the optimization. The choice of which steps in the plan that are constrained to be safe depends on the
specific characteristics of the spacecraft and the mission requirements. The objective of this chapter is to present a systematic way of embedding the safety goals into the path planning problem so that the designer can evaluate the trade-offs associated with choosing $T$.

Note that enabling passive safety abort at the end of a trajectory can eliminate potential plume impingement conflicts that may result from last-minute safe-mode maneuvers. A further benefit of guaranteeing safety at the end of the trajectory is that it gives controllers an immediate safe exit from the docking procedure during the period when the spacecraft are in the closest proximity and there is the least time available to plan emergency maneuvers.

The discrete convolution approach used in Eq. 5.4 can be used to predict the state of the chaser at step $k$ in the planning horizon in the event of a failure at time $T$, by considering all inputs after and including the input at time $T$ to be zero

$$x_{FT_k} = A_d^k x_0 +$$

$$\left[ A_d^{k-1} B_d \ A_d^{k-2} B_d \ \ldots \ A_d^{T-2} B_d \ 0 \times A_d^{T-1} B_d \ \ldots \ 0 \times A_d B_d \ 0 \times B_d \right] \begin{bmatrix} u_0 \\ \vdots \\ u_{k-1} \end{bmatrix}$$

(5.12)

where $x_{FT_k}$ is the state of the chaser spacecraft at some step $k < N$ in the planning horizon after a failure occurred at step $T$. The value of a failure state after the planning horizon is found through open-loop propagation of the state at time $N$

$$x_{FT_k} = A_d^{k-N} x_{FT_N} \text{ for } k \geq N$$

(5.13)

Passive collision avoidance is achieved by adding constraints on the failure states of the spacecraft. Define the set of position states occupied by the target as $T_k$, which can describe any polytopic region of position states, convex or otherwise. The safety horizon is the period of time after a failure during which both spacecraft are guaranteed not to collide. The safety horizon lasts $S$ steps after the end of the nominal
trajectory and is guaranteed by introducing the set of constraints

\[ x_{FT_k} \notin T_k \quad \forall \quad k \in \{T + 1 \ldots N + S\} \] (5.14)

The constraints in Eq. 5.14 are then imposed for \( T \in F \) where \( F \) is the set of every potential failure time at which the system must guarantee collision avoidance for guidance shutdowns. The parameters to be chosen in this safety formulation are \( F \) and \( S \). This choice of parameters is highly dependent on the requirements of a particular space mission. The advantage of choosing to be safe for a large number of steps and for a long safety horizon is improved likelihood of preventing a catastrophic failure scenario in which the chaser and target collide. However, imposing many safety constraints greatly reduces the number of potential solution trajectories and as a result, likely reduces fuel efficiency. The tradeoff between safety and fuel efficiency is discussed in the scenarios in Sections 5.4.1 and 5.4.2.

### 5.4 Scenarios

The rendezvous and docking scenario to be examined in this chapter involves a target spacecraft being docked with and a chaser spacecraft maneuvering to achieve that docking. Figure 5-1 shows a target spacecraft that lies at the center of an local frame. A line-of-sight (LOS) cone protrudes from the target spacecraft and it is required that rendezvous remain within this line-of-sight cone for vision-based sensing. At the interface between the LOS cone and the target is a docking port (rectangular platform). In the rotating case (Figure 5-2) the axis of rotation is the long axis of the spacecraft and the rotation rate is orbital. The choice of rotation axis and rate was arbitrary and only enter the optimization through their effect on the time-varying constraints imposed for LOS requirements, docking, and safety. The LOS requirements are

\[ A_{LOS_k} x_k \leq b_{LOS_k} \forall k = 1 \ldots N \] (5.15)
where $A_{\text{LOS}_k}$ and $b_{\text{LOS}_k}$ describe the states within the LOS cone at a step $k$ in the planning horizon. The terminal constraint is

$$A_{\text{Term}_N} x_N \leq b_{\text{Term}_N}$$  \hspace{1cm} (5.16)$$

where $A_{\text{Term}_k}$ and $b_{\text{Term}_k}$ describe the states the spacecraft must occupy at the end of the planning horizon to achieve safe docking. These constraints can be both on position (e.g., enter a region within reach of a grappling arm) and on velocity (e.g., dock within a velocity range that produces acceptable stress on the docking port). In addition, time-varying bounds are introduced on the maximum thrusting levels in order to ensure large thrusts are not planned for the period immediately before docking.

The safety constraints in Eq. 5.14 are imposed for the last quarter of the planning horizon. In the examples, an orbit with frequency $n = 0.001 \text{ rad/s}$ is used and is discretized into 20 steps and the set of inputs that can fail is $T \in \{14 \ldots 19\}$. The planning horizon is a full orbit. The chaser spacecraft, modeled after the mission in Ref. [81], has a mass of 45 kg and a maximum acceleration of $10^{-3} \text{ m/s}^2$ during the first 17 steps of the plan and $10^{-5} \text{ m/s}^2$ for the last 3 steps to prevent trajectory solutions with large terminal thrusts. In addition, the docking constraint specifies that the velocity of the spacecraft at the time of docking be less than 1 mm/s. In summary, the safety algorithm used in this section is

$$\min_{u_0, \ldots, u_{N-1}} \text{Eq. (5.6)}$$  \hspace{1cm} (5.17)$$

s.t. \hspace{0.5cm} \text{Eq. (5.11)} \hspace{0.5cm} \forall \hspace{0.1cm} k \in \{0, \ldots, N - 1\},$$

$$\text{Eq. (5.15)},$$

$$\text{Eq. (5.16)},$$

$$\text{Eq. (5.14)} \hspace{0.5cm} \forall \hspace{0.1cm} T \in \mathcal{F}$$

In these examples, the safety horizon is a full orbit. Any of the design parameters in the safety implementation can be easily adjusted and in practice one would likely conduct a simulation study or analysis [121] to find the best combination for
minimizing fuel use and guaranteeing feasible solutions.

5.4.1 Case 1: Stationary Target Satellite

An optimized trajectory with no safety constraints for the stationary target case is shown in Figure 5-3. The initial trajectory to the docking port roughly corresponds to a two impulse V-bar (in-track) [67] approach. The nominal trajectory is marked with • and the failure trajectories with ×. Failure trajectories, the paths followed by the spacecraft in the event of a guidance shutdown, are shown for last five possible inputs. Several of the failure trajectories overlap, a condition which corresponds to the nominal input at a step already being set to zero thrust. All of the failure trajectories clearly impact the target spacecraft. Figure 5-4 shows the same rendezvous situation for trajectories generated with safety constraints. In this case, none of the failure trajectories impact the target spacecraft. An apparent violation of the constraint boundary is visible in the lower left-hand corner of the figure. This type of corner violation is possible in the MILP framework, because only the discrete points are constrained, not the trajectory between discrete points. Reference [82] gives an approximation for the amount a constraint should be enlarged to ensure that any violations will not intersect the actual constrained region. As in the case without safety, several of the failure trajectories overlap. The fuel costs (measured in $\Delta V$) of the trajectory with no safety guarantees and the trajectory with safety are 1.29 mm/s and 1.91 mm/s, respectively. Hence, in this case, imposing safety results in a 48% increase in fuel use. To put these numbers in context, an optimized approach constrained to follow a V-bar trajectory (strictly in-track) would use 37.7 mm/s of fuel.

An approximate method for over-bounding the optimized numbers would be to consider an approach based on introducing in-track drift and arriving at the docking port after a full planning horizon, with no other constraints. In this case, the planning horizon is a full orbit, with no initial radial offset and no initial velocity, an initial thrust in the in-track direction will introduce a secular drift into the relative orbit.
Fig. 5-1: Target spacecraft and docking configuration

Fig. 5-2: Radial/in-track view of rotating target spacecraft and docking configuration
Fig. 5-3: Nominal trajectory planning with no safety: constraint violations occur for trajectory failures. The nominal trajectory is marked with • and the failure trajectories with ×. The failure trajectories all result in collisions with the target spacecraft.

Fig. 5-4: Trajectory planning with safety: failed trajectories deviate around the target spacecraft, preventing collision. The nominal trajectory is marked with • and the failure trajectories with ×.
Over the course of an orbit this secular drift causes the in-track position to shift by

\[ \Delta y = \Delta v_y 6\pi /n \]  

(5.18)

according to the in-track solution for Hill’s equations [106]. For the example in this section, the amount of fuel required for this maneuver would be 0.93 mm/s. An approximate upper bound on the fuel number could be obtained by forming a constrained problem similar to the LP, but forced to follow a strict in-track (V-bar) trajectory. If the 2-norm of fuel use is minimized as an approximation instead of the correct 1-norm metric and the thruster inputs are not constrained, this problem can be solved using a pseudo-inverse. By constraining the radial position state at each step \( k \) in the planning horizon to be zero and the final in-track position to lie on the edge of the docking port, the following equality constraints are formed

\[
A_Q \begin{bmatrix} u_0 \\ \vdots \\ u_{k-1} \end{bmatrix} = b_Q
\]  

(5.19)

with

\[
A_Q = \begin{bmatrix} 
H_x \Gamma_1 & 0 & \ldots & 0 & 0 \\
H_x \Gamma_2 & 0 & \ldots & 0 & 0 \\
H_x \Gamma_{N-1} & 0 & & & \\
H_x \Gamma_N & & & & \\
H_y \Gamma_N & & & & 
\end{bmatrix}, \quad 
b_Q = \begin{bmatrix} 
-H_x A_1^2 x_0 \\
-H_x A_2^2 x_0 \\
\vdots \\
-H_x A_N^2 x_0 \\
-H_y A_N^2 x_0 + y_{des} 
\end{bmatrix}
\]  

(5.20)

where \( H_x \) is a row vector that extracts the scalar radial component, \( H_y \) is a row vector that extracts the scalar in-track component, and \( y_{des} \) is the desired in-track component at step \( N \). This form has \( N + 1 \) constraints and \( 3N \) input variables to choose. The trajectory that minimizes the 2-norm of the input vectors and meets
those simple constraints is given by the pseudo-inverse solution \([83]\) of

\[
\begin{bmatrix}
I_{3N} & A_Q^T \\
A_Q & 0_{N+1}
\end{bmatrix}
\begin{bmatrix}
u_0 \\
\vdots \\
u_{k-1} \\
z
\end{bmatrix}
= \begin{bmatrix}
0_{3N,1} \\
0_p \
b_Q
\end{bmatrix}
\tag{5.21}
\]

where \(z\) is a vector of \(N + 1\) Lagrange multipliers, \(I_p\) is a \(p \times p\) identity matrix, \(0_q\) is a \(q \times q\) matrix of zeros, and \(0_p,1\) is a \(p \times 1\) vector of zeros. The fuel cost of the trajectory found using this method is 39.1 mm/s, which is very close to the optimized cost of following a strict V-bar trajectory.

### 5.4.2 Case 2: Docking Port Perpendicular to Spin Axis

The rotating docking port case uses an identical formulation to the stationary case, however, the constraint regions are time-varying. In particular, in the stationary case, \(A_1 = A_2 = A_k \forall k = 1 \ldots N\), but to formulate the rotating problem, the \(A_k\) and \(b_k\) matrices must be formed for each step of the planning horizon based on the rotation rate and, for more general cases, the motion of the target, the docking port, and the line-of-sight cone. One simple way to generate these constraints is to represent each side of an avoidance region as a plane that is specified by a sample of its constituent points. These points will remain in a plane through any reorientation of the original constraint. Thus, the rotated constraint side can be found by applying rotation matrices \([80]\) to the points and then forming the equation of a new plane, which can then be used as an inequality constraint. The translation and rotation motion of the target spacecraft should be well-characterized through observation or cooperation before starting a rendezvous maneuver, thereby allowing the prediction of its future trajectory to be used for forming constraints. All trajectory propagations of the target spacecraft used to create constraints are formed before the rendezvous maneuver is optimized. As a result, the propagation can be carried out using any method appropriate for the specific online implementation (e.g., simple linear propagation,
high accuracy numerical integration). Robustness to uncertainty in target motion can be accommodated by guaranteeing that any optimized trajectory is valid for a range of representative target initial conditions [84].

For the rotation case examined in this section, the optimized trajectory no longer matches a two-impulse V-bar approach, but is instead forced to thrust regularly to stay within the rotating LOS cone. Figure 5-5 shows this optimized trajectory with no safety constraints. As in the stationary case, in the absence of safety constraints, the nominal trajectory collides with the target in the event of guidance shutdowns. An alternate form with safety constraints prevents collisions for failures occurring in the last quarter of the trajectory (Figure 5-6). Note that in Figure 5-6, the safe trajectory appears to pass through the target, but in actuality it avoids collision because of the rotational motion of the target. The fuel costs without safety constraints and with safety constraints are 55.3 mm/s and 56.4 mm/s, respectively. In this case, the fuel cost of imposing safety as a constraint is only a 2% increase over the nominal cost. As in the non-rotating case, the increase in fuel needed to include the safety constraints is minimal and the advantage is guaranteed collision avoidance for passive abort in the last quarter of the nominal path. Another example (safe trajectory in Figure 5-7) using a target rotating at 3/2 orbital rate required 68.7 mm/s of fuel with no safety constraint and 69.1 mm/s of fuel with a safety constraint.

5.5 Probability of Collision

To judge the effectiveness of the safety algorithm introduced in Section 5.3, define a probability of collision metric, \( P_{col} \), which is the probability of a failure at any time step during a maneuver resulting in a collision between the target and chaser spacecraft. The probability of collision is given by

\[
P_{col} = \sum_{i=1}^{N} P(\text{failure at } i \mid \text{no failure before } i) \cdot P(\text{collision occurs } \mid \text{failure at } i)
\]

(5.22)
Fig. 5-5: Nominal trajectory planning with no safety in the rotating case: constraint violations occur for trajectory failures.

Fig. 5-6: Trajectory planning with safety in the rotating case: failed trajectories deviate around the target spacecraft, preventing collision.
Fig. 5-7: Trajectory planning with safety in the fast rotating (3/2 orbital speed) case. The maneuver $\Delta V$ cost with safety is 69.1 mm/s.

where the probability $P(\text{collision occurs} \mid \text{failure at } i)$ is either 1 or 0 and is evaluated by examining the trajectory followed if thrusters are disabled at step $i$ and checking for future collisions. Assuming that the probability of a failure at any step in the trajectory is $f$, then

$$P(\text{failure at } i \mid \text{no failure before } i) = (1 - f)^{i-1} f$$  \hspace{1cm} (5.23)$$

Using the metric $P_{col}$, the effectiveness of the safety approach was investigated by creating a series of safe trajectories starting from different initial conditions near the target. The initial condition positions were chosen to create a range of nearby starting points. The velocity vector for each position was chosen according to the conditions in
Ref. [68] to create a safety circle. This creates a situation where each rendezvous trajectory begins from a safe, invariant orbit within range of a final approach rendezvous trajectory.

Figures 5-8 and 5-9 show the values of $P_{col}$ for full-orbit optimized final approach trajectories, discretized into $N = 20$ steps. The trajectories are generated using $\mathcal{F} = \emptyset$ (no safety constraints) and $\mathcal{F} = \{9, \ldots, 19\}$ (guaranteed safe for the last 10 steps of the trajectory), respectively, where $f = 0.001$. These plots show that without safety, the probably of collision for a given rendezvous trajectory tends to fall between 0.005 and 0.015. However, the addition of safety for half of the trajectory brings the collision probability for most of the trajectories below 0.001. The same optimizations were performed and analyzed for a range of other $\mathcal{F}$ ranges and the results are summarized in Figure 5-10. The dashed line indicates an overbound, $\bar{P}_{col}$, for the maximum possible probability of collision, which is the case where every failure during the course of the trajectory when safety is not guaranteed (i.e., steps not in $\mathcal{F}$) would result in a collision, which is given by

$$\sum_{\{0, \ldots, N-1\}\setminus\mathcal{F}} P(\text{failure at } i \mid \text{no failure before } i)$$

(5.24)

The line marked with ♦ shows the largest probability encountered in the optimized trajectories for all initial conditions considered. This is equivalent to finding the maximum height ($z$ value) in a plot of the type in Figure 5-9 for each different set $\mathcal{F}$ used to create Figure 5-10. The minimum (line marked by •) shows that in each case, there were some initial conditions that did not result in collision, regardless of the steps in $\mathcal{F}$. In those cases, the fuel-optimal rendezvous trajectory is safe. The average $P_{col}$ (solid line), equivalent to averaging the probability heights over an area of the type in Figure 5-9, followed a similar trend to the largest $P_{col}$, but was significantly lower. This indicates that although some initial conditions are particularly prone to collision, on average the collision probabilities are significantly improved by safety and in no case has the addition of safety made collisions more likely than in the fuel-optimal case ($\mathcal{F} = \emptyset$). Furthermore, for this particular case, the trends indicate that
guaranteeing more than the last five steps safe does not significantly decrease the probability of a collision. This conclusion would be valuable from a mission planning perspective, because each additional plan step that is guaranteed safe represents a tradeoff in which computation time and nominal fuel use potentially increase.

Eq. 5.24 indicates that the overbound $\bar{P}_{\text{coll}}$ decreases with increasing length of the safe region (i.e., fewer steps in $\mathcal{F}$). For the purposes of worst-case safety guarantees, the overbound could be used as an analytic rule-of-thumb for mission design studies.

5.6 Invariant Formulation

The safety formulation introduced in Section 5.3 only guarantees passive collision avoidance until the end of the safety horizon. In previous examples, the safety horizon has been fixed at one orbit. Figure 5-11 shows a stationary-target case where a collision would occur soon after the end of a one orbit safety horizon. If the safety horizon is extended to multiple orbits, the resulting failure trajectories will tend to either drift away from target spacecraft or create invariant orbits that neither drift toward nor away the target spacecraft. Drifting away from the target orbit is preferable to collision, however it means that fuel will need to be expended to bring the chaser near the target for any future docking attempts. Furthermore, the longer controllers wait to cancel the drift, the farther apart the two spacecraft will become, thereby creating an additional timing consideration during an anomalous event. It is preferable for the chaser to drift into an invariant orbit that is near the target, but can never, under the assumptions of Keplerian dynamics, collide. The preference for invariant failure orbits can be captured by constraining a state in the failure trajectory at some step $k$ to be the same one full orbit after $k$ using a linear state transition matrix to propagate the state forward. This constraint is written

$$x_{FT_k} = A_d^N x_{FT_k} \text{ for } k \geq T \quad (5.25)$$

123
Fig. 5-8: Probability of a collision occurring for a range of initial conditions with $\mathcal{F} = \emptyset$ (No safety guarantees).

Fig. 5-9: Probability of a collision occurring for a range of initial conditions with $\mathcal{F} = \{9, \ldots, 19\}$ (latter half of trajectory guaranteed safe)
where $N_o$ is the number of steps in an orbit. By imposing this constraint for all possible failures times in $\mathcal{F}$, all failure orbits are guaranteed to be invariant with respect to the target. The safety algorithm in Eq. 5.17 is only altered by the addition of the new constraints in Eq. 5.25.

Figure 5-12 shows the same rendezvous problem from Figure 5-11, but with additional invariance constraints on failures occurring in the last quarter of the rendezvous trajectory. Imposing invariance constraints yields circular trajectories relative to the target, which are traversed once per orbit with no fuel expenditure. Some of the six failure trajectories shown in Figure 5-12 overlap each other where the optimized trajectory did not require fuel inputs. The safe trajectory with no invariance constraint used 3.4 mm/s of fuel and the safe trajectory with invariance used 7.1 mm/s of fuel. In this case, the invariance constraints have roughly doubled fuel requirements. However, this tradeoff may be beneficial when there is a danger of collision after the safety horizon or when the fuel requirements for canceling drift after a failure.
are taken into account. For this case, the lower-bound estimated fuel numbers from Eq. 5.18 predict 1.4 mm/s of fuel required and the strict V-bar approach solution from Eq. 5.21 predicts 61.4 mm/s of fuel use, which is close to the optimized V-bar solution which requires 59.2 mm/s. The cost of using an optimized rendezvous trajectory with no safety constraints is 3.1 mm/s, making safety only 1.3% more expensive. The cost of using a strict V-bar approach is roughly 18 times more expensive than the optimized safe approach and 8.6 times more expensive than the optimized safe invariant approach. Thus, when compared to the cost of using a standard non-optimized maneuver, the fuel premium for using safety with invariance is small.

The examples using invariance constraints presented in this section used Hill’s dynamics [75], meaning that the particular type of invariance achieved is effectively equivalent to constraining the relative semimajor axes of the target and chaser spacecraft to be zero. Alternate equations of motion exist which model additional orbital perturbations. For example, the dynamics presented in Chapter 2 could be used for rendezvous and docking trajectory generation to create invariant orbits that account for the effects of differential $J_2$. Likewise, the effects of differential drag can also be taken into account using the method in Ref. [84].

5.7 Convex Safety Formulation

The safety constraints introduced in Section 5.3 guaranteed that the chaser states would not collide with the target in the event of a failure. The collision avoidance in those constraints is accomplished using binary variables to capture the nonconvexity of the problem. The problem with binaries was formulated as a Mixed-Integer Linear Program (MILP) and posed to a commercial solver. Solving a MILP can be a computationally intensive task and the time required to solve tends to grow very quickly with the number of discrete variables in the problem [111]. The trajectory shown in Figure 5-4 required 8.92 seconds to solve on a 3 GHz computer. That problem had a 20 step safety horizon, an avoidance region with 8 sides, and six inputs that were safe in the event of guidance shutdown. Each avoidance region side requires a binary
Fig. 5-11: Case where end of safety horizon is followed by a collision.

Fig. 5-12: Use of invariance constraints guarantees infinite horizon passive collision avoidance and prevents failure trajectories from drifting away from the target.
variable at each step of the safety horizon and those constraints are included 6 times, each propagating forward from a different failed thruster step. Thus, implementing collision avoidance over the safety horizon for that simple example required 960 binary variables. Solving the same trajectory for a two orbit safety horizon required 24.6 seconds. Using the same horizon duration with a finer discretization step would further increase the required computation time. It is likely that online implementations would need to solve with limited computer resources and that a nonconvex implementation may be impractical for implementations requiring short discretization steps.

An alternative to the nonconvex formulation is to use a more restrictive form of collision avoidance that is convex. Instead of requiring the chaser to remain outside an avoidance region, the failure trajectories are instead constrained to remain inside a region that is known to not contain the target. This is similar to the type of convex passively safe trajectory examined for rotating satellite capture in Ref. [70], however the approach in this chapter explicitly minimizes fuel use. Figure 5-13 shows an example of the optimized trajectory for the rendezvous problem in Figure 5-4, but instead solved as a linear program (LP) using the convex safety constraints

\[ H_y x_{FT_k} \geq y_{\text{min}} \quad \forall \ k \in \{ T + 1 \ldots N + S \} \]  

(5.26)

where \( y_{\text{min}} \) is the maximum in-track position of the target spacecraft. The nonconvex case results in failure trajectories that are not permitted in the convex case and as a result, there is a fuel penalty for imposing convexity. Using the convex formulation in Eq. 5.26 does not require altering the basic safety algorithm in Eq. 5.17. Instead, only the target geometry, given by the set \( T \) in Eq. 5.14, needs to altered to ensure that its complement is convex. The nonconvex case requires 1.5 mm/s of \( \Delta V \) and the convex case requires 3.7 mm/s. The more restrictive area in which failure trajectories can lie caused the required fuel to increase by more than a factor of 2. However, the amount of time required to compute the convex trajectory was only 0.06 seconds, a decrease from the nonconvex case by a factor of 150. In cases where it is impractical
Fig. 5-13: Collision avoidance for failure trajectories using convex constraints indicated by arrows.

to dedicate significant computational resources to planning, it may be desirable to trade the fuel optimality of the more general MILP formulation for the speed of the LP formulation. In addition, the convex solution is often similar to the invariant given in Eq. 5.25. For the example in Figure 5-13, the trajectory cost when invariance and convexity are imposed is 4.2 mm/s, which is the same as the cost of using invariance in the fully nonconvex problem.

5.8 Active Safety

An alternative to passive safety is active safety, in which a set of thruster inputs is applied to ensure rendezvous safety. The active response is a set of input sequences that is used instead of passive safety. The safe input sequence can be designed a priori
(e.g., thrust in-track, thrust radially) and chosen in real-time or optimized at the time the nominal rendezvous maneuver is optimized. In either case, the safe inputs are known at all times during the maneuver and no additional optimization is required in the event of a failure. The advantages of active safety over passive safety are significant: by allowing thrusting in the event of a failure, a significantly larger portion of the nominal trajectory can be guaranteed safe and the fuel costs of guaranteeing the safety of the nominal trajectory are reduced. Passively safe trajectories can be considered a subset of active safe trajectories in which the active input sequence has no thrusting. The primary limitation of active safety is that it provides safety guarantees for a smaller set of possible system malfunctions than passive safety. In the case of passive safety, any anomaly in which the thrusters can be disabled can be made safe. Safety guarantees resulting from an active safety trajectory require that some thrusters continue to work properly and in the correct directions in the event of a failure. An extension at the end of this section will show how active safety can be modified to provide safety guarantees for single thruster failures.

To create an active safety constraint, the optimization from Section 5.3 is altered to allow the possibility of using a safe input sequence by introducing an additional discrete convolution matrix. Rewriting Eq. 5.12 for a predetermined safe input sequence

\[
x_{FT_k} = A_d^k x_0 + \left[ A_d^{k-T} B_d \ldots A_d B_d B_d \right] \begin{bmatrix} v_0 \\ \vdots \\ v_{k-T-1} \end{bmatrix}
\]

\[
+ \left[ A_d^{k-1} B_d \ A_d^{k-2} B_d \ldots A_d^{T-2} B_d \ 0 \times A_d^{T-1} B_d \ldots \ 0 \times A_d B_d \ 0 \times B_d \right] \begin{bmatrix} u_0 \\ \vdots \\ u_{k-1} \end{bmatrix}
\]

where \( T < k < N, k - T < N_s, N \) is the number of steps in the nominal plan, and \( N_s \) is the number of steps in the safe input sequence. If \( k \leq T \) then \( x_{FT_k} = x_k \) because
no potential failure could have occurred at that time. Equation 5.27 can be written more generally as

\[
\mathbf{x}_{FT_k} = \begin{cases} 
\Gamma_k \mathbf{S}(T, k) \mathbf{U}_k + A_d^k \mathbf{x}_0 + \Gamma_{k-T} \mathbf{S}(k-T, N_s) \mathbf{V}_k, & T < k \leq N, \\
A_d^{k-N} \Gamma \mathbf{S}(T, N) \mathbf{U}_N + A_d^k \mathbf{x}_0 + \Gamma_{k-T} \mathbf{S}(k-T, N_s) \mathbf{V}_k, & k > N, \ k-T \leq N_s \\
\Gamma_k \mathbf{S}(T, k) \mathbf{U}_k + A_d^k \mathbf{x}_0 + A_d^{k-N} \Gamma \mathbf{V}_N, & k \leq N, \ k-T > N_s \\
A_d^{k-N} \Gamma \mathbf{S}(T, N) \mathbf{U}_N + A_d^k \mathbf{x}_0 + A_d^{k-N} \Gamma \mathbf{V}_N, & k > N, \ k-T > N_s
\end{cases}
\] (5.28)

where \( k \) is the time step that the failure trajectory is propagated forward to, \( \mathbf{S}(q, N_q) = \text{diag}(\mathbf{I}_{3(q)}, \mathbf{0}_{3(N_q-q)}) \), \( \mathbf{I}_n \) is an \( n \times n \) identity matrix, \( \mathbf{0}_n \) is an \( n \times n \) matrix of zeros, the decision variables for the nominal input are the vector \( \mathbf{U}_k^T = [\mathbf{u}_0^T \ldots \mathbf{u}_{k-1}^T] \), and the predetermined safe input sequence \( \mathbf{V}_N^T = [\mathbf{v}_0^T \ldots \mathbf{v}_{k-1}^T] \). The possible ranges in Equation 5.28 correspond to the steps before the nominal plan has ended and before the end of the safe input sequence \( (k \leq N, \ k \leq k-T) \), the times after the nominal plan has ended and before the end of the safe input sequence \( (k > N, \ k \leq k-T) \), times before the nominal plan has ended and after the end of the safe input sequence \( (k \leq N, \ k > k-T) \), and the times after the both the nominal plan and the safe input sequence have ended \( (k > N, \ k > k-T) \). All four cases must be considered in order to allow for safe input sequences that are longer or shorter than the nominal plan length. Active safety can be guaranteed by introducing the set of constraints

\[
\mathbf{x}_{FT_k} \notin \mathcal{T}_k \ \forall \ k \in \{T+1, \ldots, N+S\}
\] (5.29)

The set of constraints in Eq. 5.29 is applied for each step \( T \) at which safety should be guaranteed in the event of a failure.

An alternate approach to active safety where the safe input sequence is optimized online can be implemented by moving the safe input sequence \( \mathbf{V}_N \) into the decision
such that the safe input sequence $\mathbf{V}_{N_s}$ is optimized at the same time as the nominal rendezvous trajectory. Active safety uses the same safety algorithm in Eq. 5.17, but with Eq. 5.14 replaced by Eq. 5.29 using the active failure trajectory given by Eq. 5.28 for a priori known safe input sequences or Eq. 5.30 for safe input sequences optimized online.

The implementation of an active safe trajectory would be similar to that of a safe trajectory. Before entering the trajectory, the spacecraft is assumed to be in a nominal state (i.e., all systems are functioning correctly). If a fault has not yet occurred, the spacecraft follows the nominal trajectory, which is given by $\mathbf{U}_N$. If a fault occurs during a step that has been guaranteed to be safe in the event of that fault, then the spacecraft begins using the safe input sequence. For the duration of the safe input sequence, the chaser and target spacecraft are guaranteed to not collide. If the invariance constraints in Section 5.6 are used, safety can be guaranteed for any time horizon over which the dynamics are valid.
5.8.1 Examples

A stochastic analysis of the type performed in Section 5.5 was conducted using active safety, guaranteeing safety for the last 3/4 of the nominal trajectory ($\mathcal{F} = \{4, \ldots, 19\}$). The maps of collision probability, fuel use, the trajectory computation time are shown in Fig. 5-14. The results indicated that the average collision probability for failures accounted for by active safety was reduced to $1.96 \times 10^{-5}$ from $0.0057$ for the optimal unsafe case and $4 \times 10^{-4}$ for the passive safety case ($\mathcal{F} = \{4, \ldots, 19\}$). The average fuel use across the grid is 27.4 mm/s and the average time required to optimize a trajectory is 1.5 seconds.

The differences between the active safety approaches are demonstrated in Figures 5-15-5-18. Figure 5-15 shows an active safe rendezvous trajectory beginning from a safety circle holding orbit. In this case, the safe input sequence $\mathbf{V}$ has been arbitrarily chosen to be an orbit of constant thrusting at $10^{-6}$ m/s$^2$ in the $-x$ direction of an LVLH frame centered on target. The last three quarters of the rendezvous trajectory have been guaranteed through constraints to be actively safe. In the figure, the nominal rendezvous trajectory (line marked with •) shows the planned rendezvous maneuver which will be followed in no failures occur. Each portion of the trajectory marked with △ shows a possible path followed by the chaser in the event that the safe input sequence is used. Constraints guarantee safe collision avoidance for the entire red portion of the trajectory, however, no safety guarantees exist for the trajectory after the safe input sequence is enacted. The trajectories marked by × show the how the path drifts after the end of each safe trajectory. In several cases, the drifting path would result in a collision at some time in the future. To ensure collision avoidance, Figure ?? shows an active safe trajectory optimized from the same initial conditions, but with the invariance constraint from Eq. 5.25 imposed. In this case, a failure at any step in the last three quarters of the nominal trajectory would result in the chaser spacecraft entering a safe, invariant trajectory near the target spacecraft. Figures 5-17 and 5-18 show the optimized active safe trajectories using the constraints in Eq. 5.30 without and with invariance constraints, respectively.
Fig. 5-14: Active safety demonstrated for a range of initial conditions with $\mathcal{F} = \{4, \ldots, 19\}$ (latter 3/4 of trajectory guaranteed safe)
Table 5.1: Comparison of various types of safe rendezvous trajectories. Fuel costs in mm/s.

<table>
<thead>
<tr>
<th></th>
<th>No Safety, Nominal</th>
<th>Passive Safety</th>
<th>Passive Safety, invariant</th>
<th>Active Safety, a priori</th>
<th>Active Safety, a priori invariant</th>
<th>Active Safety, optimized</th>
<th>Active Safety, optimized, invariant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety Cost</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.28</td>
<td>6.28</td>
<td>0.76</td>
<td>2.14</td>
</tr>
<tr>
<td>P(collision)</td>
<td>0.012</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.1 compares various approaches for creating rendezvous trajectories with and without safety using the same initial conditions as the examples in Figures 5-15-5-18. The first row refers to the fuel cost (mm/s) of implementing the nominal rendezvous trajectory. The second row gives the cost of implementing a full safe input sequence (mm/s). The last row gives the probability of collision for the trajectory using the method introduced in Section 5.5. The columns compare the fuel-optimal path with no safety to the passive safety path and paths using active safety. The active safety columns labeled \textit{a priori} use the predefined safe input sequence approach in Eq. 5.28 and the columns labeled \textit{optimized} use the approach in Eq. 5.30. The columns marked invariant also use the invariance constraints in Eq. 5.25. It is notable that for the example in the table, the probability of collision for the fuel-optimal trajectory \textit{(i.e., no safety)} is 0.012, but the addition of passive or active safety to the problem causes the probability to drop to zero. Note that this probability is predicated on the assumption that a failure is identified within a time-step of its occurrence and that the thrusters can be turned off (for passive safety) or used nominally (for active safety).

Passive safety requires more fuel for rendezvous than the case without safety, however, active safety with an optimized safe input sequence has the same cost as the fuel-optimal case. In the cases where invariance is imposed as a constraint, the fuel cost using active optimized safety is lower than the passive invariance case, but not as low as the optimal trajectory. Thus, for these initial conditions, both the nominal trajectory and the safe abort trajectory must be shaped to achieve active
invariance. The cost of safety for the nominal and passive safety trajectories is zero, because those cases do not consider safety and do not require thrusting for failures, respectively. In each active safety case, the safety cost is very small compared to the cost of the nominal trajectory, indicating that it should be possible to implement active safety on a space mission without significantly increasing the ΔV budget.

5.8.2 Active Safety for Thruster Failures

The active safety approach in Eq. 5.30 can be modified to guarantee safety for cases of individual thruster failure by optimizing multiple safe input sequences. Each safe input sequence is constrained to only use a single thruster direction, or alternately, a single thruster assuming that thrusters act through the center of gravity. This guarantees that if only one thruster fails, another safe trajectory which does not use the failed thruster still exists. Thus, in a system with at least two thrusters, any single thruster failure to the off state will be in the set of possible system failures covered by active safety. Likewise, if thrusters in the system can be used to cancel each other (e.g., a system with axial thrusters) then this active safety extension can also be used in the presence of thruster-on failures. In that case, the thruster opposite that which failed can be used to cancel erroneous thrusting while a thruster in another direction can be used to enact a preplanned safe input sequence.
Modifying Eq. 5.30 to include multiple safe input sequences yields

\[
x_{FT_k} =
\begin{cases}
\begin{bmatrix}
\Gamma_k S(T) & \Gamma_k S(T - k) H_x & 0 \\
\Gamma_k S(T) & 0 & \Gamma_k S(T - k) H_y \\
\end{bmatrix}
\begin{bmatrix}
U_k \\
V_x^k \\
V_y^k \\
\end{bmatrix}
+ A_{d}^k x_0, & k \leq N, \\
A_d^k x_0, & k - T \leq N_s
\end{cases}
\begin{cases}
\begin{bmatrix}
\Gamma_k S(T) & A_d^{k-N} \Gamma_{N_x} H_x & 0 \\
\Gamma_k S(T) & 0 & A_d^{k-N} \Gamma_{N_y} H_y \\
\end{bmatrix}
\begin{bmatrix}
U_N \\
V_x^{N_x} \\
V_y^{N_y} \\
\end{bmatrix}
+ A_{d}^k x_0, & k \leq N, \\
A_d^k x_0, & k - T > N_s
\end{cases}
\begin{cases}
\begin{bmatrix}
A_d^{k-N} \Gamma_{N_x} S(T) & A_d^{k-N} \Gamma_{N_x} H_x & 0 \\
A_d^{k-N} \Gamma_{N_y} S(T) & 0 & A_d^{k-N} \Gamma_{N_y} H_y \\
\end{bmatrix}
\begin{bmatrix}
U_N \\
V_x^{N_x} \\
V_y^{N_y} \\
\end{bmatrix}
+ A_{d}^k x_0, & k > N, \\
A_d^k x_0, & k - T > N_s
\end{cases}
\end{align}
\] (5.31)

where \( V_{N_x}^x \) is the safe input sequence of only \( x \)-direction inputs, \( V_{N_y}^y \) is the safe input sequence of only \( y \)-direction inputs, and \( H_x \) and \( H_y \) are matrices that extract only elements of \( \Gamma \) pertaining to \( u_x \) inputs and \( u_y \) inputs, respectively. The active safety algorithm remains the same, but the failure trajectory used in formulating Eq. 5.29 must be propagated using Eq. 5.30 instead of Eqs. 5.28 or 5.30.

Figure 5-19 shows an example trajectory using the multi-solution active safety form in Eq. 5.31 to solve the safe rendezvous problem for the initial conditions used in Table 5.1. The left side of the figure shows the nominal rendezvous trajectory and the safe trajectories that would be used in the event of a failure in the \( \pm y \) direction thruster (resulting from using \( V_{N_y}^y \)). The right side shows the same nominal trajectory, but the safe trajectories shown correspond to \( V_{N_x}^x \). In this case, a single optimization has produced two sets of safe input sequences, either valid at any time.
Fig. 5-15: Rendezvous trajectories using active safety.

The nominal trajectory in this case requires 13.67 mm/s of fuel, equivalent to the fuel-optimal, “unsafe” trajectory. The cost of safety for implementing $V_{N_x}^x$ is 1.43 mm/s and for $V_{N_y}^y$ is 0.96 mm/s, which follows the trend of low safety trajectory costs observed in Table 5.1.

The algorithm for using passive safety only requires that thrusters be disabled in the event of a failure and active safety only requires that a predetermined safe input sequence be used. The implementation algorithm for the modified active safety formu-
Fig. 5-16: Rendezvous trajectories using active safety with invariance constraints.

Fig. 5-17: Rendezvous trajectories using optimized active safety.
lation in this section requires an additional input from the spacecraft fault detection and isolation system which indicates the type of fault. In the case of thruster failure, this would also need to include which thruster failed and the nature of the failure. This additional information enables the active safety implementation to choose the appropriate safe input sequence to use.

5.8.3 Mitigating Impact of Process Noise and Navigation Error

The safety formulation in Equations 5.12 and 5.14 assumes that the state of the chaser spacecraft relative to the target spacecraft is precisely known. In practice, this relative state is only known to within the accuracy provided by the navigation system. Likewise, the propagation used in Eq. 5.12 is only as accurate as the linear dynamics used to formulate that equation, since the actual vehicle would be subject to nonlinear dynamics, and disturbances from effects such as drag, $J_2$, separation distance, and eccentricity. Equation 5.12 can also be rewritten to enable time-varying dynamics or an additional vector of modeled disturbances can be added to the problem without increasing the complexity of the resulting optimization [84]. This permits a more sophisticated dynamics model to be used, which could reduce some of the effects of modeling error [120]. To account for navigation error, the constraints in Eq. 5.14 can be made robust by posing them multiple times for a representative sampling of possible initial states that cover the space of likely navigation errors. Reference [115] introduces such an approach and an algorithm for minimizing the effect of robustness constraints on the size of the resulting optimization. Figure 5-20 shows a safe trajectory optimized using the same initial conditions as those used to create Table 5.1. In the figure, active safety with guaranteed collision avoidance for the last 3/4 of the trajectory is used with the addition of robustness to initial condition uncertainty. In this case, the initial velocity of the chaser is only known to within $\pm 0.75$ mm/s in the radial direction and $\pm 0.0002$ mm/s in the in-track direction. The resulting trajectory requires 13.84 mm/s nominally and the safe input
**Fig. 5-18:** Rendezvous trajectories using optimized active safety with invariance constraints.

**Fig. 5-19:** Rendezvous trajectories for using active safety optimized for two possible thruster failure directions.
Robustness to Uncertainty

• Robust Example using previous setup & ICs
  – Nominal cost: 13.84 mm/s  Safety cost: 2.1 mm/s
  – Cost w/ no safety: 13.67 mm/s

• Robustness method
  – Optimize using constraints for multiple ICs
  – Choose ICs to represent sensing uncertainty
  
  In this case:
  – ±0.75 mm/s radial vel
  – ±0.002 mm/s in-track vel

Fig. 5-20: Safe rendezvous trajectory with robustness to initial condition velocity uncertainty

sequence requires 2.1 mm/s. The cost of robustness in this case is less than 2% of the cost of the nominal safe trajectory without robustness. However, the problem is particularly sensitive to the amount of uncertainty present in the in-track velocity and can quickly become infeasible for larger uncertainties.

5.9 Summary

Safety in autonomous spacecraft rendezvous trajectory design allows abort with guaranteed collision avoidance for a class of anomalous system behaviors. This chapter introduced several online optimization formulations that guarantee passive and active safety and demonstrated in numerous simulations that the additional fuel costs are comparatively small, particularly relative to commonly considered suboptimal trajectories. Additional restrictions to guarantee failure trajectories that minimize drift
and guarantee long-horizon passive collision avoidance were shown to require fuel use on the same order of magnitude as optimized solutions that do not include safety guarantees. Approximate analytic methods for creating upper and lower bounds on the expected fuel use for several mission types yielded accurate estimates compared to optimized fuel costs. A convex formulation of the safety problem was introduced which uses approximately twice as much fuel, but more than 150 times less computation time than the nonconvex formulation. An active safety approach was developed and shown to be capable of achieving the same fuel costs as trajectories without safety while still guaranteeing collision-free escape trajectories for a large class of potential anomalies, including single thruster failures. The safety algorithms presented provide a fuel-efficient, computationally feasible framework for designing safe mode procedures for multi-spacecraft missions.
Chapter 6

Safe Docking Demonstrations on SPHERES

This section describes several safe autonomous rendezvous and docking experiments performed using the Synchronized Position Hold Engage and Reorient Experimental Satellites (SPHERES) hardware testbed aboard the International Space Station (ISS).

6.1 SPHERES Background

The SPHERES testbed (shown in Fig. 6-1) is presently in orbit aboard ISS. The SPHERES testbed is intended to facilitate the development and evaluation of control algorithms for spacecraft formation flying and autonomous rendezvous and docking. Three SPHERES satellites are onboard the ISS as of December, 2006. Each satellite has a mass of 4.2 kg and is capable of producing a 0.22 N thrust in any axial direction [78]. The SPHERES experiments flown on ISS are first tested on the SPHERES flat table testbed in the MIT SSL laboratory. The flat table testbed uses the same type of satellites that are aboard ISS, but mounted on air carriages so that the vehicles experiment double integrator dynamics in two dimensions. The mass of the combined satellite/air carriage system is 12.4 kg.
6.2 Safe Autonomous Rendezvous and Docking on the SPHERES Testbed

The safe rendezvous and docking algorithms introduced in Chapter 5 were demonstrated on the SPHERES testbed onboard the ISS. The SPHERES microsatellites did not contain a linear programming solver at the time of the safe rendezvous and docking experiment, so instead of solving for an optimized rendezvous trajectory online, a trajectory was optimized offline and flown onboard the microsatellites as a series of waypoints. The waypoints were followed using the SPHERES PID-controller [78]. The waypoints were stored in an LVLH frame in which the target and chaser satellites both begin with a specified separation distance on the $y$ axis and no offset from the $x$ or $z$ axes. During the experiment, the chaser satellite follows the waypoints in the
trajectory and the target satellite uses a station-keeping controller to stay stationary at the origin. Onboard ISS, the LVLH frame for the trajectory-following is oriented according to the following algorithm:

1. Chaser and target satellites point toward each other

2. Chaser adjusts the distance between itself and the target to the distance of the initial waypoint in the rendezvous trajectory.

3. LVLH frame is aligned to the chaser body-fixed frame

Following this frame orientation procedure allows the satellites to begin from most positions in the feasible operating areas for SPHERES on ISS. The initialization instructions for the astronauts specify that the target spacecraft should placed near the center of the operating area for SPHERES and the chaser satellite should be placed nearby.

To design the safe rendezvous trajectory for use in the ISS experiment, the system dynamics used were double integrators, because the operating area inside the ISS is sufficiently small that the Earth gravity gradient effects are minimal. In the notation of Chapter 5, double integrator dynamics are

\[
A_k = A = \begin{bmatrix} I_3 & \Delta t I_3 \\ 0_3 & I_3 \end{bmatrix}, \quad B_k = B = \begin{bmatrix} \frac{(\Delta t)^2}{2} I_3 \\ \Delta t I_3 \end{bmatrix} \quad \forall k
\] (6.1)

Two experiment scenarios were prepared for testing on ISS: safe docking with a stationary target and safe docking with a rotating target. During the ISS test session, the astronaut conducting the SPHERES test only had time to conduct the stationary safe docking experiments. This chapter will present the scenario for rotating docking as well and describe an alternate test that was performed using the SPHERES flat-table testbed in the MIT Space Systems Laboratory (SSL). All optimized rendezvous trajectories presented in this chapter assume that the chaser satellite has a mass of 12.4 kg so that the resulting trajectory can be tested on the flat table testbed or on the ISS testbed. Because only the state waypoints in the trajectory are used
as inputs to the SPHERES flight software and not the optimized thrust input, the
effect of the spacecraft mass parameter only serves to determine the effective control
authority available at each step in the plan. Passive safety will be tested by randomly
disabling the chaser satellite’s thruster during one of the last 5 time steps and leaving
the thrusters disabled for the remainder of the trajectory. If no collision occurs, then
the actual trajectory that the satellite followed will have been passively safe.

The optimized safe rendezvous path for the stationary safe rendezvous experiment
is shown in Figures 6-2 and 6-3. The chaser satellite begins with an 80 cm offset
from the target. This distance was chosen to balance the preference for a large
approach distance within the ISS SPHERES operating area against the danger of
losing metrology near the edges of the working area. The optimized trajectory in
this cases enforces safety for the last 5 steps of the trajectory. Because of the double
integrator dynamics, the trajectory in the event of a failure continues to follow the
path of the trajectory before the failure. This is particularly challenging for a safe
rendezvous problem, because any direct approach to target spacecraft would collide
in the event of a failure. The optimized safe trajectory approaches in such a way that
it succeeds in passing through the docking region even in the event of a failure. Thus,
one would expect failure, as well as nominal, trajectories to dock successfully.

Figures 6-4 and 6-5 show the optimized safe rendezvous path for docking with a
rotating target. The initial state of the chaser spacecraft is half distance from the
target as in the stationary case, because the chaser will need space to travel in a
line-of-sight cone that rotates with the target. The target rotates at 0.039 rad/s,
which was chosen to allow the experiment to complete in under two minutes (the
maximum designated duration). The dynamics of docking with a rotating spacecraft
force the velocity vector of the chaser satellite to change direction constantly. As a
result of this, the problem of guaranteeing safety is significantly less constrained for
the rotating case, allowing the last 20 steps (40 s) of the 40 step trajectory to be
guaranteed safe. For failures earlier in the guaranteed-safe region of the trajectory,
the expected abort trajectory would drift away from the target. A failure occurring
in the last step of the rendezvous trajectory would result in a passive rendezvous.
Fig. 6-2: Planned safe trajectory for stationary rendezvous: $x$-$y$ view. Note that trajectory $z$ offset is zero at all times.

Fig. 6-3: SPHERES Microsatellite in air carriage on flat table testbed: Perspective view.
Fig. 6-4: Planned safe trajectory for rotating rendezvous: x-y view. Note that trajectory z offset is zero at all times.

Fig. 6-5: Planned safe trajectory for rotating rendezvous: testbed: Perspective view
6.3 Flight Experiment Results

The results of the ISS experiment to demonstrate nominal safe docking are shown in Figures 6-7, 6-8, 6-9, and 6-10. The experiment begins with the target and chaser satellites deployed near the center of the ISS SPHERES feasible operating space. After waiting for the satellites’ estimators to converge, the vehicles adjust their attitudes to point at one another. Then, the chaser satellite adjusts its distance along the pointing vector between the two spacecraft until it achieves the initial separation specified in the pre-optimized rendezvous trajectory. In the figures below, the line corresponding to "Sat1" refers to the chaser and the line corresponding to “Sat2” refers to the target. Compared to previous test sessions, the trajectory following error was significantly reduced, with a maximum of less than 5 cm of error. The reasons for this were likely twofold: in this test session, the 5 beacon estimator was used, greatly improving the accuracy of the estimates; and the trajectory following in this test session was accomplished using a PID controller instead of a PD controller. However, it should be noted that the greatest trajectory error occurred when the desired trajectory experienced its highest rate of change, indicating that the bandwidth of the desired trajectory should be reduced for future experiments using the same tracking controller. The goal of this experiment was for the chaser satellite to dock with the target satellite and the data indicates that this goal was achieved successfully 6-6.

The results for the demonstration of a stochastic failure in the safe trajectory are shown in Figs. 6-11, 6-12, 6-13, and 6-14. The goal of this test was to follow a trajectory demonstrating safe rendezvous with a simulated failure toward the end of the trajectory. The time of the failure is decided online in the SPHERES flight software at the time the test begins. The only restriction on the failure time is that it occur during a window in which the optimized trajectory has been guaranteed to be safe in presence of failures. Since the optimized trajectory is chosen prior to the failure time, this test is a demonstration of the safety algorithm to handle arbitrary failures that occur during the time. Ideally, a test such as this one would be run many times as a stochastic demonstration of the safety properties of the safe rendezvous
and docking trajectory optimization algorithm.

The results in Figs. 6-11, 6-12, 6-13, and 6-14 indicate that the stochastic failure experiment proceeded nearly identically to nominal experiment until the end of the trajectory when the simulated failure occurred. At that point, as anticipated, the spacecraft disabled its thrusters and entered a drift mode. As in the nominal experiment, the target held its position with very little state error and the chaser spacecraft followed its trajectory very accurately. Also, as in the previous case, the maximum trajectory-following error occurred at the “knee” in the desired trajectory where the rate of change reached its maximum. The trajectory followed after the thruster failure occurred was very close to the expected trajectory in the event of a failure. According to the telemetry gathered in flight (truth data from video footage is not yet available) failure trajectory entered the predefined terminal docking box and successfully docked even for the failure case. Although a safe trajectory is defined specifically in terms of collision avoidance, the terminal docking box was not inside failure avoidance region and so the resulting rendezvous is not inconsistent with a safe maneuver.
Fig. 6-7: Position trajectory followed during nominal docking ISS experiment: $x$-$y$ view

Fig. 6-8: Position time-series for $z$ axis during nominal docking ISS experiment
Fig. 6-9: Velocity time-series view during nominal docking ISS experiment

Fig. 6-10: Velocity trajectory followed during nominal docking ISS experiment: x-y view
**Fig. 6-11:** Position trajectory followed during failed docking ISS experiment: $x$-$y$ view

**Fig. 6-12:** Position time-series for $z$ axis during failed docking ISS experiment
**Fig. 6-13:** Velocity time-series view during failed docking ISS experiment

**Fig. 6-14:** Velocity trajectory followed during failed docking ISS experiment: $x$-$y$ view
6.4 Flat Table Experiment Results

The results in Figs. 6-15, 6-16, 6-17, 6-18 show a rotating target rendezvous experiment with a stochastic thruster failure inserted. The failure occurred at the 38th step in the 40 step rendezvous trajectory. Fig. 6-15 shows that the target spacecraft performed its station-keeping task to within centimeters of the origin. Likewise, the chaser spacecraft also tracked its desired safe rendezvous trajectory well. After the simulated failure occurred, the chaser spacecraft drifted away from the chaser along an unforced trajectory. Video of the experiment indicates that the chaser vehicle did not make contact with the target. The velocity tracking data in Figs. 6-17 and 6-18 is considerably more error prone than that of the velocity tracking for the ISS experiment. This is likely the product of additional disturbances entering the flat table environment from the air carriage system and imperfections in the degree to which the table was perfectly level and flat. Future experiments should consider incorporating additional uncertainty robustness into the trajectory design.

6.5 Summary

The safe autonomous rendezvous and docking tests performed aboard the International Space Station using the SPHERES testbed successfully demonstrated a safe rendezvous with a stationary target and a successful rendezvous after a stochastic thruster failure. A similar experiment for a rotating rendezvous trajectory demonstrated collision avoidance on the flat table testbed.
Fig. 6-15: Position trajectory followed during failed docking flat table experiment: \(x-y\) view

Fig. 6-16: Position time-series for \(z\) axis during failed docking flat table experiment
Fig. 6-17: Velocity time-series view during failed docking flat table experiment

Fig. 6-18: Velocity trajectory followed during failed docking flat table experiment: $x$-$y$ view
Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis has presented, analyzed, and applied several new technologies related to autonomous control of spacecraft in proximity operations. Future missions falling into that broad category include formations of spacecraft cooperating to achieve common goals and autonomous rendezvous of spacecraft. The ideas explored in this thesis focus on commonalities between these mission types: an emphasis on relative dynamics, adherence to inter-vehicle performance constraints, and preventing relative drift. The following subsections summarize the contributions of this work.

7.1.1 Linearized Relative Dynamics

Chapter 2 derived and validated linearized relative spacecraft dynamics that incorporate the effects of Earth oblateness and are valid for widely separated, eccentric orbits. Linearization and discretization assumptions were examined and shown to be valid for modeling the relative trajectories within expected error box sizes for several planned missions (including MMS). The new dynamics extend the range of missions in which linear controllers and linear optimization-based model predictive controllers can be used. The enhanced capabilities were demonstrated by embedding the dynamics in a model predictive controller to control a high fidelity multi-week simulation of
a four spacecraft MMS-like mission. The results showed that the controller is reliable
and that formation flying using MPC with $J_2$-modified GVEs requires fuel use that
is comparable to using unmodified GVEs in simulations that do not include the $J_2$
effects.

### 7.1.2 Optimized Initialization of Semi-Invariant Orbits

Chapter 3 introduced an approach for optimizing $J_2$ invariance between spacecraft
that explicitly minimized the fuel use required to achieve the invariant states. This
approach also allowed weights to be assigned to the relative emphasis on invariance,
minimizing fuel use, and maintaining a desired fleet relative configuration. The al-
gorithm extends previous approaches to $J_2$ invariance by explicitly considering arbi-
trary time frames, as opposed to the infinite horizon, and by allowing the osculating
Cartesian drift to penalized, as opposed to the secular mean drift. Furthermore,
the approach can be formulated to use a linear optimization, enabling a fast online
implementation, whereas previous methods for optimizing multi-objective formation
initial conditions relied on highly nonlinear optimization techniques. The range of
possible optimized conditions were examined for example orbits in LEO and HEO.
The optimized conditions were compared to analytic solutions and found to provide
similar levels of drift at lower maneuver cost and more closely matching desired geo-
metric configurations. In a formation where the principal control objective is to “not
drift,” the proposed approach can be used for fuel-optimized online initialization of a
formation flying mission.

### 7.1.3 Control of Spacecraft Formations Using MPC

Chapter 4 demonstrated the use of the trajectory dynamics and optimization for-
mulation developed in Chapter 2 in a closed-loop model predictive controller. This
controller was applied to a proposed formation flying mission and demonstrated in
a series of high fidelity multi-week simulations, which modeled all realistic orbital
disturbances and navigation uncertainty. The simulations are the first instance in the
literature of formation flying spacecraft using variable thrusting constraints to enforce passive observation periods with error box constraint satisfaction. The effects of the passive observation constraints on fuel-use and feasibility are examined, as are the effects of imposing passive invariance as a control constraint and several types of error box relaxation. These simulations demonstrated that model predictive control using the new linearized dynamics can be used to control an actual mission with realistic control requirements.

7.1.4 Safety in Autonomous Rendezvous and Docking

Chapter 5 introduced a new approach for generating rendezvous trajectories that are safe in the presence a large class of system failures. This approach is flexible (i.e., same problem formulation captures many rendezvous situations including tumbling targets), fuel-optimized, and computationally feasible. Previous rendezvous approaches in the literature which provide guaranteed safety do not minimize fuel use or are not computationally feasible for online implementation. The new safe formulation was evaluated using a stochastic analysis and determined to improve safety (i.e., reducing the likelihood of a failure resulting in a collision) with minimal additional fuel costs over fuel-optimal trajectories. Approximate, but verifiable, analytic bounds were found for the maximum and minimum fuel that a safe trajectory would require. Additional restrictions to guarantee failure trajectories that minimize drift and guarantee long-horizon passive collision avoidance were shown to require fuel use on the same order of magnitude as optimized solutions that do not include safety guarantees. A convex formulation of the safety problem was introduced which uses approximately twice as much fuel, but more than 150 times less computation time than the nonconvex formulation. An active safety approach was developed and shown to be capable of achieving the same fuel costs as trajectories without safety. The active approach still guarantees collision-free escape trajectories for a large class of potential anomalies, including single thruster failures. Previous approaches to planning active abort strategies were planned offline or did not use fuel minimization. The safe trajectory generation algorithm is the first computationally feasible, fuel-optimized
approach for guaranteeing passive and/or active safety for rendezvous with stationary and tumbling targets.

7.1.5 Safe Autonomous Rendezvous Demonstration on Orbit

Chapter 6 used the safe trajectory generation algorithm from Chapter 5 to generate safe rendezvous trajectories for the SPHERES hardware testbeds. Safe trajectories were demonstrated onboard the International Space Station and on a flat table. Trajectories were followed in both nominal and simulated-failure configurations, demonstrating the feasibility of following automatically-generated fuel-minimizing safe trajectories in space. The on-orbit demonstration is the first known instance of a spacecraft rendezvous following a fuel-minimized trajectory guaranteed to be safe up to the moment of docking.

7.2 Future Extensions

The work presented in this thesis addresses several areas where spacecraft formation flying and spacecraft rendezvous and docking technologies overlap. Optimizations using relative dynamics and exploiting the concept of passive relative invariance are used to reliably plan fuel-efficient trajectories in both mission types. Imposing safety as a constraint on rendezvous and docking trajectories was examined from the unique perspective of constraining the paths followed by a spacecraft in the event of a failure. This approach has the immediate benefit of collapsing an enormous space of possible multi-vehicle safe mode responses into a simple, guaranteed-safe plan for one spacecraft and continued nominal operation for the other. This same approach could be applied to formation flying spacecraft as well, potentially greatly decreasing the complication of considering the space of potential off-nominal interactions between arbitrary numbers of vehicles in close proximity. Adding convex safety constraints to formation flying trajectories generated using the MPC formulation in Chapter 2 may prove to be a simple, computationally inexpensive way to guarantee that a glitch in one spacecraft in a formation does not result in the catastrophic failure of two or
more spacecraft.

The formation initialization algorithm presented in Chapter 3 is presented in the form of a multi-objective optimization. However, the optimization provides no guarantee that a resulting trajectory will achieve certain minimal goals in its objectives, should those goals be critical mission objectives. A modified form of the initialization algorithm could be created that would provide constraint-based minimum guarantees for the optimized objectives. Also, the initialization algorithm as presented represents the beginnings of a hybrid between initial conditions and maneuver planning, two aspects of formation flying that are usually treated separately. Expanding on this idea to incorporate robustness or sensitivity to expected process and sensor noise would improve the approach. Further improvements might make maneuver-optimized initialization a viable option for closed-loop feedback control, in which each new trajectory optimization would simultaneously allow alterations of the formation configuration.

The safe rendezvous technique could be made safer through the addition of more uncertainty robustness. In particular, types of uncertainty not considered in the current formulation include: process noise (e.g., attitude errors and thruster magnitude errors) and uncertainty in the future trajectory of the target spacecraft. Adding guarantees for constraint satisfaction in the presence of these sources of error is a difficult proposition, because additional robustness will likely lead to less feasibility. The current navigation error robustness approach is computationally simple, but known to be overly conservative in a system using feedback optimization. Enhancing the current safe trajectory generation approach to have safety guarantees in the presence of and explicitly account for feedback are critical next steps.
Appendix A: Station-keeping Costs and Maneuver Costs for MELCO Formation Flying Mission

This appendix examines several LP-designed maneuvers as planned trajectories in an LVLH frame, a differential orbital element frame, and as thruster inputs. These trajectories were generated using the $J_2$-modified GVE-based planner described in Chapter 2. Each maneuver is planned to occur over the course of two orbits using a 10.8 second time step, with inputs allowed at all times. The large size of the formations examined in this section necessitated an additional step in formation design: after creating the desired LVLH states using the procedures in Ref. [109], the desired states were converted to relative orbital elements, where any desired differential semimajor axis was eliminated. This approach ensures that formations are based on passive-apertures, but still have the same period (i.e., the satellites do not drift with respect to each other over time).

The MELCO formation flying mission consists of two different formation types: 1) in-track separation formations and 2) in-track/cross-track passive aperture formations (triangular). Figures 7-1 and 7-2 show the average amount of fuel required to maintain various sizes of in-track formations and passive aperture formations, respectively. For each type of formation at each baseline, one or more four day simulations were conducted to determine the average fuel usage. Each formation size uses an error box that is 10% of the baseline, except the 50 meter passive aperture. In that case, the error box is sufficiently small that the navigation errors [73] strongly influence the
closed-loop behavior, so a $5 \times 10 \times 5$ meter (radial/in-track/cross-track) error box is used instead. This changes allows a slightly more natural relative elliptical motion in the radial/cross-track plane (typically a $1 \times 2$ ellipse).

Maneuver trajectories generated in this section cover a wide range of different maneuver types and are summarized in Table 7.1. Fuel use for each maneuver type is summarized in Figures 7-3-7-6. Maneuvers from an in-track formation configuration to a passive aperture examine two different types of passive apertures: projected in-plane and projected out-of-plane. For each passive aperture radius, the in-plane formations require significantly less fuel to create than the out-of-plane formations. Also, note that generally it requires less fuel to create an in-track formation than a passive aperture. The costs of the maneuver types in Table 7.1 are characterized by the cost coefficient $\alpha$, the approximate fuel cost of each maneuver, $\Delta V$ in m/s is given by

$$\Delta V = \alpha \times r$$  \hspace{1cm} (7.1)

where $r$ (in meters) is the baseline of a passive aperture or the separation between satellites for an in-track formation. The table shows how the expected maneuver costs change as additional orbit disturbances are modeled in the planner. To compute the cost maneuvers with no eccentricity modeled, the eccentricity of the reference orbit was set to zero and, similarly, cost computations with no $J_2$ effect used the dynamics matrices from Chapter 2 computed with the $J_2$ constant set to zero. Interestingly, in several cases, maneuvers required slightly less fuel when the effects of $J_2$ were modeled, indicating that the trajectory optimizer was able to use the additional natural dynamics complexity in lieu of some thrusting. This phenomenon also appears when comparing the column of no disturbances with the column where only eccentricity is modeled. However, in all cases with both eccentricity and $J_2$ effects, the costs are expected to be higher than the cases in which disturbances are only partially modeled. Note that one should not conclude that it is possible to use less complicated dynamics models to achieve fuel savings. It is still desirable to use the higher-fidelity models in the control formulation, because although they may initially produce more
**Fig. 7-1:** Fuel Required to Maintain an In-track Formation

**Fig. 7-2:** Fuel Required to Maintain a Passive Aperture Formation
Table 7.1: Table of maneuvers planned using combinations of $J_2$ and eccentricity. In the table, PA $\equiv$ passive aperture, IT $\equiv$ in-track formation, IP $\equiv$ in-plane passive aperture, and OP $\equiv$ out-of-plane passive aperture. The scalar $\alpha$ is the fuel use coefficient, where the approximate fuel use in m/s is $\alpha$ multiplied by the aperture size in meters.

<table>
<thead>
<tr>
<th>Type</th>
<th>From</th>
<th>To</th>
<th>$J_2$, Ecc $\alpha$</th>
<th>No $J_2$, Ecc $\alpha$</th>
<th>$J_2$, No Ecc $\alpha$</th>
<th>No $J_2$, No Ecc $\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PA (50m)</td>
<td>IT (50m-50km)</td>
<td>0.0001693</td>
<td>0.0001206</td>
<td>0.0001170</td>
<td>0.0001214</td>
</tr>
<tr>
<td>2</td>
<td>PA (50m)</td>
<td>PA (50m-5km)</td>
<td>0.005030</td>
<td>0.004935</td>
<td>0.004941</td>
<td>0.004941</td>
</tr>
<tr>
<td>3</td>
<td>IT (1km)</td>
<td>IT (50m-50km)</td>
<td>0.0001656</td>
<td>0.0001142</td>
<td>0.0001135</td>
<td>0.0001158</td>
</tr>
<tr>
<td>4</td>
<td>IT (1km)</td>
<td>PA (IP) (50m-5km)</td>
<td>0.005055</td>
<td>0.004959</td>
<td>0.004967</td>
<td>0.004967</td>
</tr>
<tr>
<td></td>
<td>PA (OP) (50m-5km)</td>
<td>0.01052</td>
<td>0.008272</td>
<td>0.008281</td>
<td>0.008281</td>
<td></td>
</tr>
</tbody>
</table>

costly trajectories, fewer future corrective actions should be required at the end of those trajectories.
(a) Cost of maneuvering from a 50m Passive Aperture (in-plane) to an In-track Formation.

(b) Cost of maneuvering from a 50m Passive Aperture (in-plane) to a Passive Aperture (in-plane).

(c) Cost of maneuvering from a 1km In-track Formation to an In-track Formation.

(d) Cost of maneuvering from a 1km In-track Formation to a Passive Aperture (in-plane=blue, solid; out-of-plane =red,dashed).

Fig. 7-3: Maneuver costs: plans created accounting for eccentricity and $J_2$ effects.
(a) Cost of maneuvering from a 50m Passive Aperture (in-plane) to an In-track Formation.

(b) Cost of maneuvering from a 50m Passive Aperture (in-plane) to a Passive Aperture (in-plane).

(c) Cost of maneuvering from a 1km In-track Formation to an In-track Formation.

(d) Cost of maneuvering from a 1km In-track Formation to a Passive Aperture (in-plane=blue, solid; out-of-plane =red,dashed).

**Fig. 7-4:** Maneuver costs: plans created accounting for eccentricity with no $J_2$ effects modeled
(a) Cost of maneuvering from a 50m Passive Aperture (in-plane) to an In-track Formation.

(b) Cost of maneuvering from a 50m Passive Aperture (in-plane) to a Passive Aperture (in-plane).

(c) Cost of maneuvering from a 1km In-track Formation to an In-track Formation.

(d) Cost of maneuvering from a 1km In-track Formation to a Passive Aperture (in-plane=blue, solid; out-of-plane =red,dashed).

Fig. 7-5: Maneuver costs: plans created accounting for eccentricity with no $J_2$ effects modeled
(a) Cost of maneuvering from a 50m Passive Aperture (in-plane) to an In-track Formation.

(b) Cost of maneuvering from a 50m Passive Aperture (in-plane) to a Passive Aperture (in-plane).

(c) Cost of maneuvering from a 1km In-track Formation to an In-track Formation.

(d) Cost of maneuvering from a 1km In-track Formation to a Passive Aperture (in-plane=blue, solid; out-of-plane =red,dashed).

Fig. 7-6: Maneuver costs: plans created not accounting for eccentricity with no $J_2$ effects modeled
Bibliography


