A Multiple-Team Organization for Decentralized Guidance and Control of Formation Flying Spacecraft

Joseph B. Mueller*
Princeton Satellite Systems, Princeton, NJ 08542

In recent years, formation flying has become an enabling technology for several mission concepts at both NASA and the Department of Defense. In most cases, a multiple-satellite approach is required in order to accomplish the large-scale geometries imposed by the sensing objectives. In general, the paradigm shift of using a multiple-satellite cluster rather than a large, monolithic spacecraft has also been fueled by the objectives of increased robustness, greater flexibility, and reduced cost. However, the operational costs of monitoring and commanding a large fleet of close-orbiting satellites is likely to be unreasonable unless the onboard software is sufficiently autonomous, robust, and reconfigurable.

This paper presents the prototype of a system that addresses these objectives – a decentralized guidance and control system that is distributed across spacecraft using a multiple-team framework. The system is designed to provide a high-level of autonomy, to support clusters with large numbers of satellites, to enable the number of spacecraft in the cluster to change post-launch, and to provide for on-orbit software modification. The real-time distributed system will be implemented in C++ using the MANTA environment (Messaging Architecture for Networking and Threaded Applications). In this architecture, tasks may be remotely added, removed or replaced post-launch to increase mission flexibility and robustness. This built-in adaptability will allow significant or simple software modifications to be made on-orbit in a robust manner. The prototype system, which is implemented in Matlab, emulates the task-based and message-passing features of the MANTA software.

In this paper, the multiple-team organization of the cluster is described, and the relative dynamics in circular and eccentric reference orbits is reviewed. Families of periodic, relative trajectories are identified and represented with static geometric parameters. An analytic solution for impulsive maneuvering is used for whole orbit-period control in circular orbits, and linear programming techniques are used to find time-weighted, minimum-fuel control solutions. Finally, the decentralized guidance law design is presented, with a comparison between the optimal and a sub-optimal assignment algorithm.

I. Introduction

Formation flying is now a key enabling technology for several mission concepts in both the scientific and defense arenas. Examples of currently planned NASA missions include MMS (Magnetospheric Multi-Scale), SIRA (Solar Imaging Radio Array), and TPF (Terrestrial Planet Finder). In each case, a multiple-satellite approach is required in order to accomplish the large-scale geometries imposed by the science objectives. In general, the paradigm shift towards using a multiple-satellite cluster has also been fueled by the perceived advantages of increased robustness, greater flexibility, and reduced cost. However, the costs, complexity and risk of managing a large fleet of spacecraft is likely to be unreasonable unless the onboard software is sufficiently autonomous, robust, and reconfigurable.

This paper presents the prototype of a system that addresses these objectives – a decentralized guidance and control system that is distributed across spacecraft using a multiple-team framework. The system is designed to provide a high-level of autonomy, to support clusters with large numbers of satellites, to enable the number of spacecraft in the cluster to change post-launch, and to provide for on-orbit software modification. The real-time distributed system will be implemented in C++ using the MANTA environment (Messaging Architecture for Networking and Threaded Applications). MANTA (formerly ObjectAgent), has

*Senior Technical Staff, 33 Witherspoon St., jmueller@psatellite.com, (763) 561-9246. Member AIAA.

been developed at Princeton Satellite Systems over the past 5 years to support the implementation of real-time, distributed control systems. In this architecture, tasks may be remotely added, removed or replaced post-launch to increase mission flexibility and robustness. This built-in adaptability will enable software modifications to be made on-orbit in a robust manner. The prototype system, which is implemented in MATLAB, emulates the task-based and message-passing features of the MANTA software.

In this paper, the multiple-team organization of the cluster is described, and the relative dynamics in circular and eccentric reference orbits is reviewed. Families of periodic, relative trajectories are identified and represented with static geometric parameters. An analytic solution for impulsive maneuvering is used for whole orbit-period control in circular orbits, and linear programming techniques are used to find time-weighted, minimum-fuel control solutions. Finally, the decentralized guidance law design is presented, with a comparison between the optimal and a sub-optimal assignment algorithm.

II. Decentralization Using a Multiple-Team Framework

The design of a guidance, navigation and control system for formation flying spacecraft is driven by several competing objectives. We simultaneously seek to reduce the level of inter-spacecraft communication, distribute the computational effort across satellites, and maintain complete control over the cluster. These objectives are the result of physical constraints associated with the navigation sensors, communication bandwidth, and processor throughput. The problem leads to an architectural decision for the control system, based upon the trade-offs between a *centralized* and a *decentralized* approach.

In a purely centralized system, one satellite serves as a hub or a reference, while all other satellites control their trajectory relative to it. This can potentially place a considerable burden on the relative navigation hardware and software of the reference satellite. Since this satellite has complete knowledge of the cluster, it is also burdened with the computationally-intensive tasks of collision avoidance, guidance and control. The advantage of this approach is that it provides global control over the cluster, and requires a minimum level of inter-spacecraft communication. Nominally, only the desired trajectory and/or control is transmitted, which typically requires very little bandwidth.

In a purely decentralized system, each satellite has awareness of all other satellites in the cluster. In this sense, a global relative state is shared throughout, and control is applied independently at each node. It has been shown that the data transmission requirements can be minimized with a decentralized LQG control approach.[1] The main advantage of this approach is that it distributes the computational load equally throughout the cluster. However, it is likely that a significant burden will continue to be placed on the relative navigation system of each satellite. As the number of satellites in the cluster grows, it becomes increasingly difficult to realize a purely centralized or purely decentralized system. A novel solution to this problem has been developed, where the satellites are organized as a connected framework of teams. The cluster is decentralized, in that each team is controlled independently. Global control over the cluster is still maintained, however, as the teams are connected in a hierarchical framework. Figure 1 illustrates a multiple-team organization of 16 satellites.

In this example, the cluster is divided into four teams. Each team consists of a *reference* and one or more *relatives*, where the reference defines the origin of that team's relative frame. The reference of each team is shown above the relatives. All four teams are connected in a hierarchical fashion, with the relative of one team also serving as the reference to another. For example, Team A is connected to Team B in that one of the Team A relatives serves as the reference for Team B. At the top of the hierarchy, the reference of Team A is considered the *cluster reference*.

Within a given team, the relative satellites control their trajectories with respect to the reference. The reference only maneuvers in two possible cases: 1) if it is a relative member of another team, or 2) if it is the cluster reference performing a station-keeping maneuver. In each case, the control data

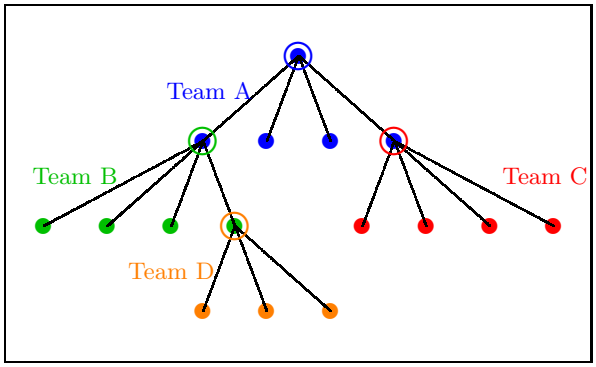

Figure 1. Example of a Multiple-Team Framework

may be transmitted to the relatives and applied in a feed-forward manner. The maneuver planning for each member is conducted locally, so that the computational effort is distributed throughout the team. A significant portion of the guidance law is distributed, as well, with each member estimating the costs to achieve all possible target states. This cost data from each member is returned to the team *captain*, who applies an assignment algorithm to determine the optimal configuration. At any given time, one spacecraft in the team serves the role of captain, and is responsible for carrying out any tasks that require coordination. The guidance law is discussed in detail in Section VI on page 15.

A. Software Architecture

A comprehensive software architecture has been developed that provides decentralized formation flying (DFF) guidance and control capability within this type of multiple-team framework. A high-level block diagram of the DFF system is shown in Figure 2. The green blocks on either side represent external systems, and the red circles indicate software interfaces to those systems. The architecture is designed for a real-time implementation within the C++ MANTA environment. As such, the system is composed of individual tasks that communicate with one another via messages. A prototype of the system has been completed in MATLAB. The message-passing functionality is emulated in the prototype to facilitate a more direct transition to C++.

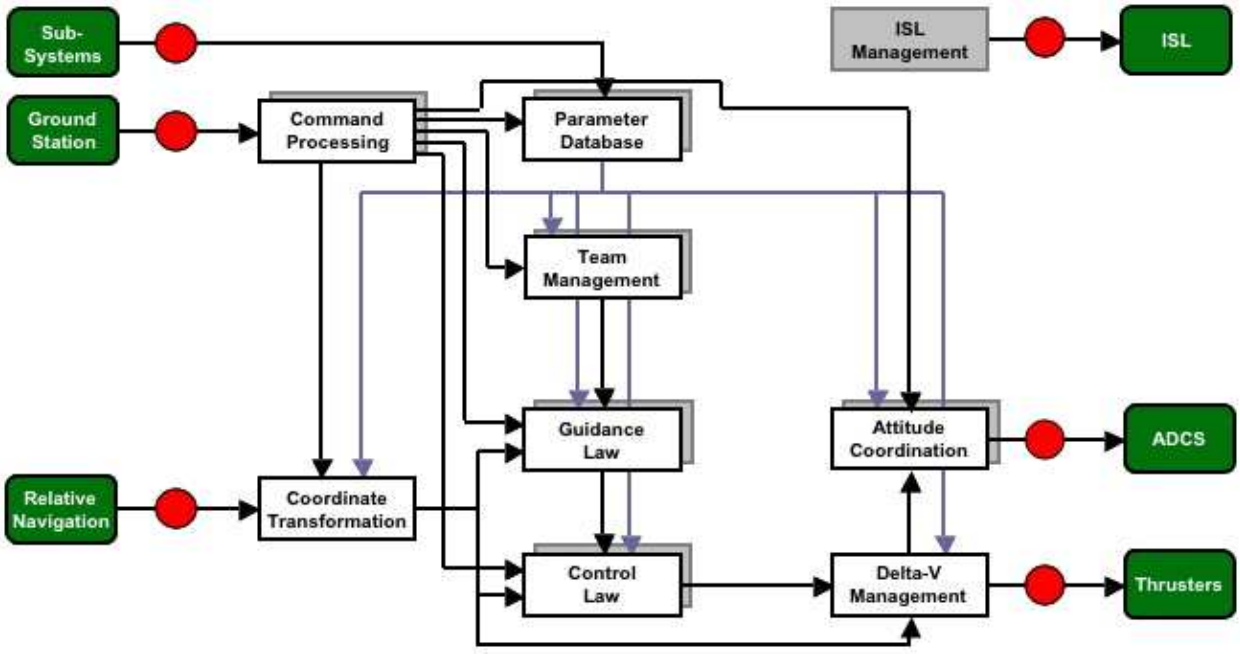

Figure 2. Block Diagram of the Task-Based Software Architecture for the DFF System

The system has been designed to be as generic as possible, so that it may be used for a variety of formation flying missions. Each of the white blocks in Figure 2 represents a high-level task that is carried out on each spacecraft. The shadowed blocks indicate tasks that require communication with other spacecraft. Inter-spacecraft communication is handled with an "ISL Management" task, which is designed to enable fault-tolerant message-passing throughout the cluster. The "Parameter Database" serves as a central repository for various types of data required by the other tasks. It is used at the beginning of the mission to initialize all of the generic tasks, and is used throughout the mission for parameter changes and telemetry requests initiated by the ground station. Note that relative navigation is treated as an external module. Because the design of a relative navigation system is highly dependent upon the spacecraft and mission characteristics, it is not included in the DFF design.

The "Team Management" task facilitates the multiple-team organization of the cluster. It provides autonomous team formation capability, and maintains the information that defines the hierarchical team framework. Satellites may be added to or removed from teams dynamically, and the roles of reference and supervisor may be changed throughout the mission.

The guidance and control functionality is suitable for a wide range of spacecraft designs, and supports all possible types of natural formations. The algorithms have been developed for both circular and eccentric reference orbits. The details of the guidance and control algorithms are discussed in later sections. First, however, a description of the relative dynamics is required.

III. Dynamics of Relative Motion

The equations governing the relative motion of close-orbiting satellites have been evolved over the course of several decades. The widely known Clohessy-Wiltshire equations,[2] or Hill's equations, provide a linear time-invariant solution for the relative motion in a circular reference orbit. Lawden's equations[3] express the relative motion in eccentric orbits, with true anomaly as the independent variable. In recent years, Lawden's equations have been used to generate periodic relative trajectories in eccentric orbits.[4] In addition, recent work by Broucke[5] has provided a nonsingular solution to elliptic relative motion with time as the independent variable.

Relative motion is defined in a rotating reference frame, with the origin fixed to a reference point that orbits with the cluster. In this paper, the reference is always defined as one of the spacecraft. Figure 3 shows the relative frame, hereafter referred to as Hill's frame. The x-axis points in the zenith direction, the z-axis is aligned with the angular momentum vector, normal to the orbital plane, and the y-axis completes the right-hand system. For circular orbits, y is always aligned with the velocity vector.

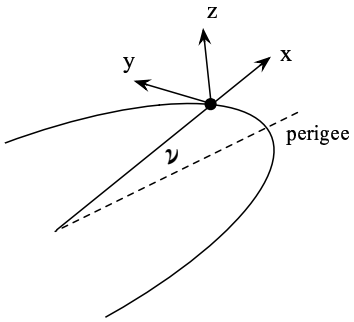

Figure 3. Relative Frame

The LTV dynamic equations of a satellite's relative motion in Hill's frame are given as follows:

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

where ν is the true anomaly, e is the eccentricity, n is the mean orbit rate, and a is the applied acceleration. In the case of zero eccentricity, the $\dot{\nu}$ terms reduce to n, the $\ddot{\nu}$ terms vanish and the familiar Hill's equations are obtained. The solution to Hill's equations may be found through a standard Laplace transformation.[6] The relative position for circular reference orbits is given below as a function of time:

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

$$y(t) = \left(6x_0 + 4\frac{4\dot{y}_0}{n}\right)\sin(nt) + \frac{2\dot{x}_0}{n}\cos(nt) - (6nx_0 + 3\dot{y}_0)\,t + \left(y_0 - \frac{2\dot{x}_0}{n}\right) \tag{3}$$

$$z(t) = z_0\cos(nt) + \frac{\dot{z}_0}{n}\sin(nt) \tag{4}$$

where the initial state is defined as $[x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0]^\top$ and t is measured from the time at which the initial state is defined. The velocity terms are found by simply taking the time-derivative of each position equation.

In the general case where $e > 0$, the solution is presented as a function of true anomaly. We use the procedure outlined in the journal article by Inalhan, Tillerson and How[4] to solve for the relative motion in eccentric orbits.

$$x(\nu) = \sin\nu\left[d_1 e + 2d_2 e^2 H(\nu)\right] - \cos\nu\left[\frac{d_2 e}{(1+e\cos\nu)^2} + d_3\right] \tag{5}$$

$$y(\nu) = \left[d_1 + \frac{d_4}{(1+e\cos\nu)} + 2d_2 e H(\nu)\right] + \sin\nu\left[\frac{d_3}{(1+e\cos\nu)} + d_3\right] + \cos\nu\left[d_1 e + 2d_2 e^2 H(\nu)\right] \tag{6}$$

$$z(\nu) = \sin\nu\left[\frac{d_5}{(1+e\cos\nu)}\right] + \cos\nu\left[\frac{d_6}{(1+e\cos\nu)}\right] \tag{7}$$

where the d_i parameters are integration constants that are determined from the initial conditions. Also, E is the eccentric anomaly, and the term $H(\nu)$ is defined as:

$$H(\nu) = \int_{\nu_0}^{\nu} \frac{\cos\nu}{(1+e\cos\nu)^3} d\nu = -(1-e^2)^{-5/2}\left[\frac{3eE}{2} - (1+e^2)\sin E + \frac{e}{2}\sin E\cos E + d_H\right] \tag{8}$$

where d_H is found by setting $H(\nu_0)$ to 0 at the initial true anomaly, ν_0. The velocity terms are found by first taking the derivative of each position equation with respect to ν. Conversion to the time-domain is then achieved by multiplying the angular derivatives by $\dot{\nu}$, where

$$\dot{\nu} = \frac{n\left(1+e\cos\nu\right)^2}{\left(1-e^2\right)^{3/2}} \tag{9}$$

The complete relative state, including the velocity terms, may be expressed at any true anomaly as a linear combination of the 6 integration constants.

$$\begin{bmatrix} x(\nu) \\ y(\nu) \\ x'(\nu) \\ y'(\nu) \\ z(\nu) \\ z'(\nu) \end{bmatrix} = \begin{bmatrix} r_1(\nu) \\ r_2(\nu) \\ r_3(\nu) \\ r_4(\nu) \\ r_5(\nu) \\ r_6(\nu) \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix} \equiv R(\nu)D \tag{10}$$

Here, each row of the $R(\nu)$ matrix is composed of the coefficients on the integration constants.

Given an initial relative state $X(\nu_0)$, the future trajectory is determined by first computing the integration constant vector D, then computing the R matrix across a range of future true anomaly values.

$$\begin{aligned} D &= R(\nu_0)^{-1} X(\nu_0) \\ X(\nu) &= R(\nu) D \end{aligned} \tag{11}$$

IV. Geometric Parameters for Periodic Relative Trajectories

Accurate solutions to the relative motion enable us to identify families of relative trajectories suitable for several different types of formation flying missions. In general, any formation is defined as the combination of individual trajectories. From a guidance standpoint, we seek to define naturally repeating trajectories so that (in the absence of noise, disturbances and modeling errors) the relative motion is maintained with zero control effort. In other words, the relative motion is "T-periodic", repeating once each orbit period.

In particular, we wish to define the target trajectory in terms of its geometric properties, to facilitate a more direct mapping to the higher-level formation objectives. Two different methods have been developed for parameterizing relative trajectories as a function of geometric components. The first method applies to circular orbits, the second to eccentric orbits. In both cases, the in-plane (x-y) and out-of-plane (z) motion are decoupled.

A. Relative Motion Geometry for Circular Orbits

In the case of circular orbits, the geometry of the relative motion is easily expressed as a superposition of along-track offset, in-plane elliptical motion, and cross-track oscillation. The following five parameters are used to fully define the geometry of any type of relative trajectory:

Table 1. Geometric Parameters for Relative Motion in Circular Orbits

Parameter	Description
y_0	Along-track offset. Defines the center of the in-plane relative ellipse.
$a_{_E}$	Semi-major axis of relative ellipse.
β_0	Phase angle on relative ellipse at ascending equator crossing. Measured positively from $-x$ axis to $+y$ axis of Hill's frame.
z_i	Cross-track amplitude due to inclination difference.
z_Ω	Cross-track amplitude due to right ascension difference.

An example trajectory is shown in Figure 4, illustrating the separate in-plane and out-of-plane motions.

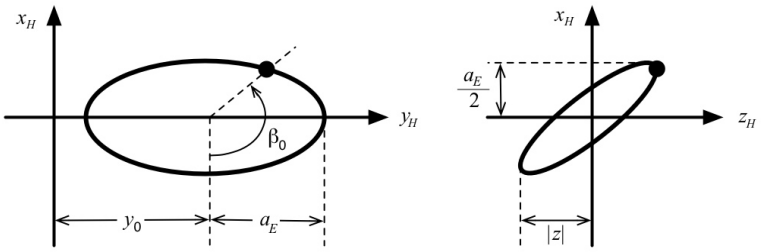

Figure 4. Example of Relative Motion in Circular Orbits

The relative ellipse is sometimes referred to as a "football orbit". It is a 2×1 elliptical shape in the x-y plane that is achieved by a small eccentricity difference. The phase angle β_0 defines the location of the satellite on this ellipse at the moment when the absolute orbit crosses the equator. The center of this ellipse is located at the along-track offset, y_0. When the semi-major axis $a_{_E}$ is zero, the elliptical motion vanishes, and the satellite remains fixed at y_0. This is often referred to as a leader-follower configuration. The amplitude of the cross-track oscillation is defined as:

$$|z| = \sqrt{{z_i}^2 + {z_\Omega}^2} \tag{12}$$

These parameters are chosen for the cross-track motion because the presence of inclination difference can result in significant secular drift from the J_2 perturbation. Thus, it is often desirable to achieve the desired cross-track amplitude with as little inclination difference as possible, so that $z_\Omega > z_i$.

The transformation from these geometric parameters to the Hill's frame position and velocity is given as follows:

$$x = -\frac{1}{2} a_{_E} \cos(\alpha_0 + \theta) \tag{13}$$

$$y = y_0 + a_{_E} \sin(\alpha_0 + \theta) \tag{14}$$

$$z = z_i \sin(\theta) - z_\Omega \cos(\theta) \tag{15}$$

$$\dot{x} = \frac{1}{2} a_{_E} n \sin(\alpha_0 + \theta) \tag{16}$$

$$\dot{y} = a_{_E} n \cos(\alpha_0 + \theta) \tag{17}$$

$$\dot{z} = n \left(z_i \cos(\theta) + z_\Omega \sin(\theta) \right) \tag{18}$$

where θ is the true latitude, and the angle α_0 is related to the original phase angle β_0 through the equation:[7]

$$\alpha_0 = \sin^{-1}\left(\frac{\sin\beta_0}{\sqrt{4-3\sin^2\beta_0}}\right) \tag{19}$$

The inverse equations, which convert from Hill's coordinates to geometric parameters, are given as:

$$y_0 = y - 2\frac{\dot{x}}{n} \tag{20}$$

$$a_{_E} = \sqrt{(y-y_0)^2 + 4x^2} \tag{21}$$

$$\alpha_0 = \tan^{-1}\left(\frac{2\dot{x}}{\dot{y}}\right) - \theta \tag{22}$$

$$\beta_0 = \sin^{-1}\left(\frac{2\sin\alpha_0}{\sqrt{1+3\sin^2\alpha_0}}\right) \tag{23}$$

$$z_\Omega = \frac{\dot{z}}{n}\sin(\theta) - z\cos(\theta) \tag{24}$$

$$z_i = \frac{\dot{z}}{n}\cos(\theta) + z\sin(\theta) \tag{25}$$

Referring back to Eq. 3 on page 4, we can identify the constraint for periodic motion. It is equivalent to setting the only coefficient on t equal to zero, to ensure that the along-track separation does not grow unbounded with time.

$$6nx_0 + 3\dot{y}_0 = 0 \tag{26}$$

Combining Equations 13 and 17 into Eq. 26, we find that the constraint is satisfied.

These transformation equations are used to define a dual-plane projected-circular formation in Section B on page 17.

B. Relative Motion Geometry for Eccentric Orbits

The relative motion occurring in eccentric orbits is fundamentally different from that found in circular orbits. The in-plane motion no longer follows a 2×1 ellipse, and the cross-track oscillation is not necessarily centered about the origin. The shape of the trajectory depends upon the eccentricity, as well as the points in the orbit where the maximum radial and cross-track amplitudes occur.

As discussed in Section III on page 4, the method of predicting the relative motion in eccentric orbits first requires that one computes a set of integration constants. Eq. 11 on page 5 shows how to compute D from an initial relative state. This method of fixing the position and velocity at a particular true anomaly is the most straightforward way to compute D. However, one may elect to introduce other types of constraints. The geometric parameters in Table 2 provide enough information to define a new set of constraints. These constraints can be used to compute integration constants that correspond to a desired trajectory.

Table 2. Geometric Parameters for Relative Motion in Eccentric Orbits

Parameter	Description
y_0	Center of along-track motion
$\bar{x}$	Maximum radial amplitude
$\nu_{\bar{x}}$	True anomaly where $\bar{x}$ occurs
$\bar{z}$	Maximum cross-track amplitude
$\nu_{\bar{z}}$	True anomaly where $\bar{z}$ occurs

A total of six constraint equations are required. The five parameters account for five of the equations. The remaining equation is obtained from the periodicity (zero-drift) constraint. Enforcing zero-drift in an

eccentric orbit is achieved by meeting the following constraint:[4]

$$y'(\nu)/x(\nu) = -\frac{1+e}{2+e} \tag{27}$$

Let us examine the constraints imposed by each parameter. First, $\bar{x}$ is defined as the maximum radial amplitude, occurring at a true anomaly of ν_x. Together these bring two constraint equations – at ν_x, the x position must equal $\bar{x}$ and the x velocity must equal 0. A similar set of constraints are obtained from $\bar{z}$ and ν_z. The y_0 parameter is defined as the center of motion in the y (along-track) direction. Thus, it can be expressed as follows:

$$y_0 = \frac{1}{2}\Big(y(\nu_{y-}) + y(\nu_{y+})\Big) \tag{28}$$

where ν_{y-} and ν_{y+} are the true anomaly values corresponding to the minimum and maximum values of y, respectively. These angles are not known a priori, so they must be determined by some method. The resulting set of constraints may be expressed as:

$$\begin{bmatrix} \bar{x} \\ y_0 \\ 0 \\ 0 \\ \bar{z} \\ 0 \end{bmatrix} = \begin{bmatrix} r_1(\nu_{\bar{x}}) \\ \frac{1}{2}\left(r_2(\nu_{y-}) + r_2(\nu_{y+})\right) \\ r_3(\nu_{\bar{x}}) \\ -(2+e)/(1+e)r_1(0) - r_4(0) \\ r_5(\nu_{\bar{z}}) \\ r_6(\nu_{\bar{z}}) \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix} \tag{29}$$

This approach is considered the "bounding" method for trajectory definition, because the x and z motions are bounded by the chosen parameters.

A double iterative procedure is used to compute ν_{y-} and ν_{y+}. First, initial guesses for ν_{y-} and ν_{y+} are made. The initial guesses are chosen to be $\nu_{\bar{x}} \pm \pi/2$. These values are then used to define the constraint equation for y_0. All other constraint equations are already fully defined, which allows the D vector to be computed. This represents the first level of iteration. Since D defines the entire trajectory, the actual minimum and maximum y values may be computed, along with the true anomalies at which they occur. This computation, which is the second level of iteration, is performed using the Newton-Raphson method with the first and second derivatives of y with respect to ν. If the extreme true anomalies are not close enough to the previous guess, then ν_{y-} and ν_{y+} are updated and the iteration continues. The approach has shown to work accurately and extremely fast in practice, typically requiring only 1-5 iterations and taking only a few tenths of a second.

If desired, a slightly different set of geometric parameters may be used. Rather than choosing the size and location of the maximum radial amplitude, one may instead choose a particular $[x, y]$ location in the relative frame, and the true anomaly where that position is to occur. Similarly, one may elect to define a particular z value, along with the true anomaly at which it occurs, instead of defining the value of the cross-track amplitude. The constraint equations associated with this alternate "positioning" method are given below:

$$\begin{bmatrix} x \\ y_0 \\ y \\ 0 \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} r_1(\nu_{xy}) \\ \frac{1}{2}\left(r_2(\nu_{y-}) + r_2(\nu_{y+})\right) \\ r_2(\nu_{xy}) \\ -(2+e)/(1+e)r_1(0) - r_4(0) \\ r_5(\nu_z) \\ r_6(\nu_{\bar{z}}) \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{bmatrix} \tag{30}$$

These transformation equations are used to define a repeating tetrahedron formation in Section C on page 19.

V. Maneuver Planning Methods for Relative Orbit Control

Formation flying maneuvers can be grouped into two different categories: *reconfiguration* and *maintenance*. In a reconfiguration, the desired trajectory of the spacecraft is changed, so that the spacecraft

maneuvers from its original trajectory to the new one. These are typically "large-scale" maneuvers, and can potentially require a significant amount of delta-v. Maintenance maneuvers, on the other hand, typically involve small corrections to eliminate small errors in the relative state. Here, the spacecraft returns to its original trajectory after disturbances have caused it to drift away.

The control objective of each type of maneuver is the same. We wish to proceed from an initial state to a target state while expending the least amount of control effort possible. In addition, we may seek to minimize the maneuver duration. A maneuver planning method has been developed for both circular and eccentric orbits which provides a time-weighted, fuel-optimal control solution to this problem.

It is worth noting that this control approach involves the intermittent application of maneuvers, with no pre-defined control frequency. It is well-suited for reconfiguration maneuvers and for coarse formation keeping. In general, it is sensible to apply this approach when the amount of time between maneuvers is much greater than the average maneuver duration. It is not intended, therefore, for precision formation control, where continuous corrections are required to maintain tight position tolerances. Under these circumstances, an active control method, such as LQG, should be used.[8]

This section describes the different maneuver planning methods for circular and eccentric orbits. In the case of circular orbits, two separate control laws are available. The first involves the analytic solution of a delta-v sequence to eliminate the error in orbital element differences. The second applies a linear program to the discretized orbit dynamics to provide the optimal control solution. For eccentric orbits, the linear programming method is applied once again, but with respect to the LTV dynamics of this regime.

A. Analytic Solution for Circular Orbits

This control law is based upon an analytic solution originally developed by Alfriend,[9] providing an impulsive delta-v sequence to correct errors in the orbital elements. The original solution is derived from Gauss' variational equations, providing a sequence of three in-plane delta-v's, spaced at intervals of a half-orbit period, followed by a single out-of-plane delta-v. The algorithm was further modified[7, 10] to compensate for cross-track coupling and to enable multi-orbit maneuver durations for the in-plane correction. The solution provides the minimum delta-v control over the time window considered.

The orbital element set used in the control law is defined as:

$$\vec{e} = \begin{bmatrix} a & \theta & i & q_1 & q_2 & \Omega \end{bmatrix}^\top \tag{31}$$

where a is the semi-major axis, θ is the true latitude, i is the inclination and Ω is the longitude of the ascending node. The parameters q_1 and q_2 are defined as:

$$\begin{aligned} q_1 &= e\cos(\omega) \\ q_2 &= e\sin(\omega) \end{aligned} \tag{32}$$

where e is the eccentricity and ω is the argument of perigee.

Let "Δ" denote the error in each element. The out-of-plane, or orbit-normal, delta-v is given by the equation:

$$\Delta v_n = \frac{h}{r}\sqrt{(\Delta i)^2 + (\Delta\Omega\sin i)^2} \tag{33}$$

where h is the angular momentum and r is the magnitude of the position vector. For circular orbits, the h/r term reduces to v, the orbit velocity. The delta-v must be applied at the true latitude of:

$$\theta_n = \tan^{-1}\left(\frac{\Delta\Omega\sin i}{\Delta i}\right) \tag{34}$$

The out-of-plane burn adjusts the amplitude of the cross-track oscillation by changing the inclination difference and/or right ascension difference. It can be seen from Gauss' variational equations, however, that this out-of-plane burn will also affect the elements q_1, q_2 and θ if it is applied anywhere *other than* the equator crossing ($\theta_n = 0, \pi$). Referring to Eq. 34, this is equivalent to a non-zero change in right ascension. To compensate, these cross-coupling effects are added to the initial measured errors (denoted with "0" subscripts) before computing the in-plane burns, as follows:

$$\Delta q_1 = \Delta q_{1,0} - \frac{r}{p} q_2 \sin\theta_n \cot i \Delta v_n \tag{35}$$

$$\Delta q_2 = \Delta q_{2,0} + \frac{r}{p} q_1 \sin\theta_n \cot i \Delta v_n \quad (36)$$

$$\Delta\theta = \Delta\theta_0 + \frac{r}{h} q_2 \sin\theta_n \cot i \Delta v_n \quad (37)$$

where p is the semi-latus rectum, equivalent to r and a for circular orbits.

The in-plane delta-v sequence can now be computed according to the following equations:

$$\Delta v_1 = \frac{na}{3N\pi}\left[\Delta\theta - \frac{3}{2}\Delta\bar{a}\,(\theta_1 - \theta_0) - 2\Delta\tilde{q}_0\right] + \frac{na}{4}\left[\left(\frac{M}{N}+1\right)\Delta q - \left(\frac{M}{N}-1\right)\Delta\bar{a}\right] \quad (38)$$

$$\Delta v_2 = \frac{na}{4}\left(\Delta q - \Delta\bar{a}\right) \quad (39)$$

$$\Delta v_3 = -\frac{na}{3N\pi}\left[\Delta\theta - \frac{3}{2}\Delta\bar{a}\,(\theta_1 - \theta_0) - 2\Delta\tilde{q}_0\right] - \frac{na}{4}\left[\frac{M}{N}\Delta q - \frac{M}{N}\Delta\bar{a}\right] \quad (40)$$

where n is the mean orbit rate, θ_0 is the current argument of latitude, and θ_1 is the argument of latitude at which the first burn is to be applied:

$$\theta_1 = \tan^{-1}\left(\frac{\Delta q_2}{\Delta q_1}\right) \quad (41)$$

Furthermore, we have:

$$\Delta q = \Delta q_1 \cos(\theta_1) + \Delta q_2 \sin(\theta_1) \quad (42)$$

$$\Delta\tilde{q}_0 = \Delta q_1 \sin(\theta_0) - \Delta q_2 \cos(\theta_0) \quad (43)$$

$$\Delta\bar{a} = \frac{\Delta a}{a} \quad (44)$$

The parameters M and N are maneuver time variables. Each is a positive integer, with M representing the number of half-orbits between the 1^{st} and 2^{nd} burn, and N the number of half-orbits between the 1^{st} and 3^{rd} burn. They are subject to the following constraints:

$$N \geq M + 1 \quad (45)$$

$$N \in Even \quad (46)$$

$$M \in Odd \quad (47)$$

A time window is supplied by the operator, defining the minimum and maximum bounds on the maneuver duration. The M and N parameters are varied within this specified time window to find the combination which requires the smallest delta-v. As an example, consider the following reconfiguration scenario. Two spacecraft are initialized at 550 km altitude and 34.5 degree inclination. The initial relative motion is an in-plane leader-follower with 300 meters separation, and we desire a centered in-plane elliptical trajectory with a 60 m semi-major axis for the relative ellipse.

The in-plane delta-v sequence is computed using Eqs. 38 to 40. In one case, we allow the maneuver to last up to 3 orbits. This requires that $N \leq 6$. In the second case, we allow the maneuver to take only one orbit, so that $N = 2$. The relative trajectories of both cases are shown in Figure 5 on the following page. The three red dots in each plot indicate the points at which each burn is applied. The delta-v for the multi-orbit case is 16 mm/s, compared with 43 mm/s for the single orbit case. In the multi-orbit case, the along-track position change is less costly because the spacecraft is allowed to drift with a smaller relative velocity over a longer period of time.

B. Linear Programming Method for Circular Orbits

The benefit of the analytic method is that it provides a minimum delta-v control solution and requires very little computational effort. It is limited, however, in that the maneuver must last a whole number of orbit periods. To provide greater flexibility over the choice of maneuver duration, a linear programming (LP) method is used.

The application of LP methods to the problem of relative orbit control has been presented in various forms in recent years.[11–14] In general, the LP approach is used to compute an impulsive delta-v sequence over a fixed time window, such that the desired relative state is achieved at the final time. Any time window may be chosen, providing greater flexibility than the analytic method.

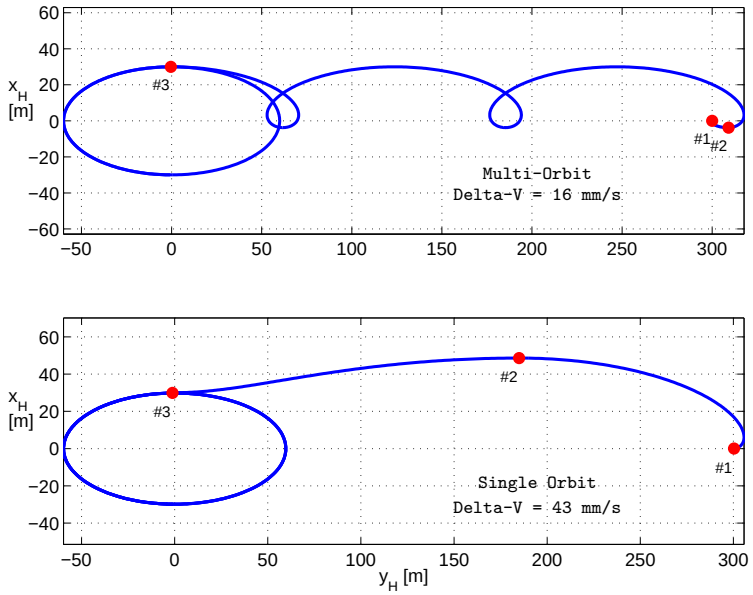

Figure 5. In-Plane Trajectories of Single and Multi-Orbit Reconfigurations

The application of LP control requires that the relative dynamics be expressed as a linear system in state-space. The relative dynamics for a circular reference orbit are linear time-invariant (LTI), and may be expressed as:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) \end{aligned} \tag{48}$$

where x is the state vector, u is the control input, and y is the output. The state vector consists of the relative position and velocity in Hill's frame, $\boldsymbol{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^\top$, the control input is the applied acceleration in Hill's frame, and the output is equal to the state vector (C is identity).

The A and B matrices are taken directly from the linearized equations of motion. For a circular reference orbit, the matrices are independent of time:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{49}$$

where n is the mean orbit rate. The continuous-time system is discretized using a zero-order hold over a time-step Δt to obtain:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + B\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \boldsymbol{x}_k \end{aligned} \tag{50}$$

where A and B now denote the discrete-time state-space matrices. The expression for the second state can be found as follows:

$$\begin{aligned} \boldsymbol{x}_2 &= A\boldsymbol{x}_1 + B\boldsymbol{u}_1 \\ &= A\left(A\boldsymbol{x}_0 + B\boldsymbol{u}_0\right) + B\boldsymbol{u}_1 \\ &= A^2\boldsymbol{x}_0 + AB\boldsymbol{u}_0 + B\boldsymbol{u}_1 \end{aligned} \tag{51}$$

Extending to the N^{th} state, we have:

$$\boldsymbol{x}_N = A^N \boldsymbol{x}_0 + \left[\begin{array}{cccc} A^{N-1}B & A^{N-2}B & \cdots & AB \end{array} \right] \times \left[\begin{array}{c} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{array} \right] \tag{52}$$

For future convenience, let us introduce the following definitions:

$$\begin{aligned} \acute{B} &= \left[\begin{array}{cccc} A^{N-1}B & A^{N-2}B & \cdots & AB \end{array} \right] \\ \acute{\boldsymbol{u}} &= \left[\begin{array}{cccc} \boldsymbol{u}_0 & \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_{N-1} \end{array} \right]^\top \end{aligned} \tag{53}$$

so that

$$\boldsymbol{x}_N = A^N \boldsymbol{x}_0 + \acute{B}\acute{\boldsymbol{u}} \tag{54}$$

This gives us an expression for the N^{th} state in terms of the initial state, $\boldsymbol{x}_0$, and the control history, $\boldsymbol{u}_k$, for $k = 0 \rightarrow N-1$. The objective is to find a control history that requires the minimum cumulative delta-v, subject to the constraint that the desired state is achieved at $k = N$.

The terminal constraint may be written as:

$$|\boldsymbol{x}_N - \boldsymbol{x}^*| \leq \boldsymbol{\epsilon} \tag{55}$$

for a sufficiently small $\boldsymbol{\epsilon}$ vector. Noting that the right-hand side is an absolute value, we may rewrite the above expression as two inequalities.

$$\begin{aligned} \boldsymbol{x}_N - \boldsymbol{x}^* &\geq -\boldsymbol{\epsilon} \\ \boldsymbol{x}_N - \boldsymbol{x}^* &\leq +\boldsymbol{\epsilon} \end{aligned} \tag{56}$$

Substituting Eq. 54 into Eq. 56, we obtain:

$$\begin{aligned} A^N \boldsymbol{x}_0 + \acute{B}\acute{\boldsymbol{u}} - \boldsymbol{x}^* \geq -\boldsymbol{\epsilon} \\ A^N \boldsymbol{x}_0 + \acute{B}\acute{\boldsymbol{u}} - \boldsymbol{x}^* \leq +\boldsymbol{\epsilon} \end{aligned} \tag{57}$$

Further algebraic manipulation yields:

$$\begin{aligned} -\acute{B}\acute{\boldsymbol{u}} &\leq \boldsymbol{\epsilon} + A^N \boldsymbol{x}_0 - \boldsymbol{x}^* \\ \acute{B}\acute{\boldsymbol{u}} &\leq \boldsymbol{\epsilon} - A^N \boldsymbol{x}_0 - \boldsymbol{x}^* \end{aligned} \tag{58}$$

Now let $\tilde{A}$ and $\tilde{\boldsymbol{b}}$ be defined as:

$$\begin{aligned} \tilde{A} &= \left[\begin{array}{c} -\acute{B} \\ \acute{B} \end{array} \right] \\ \tilde{\boldsymbol{b}} &= \left[\begin{array}{c} \boldsymbol{\epsilon} + A^N \boldsymbol{x}_0 - \boldsymbol{x}^* \\ \boldsymbol{\epsilon} - A^N \boldsymbol{x}_0 + \boldsymbol{x}^* \end{array} \right] \end{aligned} \tag{59}$$

so that the inequality may be written as follows:

$$\tilde{A}\acute{\boldsymbol{u}} \leq \tilde{\boldsymbol{b}} \tag{60}$$

The problem is now posed in a form suitable for the Simplex algorithm, which is a well-known technique for solving LP problems. The objective is to minimize the cost $\tilde{\boldsymbol{c}}\acute{\boldsymbol{u}}$ subject to the constraint defined in Eq. 60. Here, the cost coefficient vector, $\tilde{\boldsymbol{c}}$, has $3N$ columns, and is nominally composed of all 1's. The weights may be adjusted to provide a greater or lesser penalty at different times, or for different axes in the relative frame.

Consider an in-plane reconfiguration maneuver, in which we transition from a leader-follower configuration to a non-centered elliptical path. The initial state has a constant +1 km along-track offset; the desired

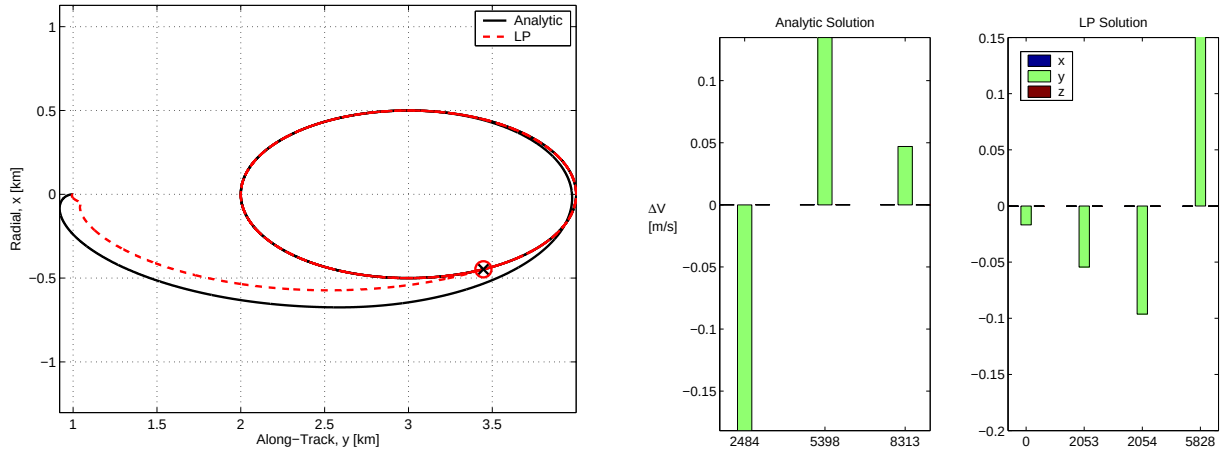

Figure 6. Comparison of In-Plane Trajectories and Delta-V Sequences

trajectory has a 1 km semi-major axis for its relative ellipse, and the ellipse center is offset by +3 km. In one case, we use the analytic solution method to compute the delta-v sequence for reconfiguration. In the second case, we use the LP method with a 1 second time-step. The delta-v sequence for both methods is shown in Figure 6. The maneuver duration is limited to 1 orbit.

In each case, control is applied only in the y-direction (along-track). The total delta-v resulting from the analytic method is 0.36351 m/s, while the total for the LP method is just 0.33495 m/s – nearly 8% lower. The important difference between the two is that the LP solution begins at time 0, while the analytic case begins at 2,484 seconds – half of an orbit period. This delay is due to the fact that, in the analytic method, the first burn must occur at a particular time in the future, which is a pure function of the relative state error. The plot of the in-plane trajectory shows that the desired motion is achieved in both cases. The red circle and black "x" plotted on the lower-right portion of the ellipse indicate the final position for each case after simulating for 4 orbits. The position difference is less than 16 cm.

C. Linear Programming Method for Eccentric Orbits

The relative dynamics for an eccentric reference orbit are time-varying. Thus, the continuous-time state-space representation of the dynamics is now:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= C(t)\boldsymbol{x}(t) \end{aligned} \tag{61}$$

Alternatively, we may use the true anomaly, ν, as the independent variable instead of time. The input matrix B, is the same as that defined in Eq. 49 on page 11. The A matrix, however, now becomes a function of true anomaly.

$$A(\nu) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \dot{\nu}^2 + 2h & \ddot{\nu}^2 & 0 & 0 & 2\dot{\nu} & 0 \\ -\ddot{\nu}^2 & \dot{\nu}^2 - h & 0 & -2\dot{\nu} & 0 & 0 \\ 0 & 0 & -h & 0 & 0 & 0 \end{bmatrix} \tag{62}$$

where h is defined as:

$$h = n^2 \left(\frac{1 + e\cos\nu}{1 - e^2} \right)^3 \tag{63}$$

Once again, we wish to transition from continuous-time to discrete-time. Because the system is now LTV, it becomes necessary to perform the zero-order hold operation at every time-step. Each time, a unique set of discrete A and B matrices are obtained. Although the continuous-time B matrix is not a function of true

anomaly, the discrete version of the matrix changes over time because the length of the time-step Δt varies. This is because we choose to discretize the orbit with an even spacing of true anomaly. The equations for the first two states are given below:

$$\begin{aligned} \boldsymbol{x}_1 &= A_0\boldsymbol{x}_0 + B_0\boldsymbol{u}_0 && (64)\\ \boldsymbol{x}_2 &= A_1\boldsymbol{x}_1 + B_1\boldsymbol{u}_1 && (65)\\ &= A_1A_0\boldsymbol{x}_0 + A_1B_0\boldsymbol{u}_0 + B_1\boldsymbol{u}_1 && (66)\\ & && (67) \end{aligned}$$

Extending to the N_{th} state, we find:

$$\begin{aligned} \boldsymbol{x}_N &= \left[\begin{array}{cccc} A_{N-1} & A_{N-2} & \cdots & A_0 \end{array}\right]\boldsymbol{x}_0 && (68)\\ &+ \left[\begin{array}{cccc} A_{N-1} & A_{N-2} & \cdots & A_1 \end{array}\right]B_0\boldsymbol{u}_0 && (69)\\ &+ \left[\begin{array}{cccc} A_{N-1} & A_{N-2} & \cdots & A_2 \end{array}\right]B_1\boldsymbol{u}_1 && (70)\\ &+ \vdots && (71)\\ &+ B_{N-1}\boldsymbol{u}_{N-1} \end{aligned}$$

This can be simplified to a form similar to Eq. 52 on page 12. We must first define the following two matrices:

$$\begin{aligned} \acute{A} &= \left[\begin{array}{cccc} A_{N-1} & A_{N-2} & \cdots & A_0 \end{array}\right] && (72)\\ \acute{B} &= \left[\begin{array}{cccc} \left[\begin{array}{cccc} A_{N-1} & A_{N-2} & \cdots & A_1 \end{array}\right]B_0 & \left[\begin{array}{cccc} A_{N-1} & A_{N-2} & \cdots & A_2 \end{array}\right]B_1 & \cdots & B_{N-1} \end{array}\right] \end{aligned}$$

We can now express the N^{th} state as follows:

$$\boldsymbol{x}_N = \acute{A}\boldsymbol{x}_0 + \acute{B}\acute{\boldsymbol{u}} \tag{73}$$

where the definition of $\acute{\boldsymbol{u}}$ is the same as before. Now, however, A^N is replaced by $\acute{A}$, and the definition of $\acute{B}$ has changed. Applying the same terminal constraint shown in Eq. 55 on page 12, and following the same steps that lead to Eq. 59 on page 12, we find that:

$$\begin{aligned} \tilde{A} &= \left[\begin{array}{c} -\acute{B} \\ \acute{B} \end{array}\right] && (74)\\ \tilde{\boldsymbol{b}} &= \left[\begin{array}{c} \boldsymbol{\epsilon} + \acute{A}\boldsymbol{x}_0 - \boldsymbol{x}^* \\ \boldsymbol{\epsilon} - \acute{A}\boldsymbol{x}_0 + \boldsymbol{x}^* \end{array}\right] \end{aligned}$$

As before, these matrices are used in the inequality, $\tilde{A}\acute{\boldsymbol{u}} \leq \tilde{\boldsymbol{b}}$, which may be solved by Simplex.

D. Time-Weighted Maneuver Selection

Whichever control method is used, we search for a solution that represents the combination of minimum-time *and* minimum-fuel objectives. First, the the control trajectory is computed several times, over a range of possible maneuver durations. The nominal cost for each case, c_i, is defined as the total delta-v required for the maneuver. This cost is then weighted by the maneuver duration, T_i, so as to penalize the longer maneuvers. An exponential weighting method is used in this system, where the weighted cost w_i is computed as:

$$w_i = c_i \times \left(\frac{T_i}{T_{min}}\right)^x \tag{75}$$

where T_{min} is the minimum maneuver duration considered, and $x \geq 0$ is an adjustable parameter. Setting $x = 0$ causes the duration to be ignored. In this case, only the longest possible duration would be considered, since it must correspond to the minimum delta-v. Increasing x causes greater importance to be placed on the maneuver duration.

VI. Distributed Guidance Law

Formations are designed by combining several relative trajectories in such a way that a desired geometric configuration is achieved. In order to realize a formation, each spacecraft must achieve one of the relative trajectories. It is typically not important which spacecraft achieves which trajectory, as long as all desired trajectories are met so that the overall formation geometry is realized. Therefore, the main function of the formation guidance law is that of trajectory assignment. The objective is to achieve all desired trajectories in such a way that minimizes the total cost.

The guidance law designed for the DFF system operates at the team level. The set of target states for a team may be defined by the operator according to three different levels of autonomy:

- **Low Autonomy** – The target state for each spacecraft is defined explicitly. It may be defined in Hill's frame coordinates, orbital element differences, or in terms of the geometric parameters. The control law immediately tracks the new desired trajectory.
- **Medium Autonomy** – A set of geometric goals is supplied for the entire team, where each goal set corresponds to a desired trajectory. The guidance law works to find the optimal configuration, assigning each target to a different team member such that the total cost is minimized.
- **High Autonomy** – A few high-level objectives are supplied, such as a pre-defined formation type, size, and orientation. The geometric goals for the team are computed online to meet the given objectives. Each target is then assigned to a different team member such that the total cost is minimized.

The geometric goals are described in Tables 1 and 2 for circular and eccentric orbits.

The guidance law is not used in the "Low Autonomy" case, as the target states are assigned directly from the operator. At the medium and high autonomy levels, however, the targets are assigned on-orbit. The general procedure is as follows:

1. The geometric goals for the team are either supplied to or computed by the team captain.
2. The captain distributes the team goals to all relative team members.
3. Each recipient uses the control law to estimate the cost to achieve all desired trajectories.
4. The vector of costs from each relative member is returned to the captain.
5. The captain assembles all cost vectors into a single cost matrix.
6. The captain applies an assignment algorithm to the cost matrix to find a solution that minimizes the total cost.
7. The captain sends out the newly assigned geometric goals to each relative team member.

The guidance law is therefore both distributed and centralized. The cost estimation is distributed across all satellites in the team, but the assignment task is performed centrally, by the captain.

The total cost is defined as a weighted sum of the total delta-v's for each spacecraft to achieve its target state. The individual delta-v's are weighted according to their remaining fuel percentage, in order to promote equal fuel usage throughout the cluster. The cost for the i^{th} satellite to reach the j^{th} target state is:

$$c_{ij} = f_i^{-x} \times \Delta V_{ij} \tag{76}$$

where f_i is the remaining fuel percentage of the i^{th} spacecraft, and $x > 0$ is an adjustable parameter indicating the importance of fuel equalization.

The general problem is to assign N target states to N satellites such that the total resulting cost is minimized. Two different approaches to solving the assignment problem have been implemented within the guidance law. The first approach, termed the "optimal method", involves searching over all possible combinations to find the one with the minimum total cost. The total number of unique combinations is $N!$ for a team of $N + 1$ members (the reference is not included). This approach is therefore computationally cumbersome as N becomes large, i.e., ≥ 8. The advantage is that a globally optimal solution is guaranteed.

The second approach is called the "privileged method". This technique requires considerably less computation, but does not guarantee that a globally optimum solution is found. It consists of the following steps:

1. Determine the minimum projected cost of each satellite.

2. Determine which satellite has the highest minimum cost.

3. Assign that satellite to the target state corresponding to its minimum cost.

4. Repeat steps 1-2 for all remaining members and remaining target states.

A similar technique is presented by Tillerson.[11] The optimal and privileged approaches are compared in Section A below.

Whichever method is used, the ground operator has the flexibility of restricting the target state distribution. Each target state may be restricted so that it is considered by only a specific subset of team members. This is accomplished in the cost estimation stage by forcing c_{ij} to be an extremely large number for cases where the j^{th} target state is not allowed to be assigned to the i_{th} satellite. The assignment algorithm naturally avoids these high cost combinations.

The guidance law has also been designed to take advantage of the additional freedom available in circular reference orbits. Here, we allow two types of target states to be identified: *fixed* and *variable*. With the fixed target state, we specify both the desired state and the point in the orbit at which it is to occur. With the variable target state, we consider all possible states along the trajectory. When multiple target states are defined along the same trajectory, the original phase separation is maintained as a constraint. The details of this "variable state" method require a lengthy discussion and are beyond the scope of this paper.

In eccentric orbits, only fixed target states are possible. This is due to the restricted nature of the relative trajectories in eccentric orbits, where each state along a given trajectory must occur at a particular true anomaly. In the cost estimation stage of the guidance law, the fixed target state is defined by using the true anomaly associated with the longest allowable maneuver duration.

It is also interesting to consider the task of assigning target states to a cluster which is composed of multiple teams. This requires inter-team cooperation, and transforming the relative states from each team to a common reference frame. First, the relative state of each spacecraft is computed with respect to the cluster reference. The target states are also defined with respect to the cluster reference. The usual method is then followed, with each satellite computing the set of costs to achieve all targets, and transmitting the cost vector to the cluster captain. The privileged assignment method is then used to quickly find a near-optimal configuration, and the targets are distributed accordingly throughout the teams. The final step is for each relative satellite to transform the target state to its team-based coordinate frame before reconfiguring to the new trajectory.

A. Example Assignment Problem

Consider a team with 8 relative spacecraft flying in a circular orbit at 550 km altitude. The team is initialized in a leader-follower formation with a common along-track spacing of 200 m. Each spacecraft also has an out-of-plane component so that they follow the same ground-track. We wish to reconfigure the formation so that two equally phased projected-circle geometries are achieved, one with a 1 km radius and the other with a 500 m radius. In both cases, the reference is located at the center.[a] The first step is to estimate the cost for each satellite to achieve all possible trajectories. In this case, all desired states lie on two projected circles. We therefore define the targets as two sets of four equally phased "variable states".

Each satellite uses the analytic control law to estimate the total delta-v required to reach the new trajectory. Using an angular resolution of 5 deg, the computation is performed for a total of 144 possible target states (72 for each trajectory). The final cost for each satellite is then weighted according to its remaining fuel percentage. In this example, the fuel weighting exponent is $x = 1$, and the i^{th} satellite has a remaining fuel percentage of:

$$f_i = 10 \times (1 + i)\% \tag{77}$$

The cost estimates for each satellite are then submitted to the captain, who performs the assignment. The results from both the privileged and optimal assignment method are compared in Table 3 on the next page. Table 3 on the following page summarizes the results of the comparison. The privileged assignment method finds a solution that is just 0.6% higher than the optimal result, and it provides a huge savings in computation time. The optimal method takes a considerable amount of time to run because it is comparing

[a]The projected-circle geometry is described in Section B on the following page

Table 3. Optimal vs. Privileged Assignment Methods: Cost and Computation Time

Method	Weighted Cost	Delta-V [m/s]	Computation Time [sec]*
Privileged	16.28	8.30	0.004
Optimal	16.22	8.16	301.8

* This example was run in MATLAB version 6.5.1 on a PowerMac G4 running OS X.

the cost associated with $8! = 40,320$ unique combinations. The worst possible configuration had a weighted cost of 21.3 m/s, which is about 30% higher than the optimal solution.

It is interesting to note the effect of weighting the costs. The four spacecraft with the lowest remaining fuel percentages (20, 30, 40 and 50%) are assigned to the smaller circle, while those with more fuel are assigned to the larger circle. This is due to the fact that the larger circle requires twice the out-of-plane component as the smaller, and the delta-v is extremely sensitive to out-of-plane changes.

The relative motion associated with this reconfiguration maneuver is shown in Figure 7. The initial and final locations are marked. The ellipses in the x-y plane show that the in-plane motion is achieved first. A single cross-track burn applied at either end of the ellipse then puts each spacecraft on the projected-circle.

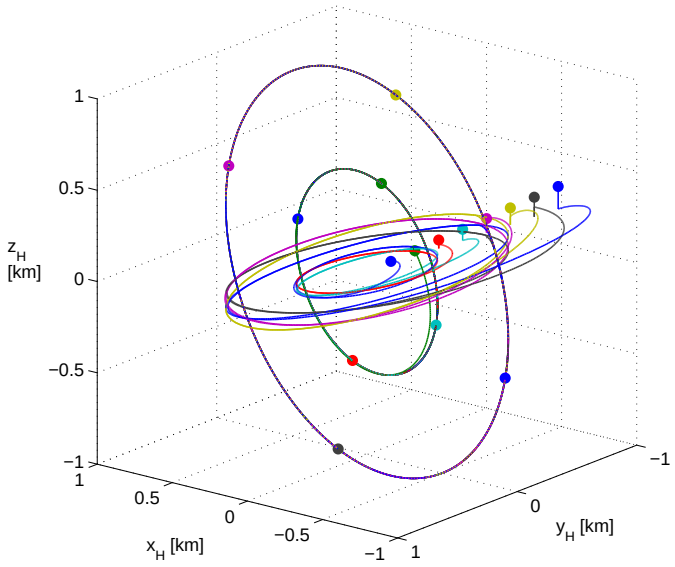

Figure 7. Relative Motion for Reconfiguration Maneuver

B. Defining a Projected-Circular Geometry

The "projected-circular" formation is of particular interest for low-earth orbit applications. Here, one or more spacecraft follow a specific trajectory about the reference such that the motion in the y-z Hill's frame follows a circular path. NASA has identified the sustained control of this formation as a LEO benchmark problem[15] for distributed space systems (DSS).

The reference orbit of the benchmark problem is sun-synchronous, with an altitude of 400 km and an inclination of 97.03 deg. The formation consists of six spacecraft, equally phased along two separate projected circles, each with a 500 m radius. The trajectories of the two projected circles lie in separate planes, which are canted at ± 26.565 deg about the y-axis. To reflect the two distinct planes of motion, this particular formation is termed a "dual-plane" projected circle. The formation is shown in Figure 8 on the following page. The circular projection onto the y-z plane is shown, as is the elliptical projection on the x-y plane.

A projected-circular trajectory can be easily defined using the geometric parameter set defined in Table 1 on page 6. Given the desired radius of the circle, R, and the phase location on the circle at the equator

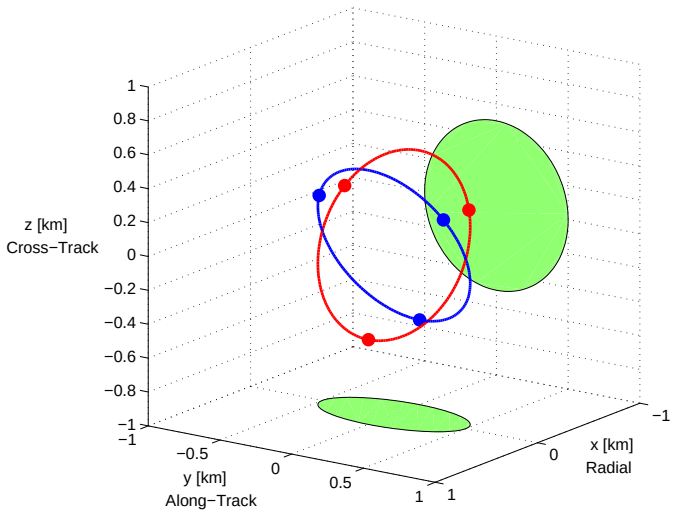

Figure 8. Dual-Plane Projected-Circular Formation

crossing, α_0, the geometric parameters are computed as follows:

$$\begin{aligned} y_0 &= 0 \\ a_{_E} &= R \\ \beta_0 &= \sin^{-1}\left(\frac{2\sin\alpha_0}{\sqrt{1+3\sin^2\alpha_0}}\right) \\ z_\Omega &= \pm R\cos(\alpha_0) \\ z_i &= \pm R\sin(\alpha_0) \end{aligned} \tag{78}$$

where α_0 is measured along the projected circle from the negative z-axis towards the positive y-axis. The equation for β_0 is simply the inverse of Eq. 19 on page 7. The $\pm$ indicates which plane the trajectory lies in. The blue trajectory in Figure 8 lies in the positive plane, while the red trajectory occupies the negative plane. Note that, for $\alpha_0 = 0$, the cross-track amplitude is associated with a pure right ascension difference, and zero inclination difference ($z_i = 0$). This corresponds to the maximum cross-track value occuring at the equator crossing.

The formation shown in Figure 8 is initialized by first defining the geometric parameters. We set $R = 0.5\ km$, and choose an array of six equally spaced phase angles. The sign of the plane alternates from positive to negative for each subsequent phase angle. Table 4 summarizes the parameters assigned to each satellite.

Table 4. Geometric Parameters for the Dual-Plane Projected Circle

Satellite #	Sign	y_0 [km]	$a_{_E}$ [km]	α_0 [deg]	β_0 [deg]	z_i [km]	z_Ω [km]
1	+	0	0.5	0	0	0	0.5
2	−	0	0.5	60	73.9	$-\sqrt{3}/4$	-0.25
3	+	0	0.5	120	106.1	$\sqrt{3}/4$	-0.25
4	−	0	0.5	180	180	0	0.5
5	+	0	0.5	-120	-106.1	$-\sqrt{3}/4$	-0.25
6	−	0	0.5	-60	-73.9	$\sqrt{3}/4$	-0.25

With the geometry of each trajectory defined, we now apply the transformation equations (Eqs. 13 to 18), which convert the geometric parameters to Hill's frame coordinates. A true latitude of $\theta = 0$ is arbitrarily

chosen. This gives us the desired relative state of each satellite with respect to the center of the projected circles. Let the state of the i^{th} satellite be called X_i^{cc}, where the cc superscript refers to the reference being the circle's center.

Since no satellite exists at the center of the projected circles, we must define one of the satellites that lies *on* a circle as the reference. Any satellite may be chosen; let us select the first satellite, in which $\alpha_0 = 0$. We now compute the relative state of all other satellites with respect to this one. Varying i from 2 to 6,

$$X_i = X_i^{cc} - X_1^{cc} \tag{79}$$

With the relative states now defined with respect to another satellite, the geometric parameters may be computed using Eqs. 20 through 25. The same value that was used for θ before must be used again. The trajectories of the other five satellites with respect to the new reference is shown in Figure 9. Although the geometry is more complicated when viewed in this reference frame, it still comprises the same relative motion as before. When viewed in the y-z frame, the relative satellites appear to "roll" about the reference in two opposite directions. The geometric parameters associated with each satellite's desired trajectory are

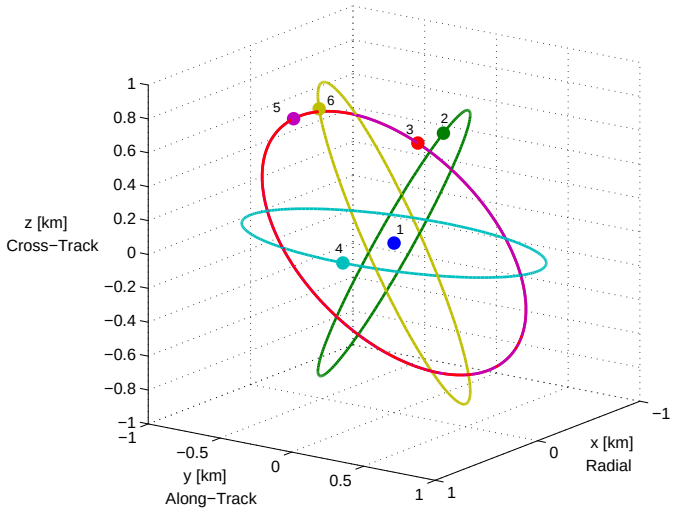

Figure 9. Dual-Plane Projected-Circular Formation with a Non-Centered Reference

given in Table 5. Note from Figure 9 that satellites #3 and #5 lie on the same trajectory, they are just

Table 5. Geometric Parameters for the Dual-Plane Projected Circle with a Non-Centered Reference

Satellite #	y_0 [km]	a_E [km]	α_0 [deg]	β_0 [deg]	z_i [km]	z_Ω [km]
2	0	0.5	120	106.1	$-\sqrt{3}/4$	-0.75
3	0	$\sqrt{3}/2$	150	130.9	$\sqrt{3}/4$	-0.75
4	0	1	180	180	0	0
5	0	$\sqrt{3}/2$	-150	-130.9	$-\sqrt{3}/4$	-0.75
6	0	0.5	-120	-106.1	$\sqrt{3}/4$	-0.75

separated by a phase offset. The three remaining satellites occupy completely different trajectories. Thus, there are a total of four unique trajectories. The set of team goals is then comprised of four unique variable states, plus one duplicated variable state.

C. Defining a Tetrahedron Geometry

NASA has identified a benchmark problem for highly elliptic orbits in which four spacecraft repeatedly achieve a regular tetrahedron formation at apogee.[15] The reference follows a 1.2 by 18 Earth radii orbit,

corresponding to an eccentricity of 0.875 and a semi-major axis of about 61.2 km. At apogee, the four spacecraft occupy the four points of the tetrahedron, with a common separation of 10 km between all spacecraft. The orientation of the tetrahedron is arbitrary. An additional objective is that the spacecraft avoid close approaches (less than 1 km) throughout the orbit.

The initialization of the target trajectories for this formation can be achieved using the "positioning" method outlined in Eq. 30 on page 8. One spacecraft serves as the reference, leaving three trajectories to be defined. For each of the three relative spacecraft, we specify the desired position in the x, y, and z axes at a true anomaly of $\nu = 180$ deg. Table 6 summarizes one possible set of geometric parameters for each trajectory. In this case, we specify only one trajectory to have an out-of-plane component. The relative

Table 6. Geometric Parameters for the Eccentric Tetrahedron Formation

Trajectory	y_0 [km]	x [km]	y [km]	ν_{xy} [deg]	$\bar{z}$ [km]	$\nu_{\bar{z}}$ [deg]
A	0	0	10	180	0	180
B	0	$5\sqrt{3}$	5	180	0	180
C	0	$5/\sqrt{3}$	5	180	$10\sqrt{2/3}$	180

trajectories associated with these geometric parameters are shown in Figure 10. The plots verify that the expected tetrahedron geometry occurs at apogee. The closest distance between any two spacecraft over the course of the orbit is 3.7 km. These trajectories represent just one orientation – an infinite number of orientations are possible. One may rotate the $[x, y, z]$ vector of each non-reference point in the tetrahedron through a common set of euler angles to achieve a different orientation.

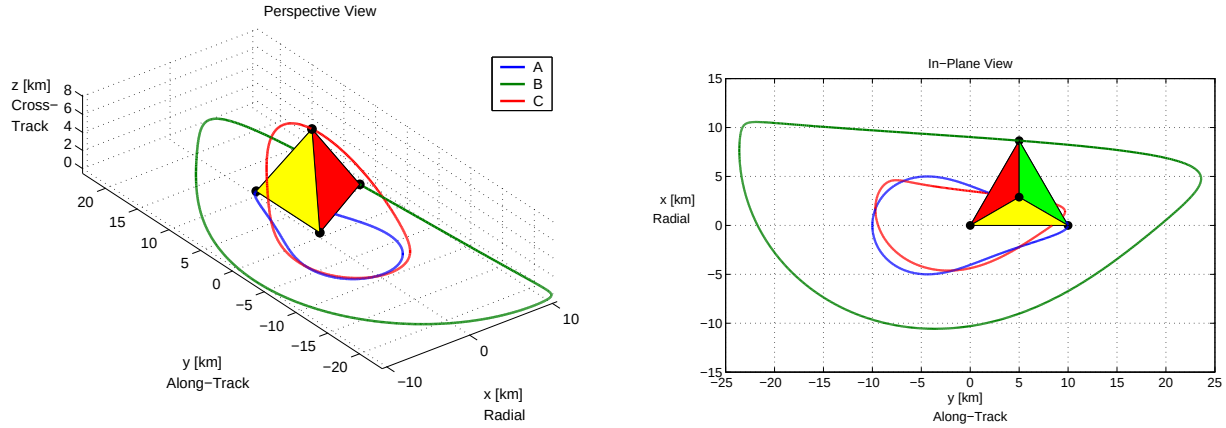

Figure 10. Trajectories for the Eccentric Tetrahedron Geometry

VII. Conclusions

This paper has presented a new concept for decentralized guidance and control in formation flying missions. The hierarchical, multiple-team framework represents a compromise between wholly centralized or decentralized designs. At the team level, a distributed guidance law is implemented in which cost estimates for potential target states are computed in a decentralized manner, and the assignment of those targets is undertaken centrally by a team captain. The cost is weighted separately for each spacecraft by its remaining fuel percentage, promoting fuel equalization. The target states commanded to the guidance law lie on period relative trajectories, which are defined in a geometric sense to give the operator a flexible, intuitive system for designing formations. The trajectories assigned to various team members are controlled in a decentralized fashion, with each non-reference spacecraft planning its own optimal, impulsive burn sequence. Further, the cost function minimized by the control law is a time-weighted delta-v, enabling the combined objectives of a minimum-fuel and minimum-time maneuver.

The current system is capable of providing coarse control for formations of arbitrary size in both circular and eccentric orbits. This control approach is model-predictive, in which maneuvers are planned so that a target state is reached at a certain future time. While this method is suitable and in fact preferable for formation acquisition and general reconfiguration maneuvers, it is not suitable for precise formation keeping. It is therefore desirable to add the capability for active feedback control, so that tight position control may be attained for sustained periods of time. However, the manner in which this level of control is implemented depends greatly on the spacecraft design, including sensor and actuator performance and limitations, as well as mission constraints. The challenge is to develop a generic control framework that accounts for a wide range of possible objectives and constraints.

Future research will focus on the unique issues associated with a multiple-team framework. In particular, strategies for effective collision monitoring and avoidance must be developed. This becomes an increasingly challenging problem as the size of the cluster grows, and it also becomes difficult to identify a proper approach when the cluster is divided into separate teams. In addition, the tradeoffs associated with different team organizations should be considered and quantified. This includes the effect on communication, computation load, and fuel usage.

Acknowledgments

This work was performed under Phase I and Phase II SBIR funding through the NASA Goddard Space Flight Center, contract numbers NAS5-03027 and NNG04CA08C.

References

[1]Speyer, J. L., "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem," *IEEE Trans. Automat. Contr.*, 2003.

[2]Clohessy, W. H. and Wiltshire, R. S., "Terminal Guidance System for Satellite Rendezvous," *Journal of Aerospace Science*, , No. 27, 1960, pp. 653–658.

[3]Lawden, D. F., *Optimal Trajectories for Space Navigation*, Butterworth, London, 1963.

[4]Inalhan, G., Tillerson, M., and How, J., "Relative Dynamics and Control of Spacecraft Formations in Eccentric Orbits," *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 1, 2002.

[5]Broucke, R. A., "Solution of the Elliptic Rendezvoud Problem with Time as the Independent Variable," *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 4, 2003, pp. 615–621.

[6]Vallado, D., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, Space Technology Series.

[7]Mueller., J. and Brito, M., "A Distributed Flight Software Design For Satellite Formation Flying Control," *Proceedings of the AIAA Space 2003 Conference*, No. AIAA 2003-6373, AIAA, Wahington, DC, 2003.

[8]Hamilton, N., Folta, D., and Carpenter, R., "Formation Flying Satellite Control Around the Sun-Earth L2 Libration Point," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, No. 2002-4528, AIAA, 2002.

[9]Alfriend, K., "TechSat 21 Orbit Maintenance Strategy," Tech. Rep. 2, Air Force Research Laboratory, August 2002.

[10]B., M. J., "TechSat 21 Algorithm Description Document: Optimal In-Plane Burn Sequence," Tech. rep., Princeton Satellite Systems, January 2003.

[11]Tillerson, M., Inalhan, G., and How, J., "Co-ordination and control of distributed spacecraft systems using convex optimization techniques," *International Journal of Robust Nonlinear Control*, Vol. 12, No. 1, 2002, pp. 207–242.

[12]Breger, L., Ferguson, P., and How, J., "Distributed Control of Formation Flying Spacecraft Built on OA," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Austin, TX, 2003.

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

[14]Tillerson, M. and How, J., "Formation Flying Control in Eccentric Orbits," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Montreal, Canada, 2001.

[15]Carpenter, J. R., Leitner, J., Folta, D., and Burns, R., "Benchmark Problems for Spacecraft Formation Flying Missions," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, No. AIAA 2003-5364, AIAA, Austin, TX, 2003.