AIAA/AAS Astrodynamics Specialist Conference and Exhibit
18 - 21 August 2008, Honolulu, Hawaii

AIAA 2008-6436

AIAA/AAS Astrodynamics Specialists Conference and Exhibit, August 18-21, Honolulu, Hawaii

Actuator Constrained Optimal Control of Formations Near the Libration Points

Stuart A. Stanton* and Belinda G. Marchand †

Dept of Aerospace Engineering and Engineering Mechanics,

The University of Texas at Austin, 1 University Station, C0600, Austin, TX, 78712-0235

Spacecraft applications like space-based interferometry have motivated investigations into the feasibility of spacecraft formations near the libration points of the Sun-Earth/Moon system. The sensitivity of the dynamical regime imposes extremely precise control requirements for non-natural formations. Unconstrained control laws tend to require acceleration magnitudes smaller than existing thruster technology can provide. In the present study, a methodology is presented that identifies actuator constrained solutions in the nonlinear model. The optimal solutions that ensue are attainable with existing technology and demonstrate the feasibility of spacecraft formationkeeping in the presence of actuator constraints in the Circular Restricted Three-Body Problem.

Nomenclature

$\mathcal{B}$	body-fixed frame
$\boldsymbol{c}$	constraint vector
$\boldsymbol{f}, \tilde{\boldsymbol{f}}$	dynamics function
J	cost function
G	gravitational constant
$\mathcal{I}$	inertial frame
m	mass
n	number of variables in $\boldsymbol{x}$
n_c	number of constraints
n_d	number of deputy spacecraft
n_g	number of gridpoints
n_k	number of knots
n_n	number of nodes
n_s	number of segments
n_u	number of controls
n_y	number of states
$\mathcal{R}$	rotating frame
$\boldsymbol{r}$	position vector
T	thrust acceleration magnitude
t	time
U	potential function
$\mathcal{U}$	control space
$\boldsymbol{u}$	control vector
$\boldsymbol{v}$	velocity vector
$\boldsymbol{x}$	parameter vector
$\boldsymbol{y}$	state vector
$\boldsymbol{\beta}$	path constraints
Δt	time interval
$\boldsymbol{\psi}$	point constraints
ω	rotation rate

Subscript

c	chief spacecraft
d	deputy spacecraft
i, ι	vector element counter
j	node counter
k, κ	knot or segment counter
l, λ	spacecraft counter
m	midpoint
s	star
$\odot$	Sun
$\oplus$	Earth/Moon barycenter

*Capt, USAF; Ph.D. Candidate, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712.

†Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712.

I. Introduction

Interest in space-based interferometry has motivated many investigations[1–16] into the feasibility of spacecraft formations near the libration points of the Sun-Earth/Moon system. The work to date can be characterized by the dynamic model used to describe the motion, the class of the control scheme presented, and the optimality of the controller's performance. Early investigations focus on impulsive unconstrained formationkeeping strategies in the Circular Restricted Three-Body Problem (CR3BP) based on target point and Floquet methods.[1] Continuous control techniques are later explored to track arbitrary reference trajectories.[2–7] Both linear (quadratic control with linearized dynamics) and nonlinear (Lyapunov-based) methods are considered in these investigations. These techniques are later extended to formations determined in the Ephemeris dynamical model.[8–14] Recent efforts employ numerical optimization techniques in the design of optimal spacecraft formations.[15,16]

The sensitivity of the dynamical regime near the libration points is apparent in the precise control requirements for non-natural formations derived in the above investigations. Some studies[17–22] focus on reducing the lower performance bounds on thruster technology. However, unconstrained control laws in this regime often require thrust levels *below* the limits of the technology presently available. The associated control accelerations, though small, cannot be neglected if precision formation keeping is desired.

Missions like TPF[23,24] and MAXIM[25] require precision tracking during data collection phases, often with the expectation that this be done in the absence of simultaneous actuation. Previous studies suggest this is an unrealistic expectation. At the same time, high precision tracking is achievable only through precise unconstrained control. The present study seeks to establish a systematic method for determining realistic expectations for libration point formations in the context of the sensitive dynamic environment and the current state of propulsive technology.

Constrained finite burn solutions, achievable with currently available technology, are the focus of this investigation. Thus, solutions are constrained to account for limitations in thruster technology. Mission requirements are also considered. Interferometry applications are of particular interest in the development of actuator constraints consistent with potential capabilities of member satellites, leading to significant spacecraft orientation restrictions. The restrictive—but realistic—assumptions applied here seek to establish reasonable expectations for tracking accuracy for libration point formations.

Direct optimization methods are employed as the basis of the current analysis. The optimal control problem is transcribed into a parameter optimization problem, and the solution is subsequently determined using existing Nonlinear Programming (NLP) algorithms, such as SNOPT.[26] The unique nature of this problem motivates the development of a novel approach to solving complex actuator constrained optimal control problems. An effective switching control scheme is presented that is critical to this development.

The formulation presented is applicable to generalized nonlinear systems. The approach is validated with examples based on the CR3BP.

CR3BP Equations of Motion and the Reference Orbit

In this study, the formation is centered around an artificial reference point, c, that evolves along a Halo orbit in the CR3BP. This reference point is employed in the development of the relative equations of motion. Conceptually, this point may represent a central "chief" spacecraft in the formation. Subsequently, all other vehicles are referred to as "deputies," and their motion is determined relative to the state of the chief in the synodic rotating frame of the CR3BP.

1. Equations of Motion

Define an inertial frame, $\mathcal{I} \equiv \{\hat{\boldsymbol{x}}_{\mathcal{I}}, \hat{\boldsymbol{y}}_{\mathcal{I}}, \hat{\boldsymbol{z}}_{\mathcal{I}}\}$, with origin at the barycenter, B, of the two primary masses (in this case, the Sun and the Earth/Moon) such that the orthogonal unit vectors $\hat{\boldsymbol{x}}_{\mathcal{I}}$ and $\hat{\boldsymbol{y}}_{\mathcal{I}}$ define the plane of motion of the primary bodies. Subsequently, the unit vector $\hat{\boldsymbol{z}}_{\mathcal{I}}$ is normal to the primary plane, and the primaries rotate about the $+\hat{\boldsymbol{z}}_{\mathcal{I}}$ axis.

In the CR3BP, it is convenient to describe the motion of a spacecraft in terms of a rotating coordinate system, $\mathcal{R}$, also centered at B, whose rotation coincides with the mean motion of the primaries. Let the orthogonal unit vectors $\{\hat{\boldsymbol{x}}_{\mathcal{R}}, \hat{\boldsymbol{y}}_{\mathcal{R}}, \hat{\boldsymbol{z}}_{\mathcal{R}}\}$ define the rotating frame, $\mathcal{R}$, where $\hat{\boldsymbol{z}}_{\mathcal{R}} = \hat{\boldsymbol{z}}_{\mathcal{I}}$, and $\hat{\boldsymbol{x}}_{\mathcal{R}}$ is directed from the Sun to the Earth/Moon barycenter. The remaining axis, $\hat{\boldsymbol{y}}_{\mathcal{R}}$, completes the right-handed system such

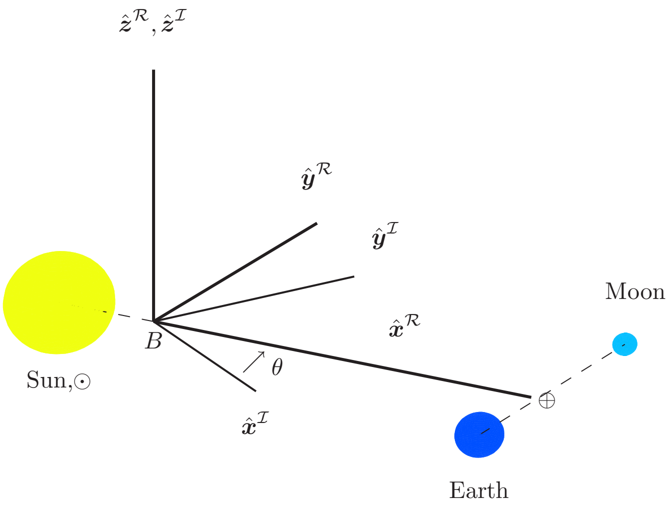

Figure 1. CR3BP Frame

that $\hat{\boldsymbol{y}}_{\mathcal{R}} = \hat{\boldsymbol{z}}_{\mathcal{R}} \times \hat{\boldsymbol{x}}_{\mathcal{R}}$. The $\mathcal{R}$ frame rotates about $\hat{\boldsymbol{z}}_{\mathcal{R}} = \hat{\boldsymbol{z}}_{\mathcal{I}}$ at a rate of ω, equal to the mean motion of the primaries. The system is shown in Figure 1.

Let the position of the chief spacecraft, in terms of the $\mathcal{R}$-frame unit vectors, be defined as $\boldsymbol{r}_c^{\mathcal{R}} \equiv [x_c \; y_c \; z_c]^T$. In addition, assume that

$$
\begin{aligned}
m_\odot &= \text{mass of the Sun,} \\
m_\oplus &= \text{mass of the combined Earth/Moon system,} \\
r_{\odot c} &= \text{distance between the Sun and the chief, and} \\
r_{\oplus c} &= \text{distance between the Earth/Moon barycenter and the chief.}
\end{aligned}
$$

Given the potential function,

$$U = \frac{(x_c^2 + y_c^2)\omega^2}{2} + \frac{Gm_\odot}{r_{\odot c}} + \frac{Gm_\oplus}{r_{\oplus c}} \tag{1}$$

the chief dynamics are described as,

$$
\begin{aligned}
\begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \\ \ddot{z}_c \end{bmatrix} &= \begin{bmatrix} U_{x_c} \\ U_{y_c} \\ U_{z_c} \end{bmatrix} + 2\omega \begin{bmatrix} \dot{y}_c \\ -\dot{x}_c \\ 0 \end{bmatrix} + \begin{bmatrix} u_{x_c} \\ u_{y_c} \\ u_{z_c} \end{bmatrix} \\
&= -\frac{Gm_\odot}{r_{\odot c}^3} \begin{bmatrix} x_c + r_\odot \\ y_c \\ z_c \end{bmatrix} - \frac{Gm_\oplus}{r_{\oplus c}^3} \begin{bmatrix} x_c - r_\oplus \\ y_c \\ z_c \end{bmatrix} + 2\omega \begin{bmatrix} \dot{y}_c \\ -\dot{x}_c \\ 0 \end{bmatrix} + \omega^2 \begin{bmatrix} x_c \\ y_c \\ 0 \end{bmatrix} + \begin{bmatrix} u_{x_c} \\ u_{y_c} \\ u_{z_c} \end{bmatrix},
\end{aligned} \tag{2}
$$

where G is the gravitational constant, and $\boldsymbol{u}_c = [u_{x_c} \; u_{y_c} \; u_{z_c}]^T$ is the external control acceleration.

Let the state of the chief be described by the vector $\boldsymbol{y}_c \equiv \left[\boldsymbol{r}_c^T \; \dot{\boldsymbol{r}}_c^T\right]^T = [x_c \; y_c \; z_c \; \dot{x}_c \; \dot{y}_c \; \dot{z}_c]^T$. Equation 2 is easily transformed into first order form,

$$\dot{\boldsymbol{y}}_c = \tilde{\boldsymbol{f}}(\boldsymbol{y}_c, \boldsymbol{u}_c) = \tilde{\boldsymbol{f}}(\boldsymbol{y}_c, \boldsymbol{0}), \tag{3}$$

where $\boldsymbol{u}_c = \boldsymbol{0}$ since the chief spacecraft is assumed to evolve along a natural arc.

The general form of the motion in Equation 2 applies to the states of the l^{th} deputy, such that the state space representation is given by

$$\dot{\boldsymbol{y}}_{d_l} = \tilde{\boldsymbol{f}}(\boldsymbol{y}_{d_l}, \boldsymbol{u}_l). \tag{4}$$

The motion of the l^{th} deputy relative to the chief is determined first by identifying the relative state vector,

$$\boldsymbol{y}_l \equiv \boldsymbol{y}_{d_l} - \boldsymbol{y}_c. \tag{5}$$

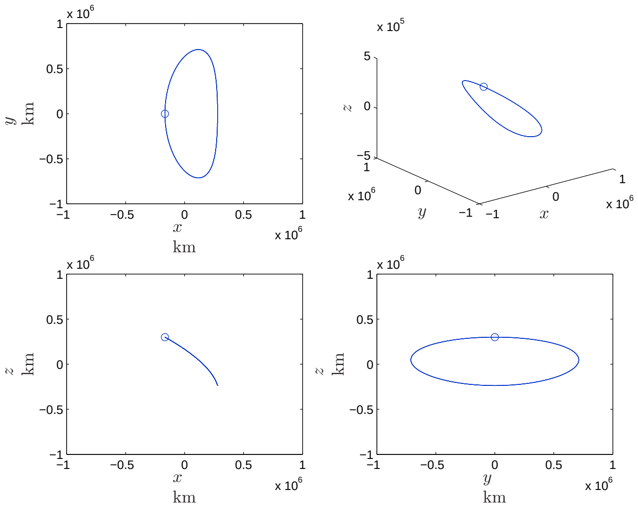

Figure 2. Reference Halo Orbit for Chief Spacecraft with Origin at L$_1$

Subsequently, differentiating Equation 5 implies that

$$\begin{aligned} \dot{\boldsymbol{y}}_l &= \tilde{\boldsymbol{f}}(\boldsymbol{y}_{d_l}, \boldsymbol{u}_l) - \tilde{\boldsymbol{f}}(\boldsymbol{y}_c, \boldsymbol{0}), && (6) \\ &= \tilde{\boldsymbol{f}}(\boldsymbol{y}_c + \boldsymbol{y}_l, \boldsymbol{u}_l) - \tilde{\boldsymbol{f}}(\boldsymbol{y}_c, \boldsymbol{0}), && (7) \\ &= \boldsymbol{f}(t, \boldsymbol{y}_l, \boldsymbol{u}_l). && (8) \end{aligned}$$

Notice that the functions $\tilde{\boldsymbol{f}}$ and $\boldsymbol{f}$ are not the same. It is apparent from Equation 7 that the relative dynamics of the l^{th} deputy depend on the absolute state of the chief, the relative state of the deputy, and the control effort exerted by the deputy. Through manipulation of the arguments in Equation 7, it is observed that the velocity terms related to the chief, $\dot{\boldsymbol{r}}_c$, cancel away. Since the chief satellite is assumed to follow a natural trajectory from some epoch and the path is predetermined, $\tilde{\boldsymbol{f}}$ reduces to $\boldsymbol{f}$ in Equation 8, which depends on time, the relative state of the deputy, and the external control of the deputy. Equation 8 is the dynamical model used to describe the motion of a deputy spacecraft.

Note that traditional formulations of the CR3BP equations employ nondimensional variables.[27] The formulation presented here, however, uses dimensional variables to allow increased flexibility in the scaling of the numerical algorithm.

2. *Reference Halo Orbit*

As previously stated, the chief spacecraft is assumed to evolve along an L$_1$ Halo Orbit, as determined in the Sun-Earth/Moon CR3BP. The initial state at the reference epoch is defined in $\mathcal{R}$-frame coordinates as

$$\begin{aligned} \boldsymbol{r}_c(t_0) &= \boldsymbol{r}_{L_1} + \begin{bmatrix} -166,783.32 \\ 0 \\ 300,000.00 \end{bmatrix} \text{km} && (9) \\ \dot{\boldsymbol{r}}_c(t_0) &= \begin{bmatrix} 0 \\ 281.28 \\ 0 \end{bmatrix} \text{m/s}. && (10) \end{aligned}$$

The position vector, $\boldsymbol{r}_{L_1}$, is the position of L$_1$ from the Sun-Earth/Moon barycenter, approximately 148×10^6 km in the $\hat{\boldsymbol{x}}_{\mathcal{R}}$-direction. Thus, the Halo orbit begins at its northern most point, 300,000 km north of the libration point. The Halo orbit is displayed in Figure 2. The trajectory shown takes place over one orbital period, approximately 178 days.

II. Transcription Formulations for the Actuator Constrained Optimal Control Problem

For a formation near a libration point, spacecraft maneuvering exhibits the unusual limitation that ideal control solutions require thrust acceleration magnitudes smaller than existing technology can provide.[17–22] In addition, it is conceivable that spacecraft orientation may be restricted by mission requirements, which can inhibit maneuverability depending on the number and placement of thrusters on the spacecraft body. When these two conditions are combined, the control problem is extremely constrained, motivating the development of unique solution methods to handle these constraints. It is clear that with limitations in thrust magnitude and direction, precision tracking is not achievable. However, optimal control methods can be used to determine the level of tracking accuracy achievable under the circumstances.

In this context, direct numerical optimization methods are modified to treat these unique constraints. The foundation of this approach is a general collocation method, presented first. The method is then altered to divide a problem into multiple segments, allowing for ideal treatment of the control discontinuities associated with finite burn control. Finally, the formulation is expanded to account for spacecraft orientation restrictions. This ultimately requires dynamic changes in the time dependencies of each segment, but it is this modification that leads to feasible formation control solutions.

A. A General Direct Method for Solving Optimal Control Problems

Consider a general Optimal Control Problem presented in the form of Bolza.[28] The objective is to determine the control history, $\boldsymbol{u}(t)$, which minimizes the cost function,

$$J = \phi(t_0, \boldsymbol{y}_0, t_f, \boldsymbol{y}_f) + \int_{t_0}^{t_f} L(t, \boldsymbol{y}, \boldsymbol{u})\, dt, \tag{11}$$

where t_0 is the initial time, t_f is the final time, $\boldsymbol{y} \in \mathbb{R}^{n_y}$ denotes the state vector, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ represents the control vector, $\boldsymbol{y}_0 = \boldsymbol{y}(t_0)$, and $\boldsymbol{y}_f = \boldsymbol{y}(t_f)$. The optimal solution must satisfy the vector dynamical constraint,

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{u}), \tag{12}$$

as well as the constraints,

$$\begin{aligned} \mathbf{0} &= \boldsymbol{\psi}_0(t_0, \boldsymbol{y}_0), && (13)\\ \mathbf{0} &= \boldsymbol{\psi}_f(t_f, \boldsymbol{y}_f), && (14)\\ \mathbf{0} &= \boldsymbol{\beta}(t, \boldsymbol{y}, \boldsymbol{u}). && (15) \end{aligned}$$

Here, $\boldsymbol{\psi}_0 \in \mathbb{R}^{n_{\psi_0}}$ is a vector of initial conditions, $\boldsymbol{\psi}_f \in \mathbb{R}^{n_{\psi_f}}$ denotes the terminal constraints, and $\boldsymbol{\beta} \in \mathbb{R}^{n_\beta}$ represents the path constraints imposed over all $t \in [t_0, t_f]$. Note, though all the constraints presented here are formulated as equality constraints, that is not a necessary restriction. Extensions are available to accommodate inequality constraints as well. For example, any inequality constraint, $\tilde{c} \leq 0$, can be converted into an equality constraint, $c = 0$, with the use of a slack variable:[28]

$$c = \tilde{c} + \alpha^2 = 0. \tag{16}$$

In this case, the slack variable, α, is treated as a control parameter. When $\tilde{c}$ is satisfied along the boundary, $\alpha = 0$.

A direct collocation method begins with discretization of the states and controls. Let the optimal control problem be discretized in time at n_n different places called *nodes* associated with times t_j, $j = 1, \ldots, n_n$. If the discretized times are assumed to be uniformly spaced over the interval $[t_0, t_f]$, then

$$t_j = t_0 + \frac{j-1}{n_n - 1}\left(t_f - t_0\right). \tag{17}$$

Consider a parameter vector containing each of the states and controls at the nodes, along with the initial and final time.

$$\boldsymbol{x} = \left[y_1(t_0) \ \cdots \ y_i(t_j) \ \cdots \ y_{n_y}(t_f) \ u_1(t_0) \ \cdots \ u_i(t_j) \ \cdots \ u_{n_u}(t_f) \ t_0 \ t_f\right]^T \tag{18}$$

State variables, control variables, and time are all represented in this parameterization so that cost and constraint functions in terms of $\boldsymbol{x}$ can easily be derived to represent the optimal control problem. The subscript i identifies the i^{th} element of a state or control vector, while j counts the nodes. This convention is maintained throughout the present development. The parameter vector in Equation 18 is of dimension

$$n = (n_y + n_u)n_n + 2, \tag{19}$$

indicating the number of parameters that are optimized by the NLP algorithm.

The next step in collocation is to define cost and constraint functions in terms of $\boldsymbol{x}$. The vector constraint equation, $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0}$, accounts for the initial, final, path, and dynamical constraints. Specifically, let $\boldsymbol{c}(\boldsymbol{x})$ be decomposed as

$$\boldsymbol{c}(\boldsymbol{x}) = \left[\boldsymbol{c}_{\psi_0}^T(\boldsymbol{x})\ \boldsymbol{c}_{\psi_f}^T(\boldsymbol{x})\ \boldsymbol{c}_{\beta}^T(\boldsymbol{x})\ \boldsymbol{c}_{\dot{y}}^T(\boldsymbol{x})\right]^T. \tag{20}$$

Each of the elements of Equation 20 is defined over the next sections of this document.

1. Point and Path Constraints

The initial and terminal constraints are defined as

$$\begin{aligned} \boldsymbol{c}_{\psi_0}(\boldsymbol{x}) &= \boldsymbol{\psi}_0(t_0, \boldsymbol{y}(t_0)) \quad (21)\\ \boldsymbol{c}_{\psi_f}(\boldsymbol{x}) &= \boldsymbol{\psi}_f(t_f, \boldsymbol{y}(t_f)), \quad (22) \end{aligned}$$

where $\boldsymbol{c}_{\psi_0}$ contributes n_{ψ_0} constraints and $\boldsymbol{c}_{\psi_f}$ consists of n_{ψ_f} constraints.

The path constraint, $\boldsymbol{\beta}$, is enforced continuously for all $t \in [t_0\ t_f]$. With a parameterized set of states, controls, and times, a logical way of implementing a path constraint is at each node individually. The n_β path constraints are imposed at each of the nodes, such that

$$\begin{aligned} \boldsymbol{c}_{\beta_j} &= \boldsymbol{\beta}(t_j, \boldsymbol{y}(t_j), \boldsymbol{u}(t_j)), \quad (23)\\ \boldsymbol{c}_\beta &= \left[\boldsymbol{c}_{\beta_1}^T \cdots \boldsymbol{c}_{\beta_{n_n}}^T\right]^T. \quad (24) \end{aligned}$$

This leads to $n_\beta n_n$ path constraints. Note that in this formulation, the path constraints are not necessarily enforced between the nodes. However, this is easily resolved, for example, by selecting a representative number of nodes in the transcription. Making n_n large limits the spacing between nodes and increases the number of path constraints. As $n_n \to \infty$, the constraints of $\boldsymbol{c}_\beta$ approach the continuous path constraints $\boldsymbol{\beta}$.

2. Dynamical Constraints

Many different explicit or implicit integration schemes are available to model the continuity constraints that enforce the dynamics of Equation 12. The scheme selected determines the characteristics of the transcription. In this investigation, a Hermite-Simpson integration scheme is employed. This implicit integration scheme requires three function evaluations and delivers an accuracy on the order of $\mathcal{O}(h^5)$, where $h = t_{j+1} - t_j$. Define

$$\begin{aligned} t_m &= \frac{1}{2}\left(t_j + t_{j+1}\right), \quad (25)\\ \boldsymbol{y}_m &= \frac{1}{2}\left(\boldsymbol{y}_j + \boldsymbol{y}_{j+1}\right) + \frac{h}{8}\left(\boldsymbol{f}_j - \boldsymbol{f}_{j+1}\right), \quad (26)\\ \boldsymbol{u}_m &= \frac{1}{2}\left(\boldsymbol{u}_j + \boldsymbol{u}_{j+1}\right), \quad (27)\\ \boldsymbol{f}_m &= \boldsymbol{f}(t_m, \boldsymbol{y}_m, \boldsymbol{u}_m). \quad (28) \end{aligned}$$

Then, the estimate of the state at t_{j+1} is given by[29]

$$\hat{\boldsymbol{y}}_{j+1} = \boldsymbol{y}_j + \frac{h}{6}\left(\boldsymbol{f}_j + 4\boldsymbol{f}_m + \boldsymbol{f}_{j+1}\right). \tag{29}$$

Note that the process is not self-starting. An initial guess for $\boldsymbol{y}_{j+1}$ is necessary when evaluating this equation. Also, the new estimate, $\hat{\boldsymbol{y}}_{j+1}$, does not necessarily match the known states at time t_{j+1}. However, an equality constraint can be imposed in the parameter optimization problem to ensure that they do match at the final iteration. Thus, define the constraint,

$$\begin{aligned}\boldsymbol{c}_{\dot{y}_j} &= \boldsymbol{y}_{j+1} - \hat{\boldsymbol{y}}_{j+1} \\ &= \boldsymbol{y}_{j+1} - \boldsymbol{y}_j - \frac{h}{6}\left(\boldsymbol{f}_j + 4\boldsymbol{f}_m + \boldsymbol{f}_{j+1}\right). \qquad (30)\end{aligned}$$

The residual equation is a vector of size n_y, and it can be imposed between each node for $j = 1, \ldots, n_n - 1$. Assembling all of these constraints into a single vector leads to $n_y(n_n - 1)$ continuity constraints,

$$\boldsymbol{c}_{\dot{y}} = \left[\boldsymbol{c}_{\dot{y}_1}^T \ \cdots \ \boldsymbol{c}_{\dot{y}_{n_n-1}}^T\right]^T. \qquad (31)$$

When $\boldsymbol{c}_{\dot{y}} = \boldsymbol{0}$ to within some specified tolerance, the dynamical constraint, $\dot{\boldsymbol{y}} = \boldsymbol{f}$, is met.

B. Multiple Segments and Knots

Consider the parameterization defined by Equation 18. A byproduct of this formulation is that the nodes along the solution are evenly spaced between $[t_0, t_f]$. The time associated with a given node is assigned based on Equation 17. Of course, a constraint can be imposed anywhere along the path, not necessarily at a point that coincides with a node. Consider, for example, a constraint associated with t = 4.2. If $n_n = 10$, $t_0 = 1$ and $t_f = 10$, then $t_j = j$, and the time at each node is an integer value. Clearly, the parameterization considered thus far is not equipped to handle this constraint. The ability to incorporate constraints at any point along a trajectory is necessary in many applications.

For example, state discontinuities require intermediate constraints. An orbital transfer that employs impulsive maneuvers requires some condition $\boldsymbol{v}^+ - \boldsymbol{v}^- = \Delta\boldsymbol{v}$. A staged rocket experiences mass discontinuities between stages. In each of these cases, the dynamics are impacted by these discontinuities.

Control discontinuities, on the other hand, do not necessarily require intermediate constraints. It is conceivable for an optimal control history to exhibit "bang-bang" or "bang-off-bang" behavior, and optimality conditions exist under these circumstances.[30] However, there are no dynamic relations for the control effort between nodes. Thus, a parameterized solution may exhibit $u(t_j)$ at one extreme and $u(t_{j+1})$ at the other extreme, and this may sufficiently represent the discontinuity. If the optimal solution includes control discontinuities, though, an alternate parameterization is sought that is capable of implementing perfect (instantaneous) control switching.

The discontinuity divides the trajectory into separate subarcs, or *segments.* In addition to discontinuities, dividing a trajectory into different segments may be desirable to accomodate changes in the dynamic model along the path. The path constraints on the states and/or controls may also change based on various conditions. Thus, it is advantageous to parameterize the problem into multiple segments.

In the event a solution includes multiple subarcs, knots represent one alternative to devise a modified transcription that accounts for any discontinuities introduced in the solution. A *knot* is a point that divides two segments and may be either fixed or free in time. Knots represent an effective means of incorporating constraints that vary by segment or discontinuities in either the state or control variables. Jump discontinuities occur at a knot. For example, with finite burn control, the control may take on only a discrete set of values, such as zero and a specified thrust magnitude, i.e. $u \in \{0, T^*\}$. Thus, a knot is placed at every point where the control switches value.

Figure 3 illustrates the conceptual relation between knots, nodes, and segments at two consecutive iterations. Consider a formulation with n_s segments. Each segment starts at either t_0 or a knot and ends at either another knot or t_f. Thus, the formulation consists of $n_k = n_s - 1$ knots.

Let every segment consist of n_n nodes. Although, conceptually, each segment may consist of a different number of nodes, the transcription is simplified by assuming an equal number of nodes per segment. Furthermore, while the nodes are uniformly spaced within a segment, they are not necessarily uniformly spaced along the course of the trajectory. That is, each segment may span a different length of time. In most cases, it is desirable for the segment duration (or the location of the knots) to be optimized as well. In Figure 3, the knot times change between iteration p and iteration $p + 1$ as they are optimized by the NLP algorithm. Consequently, assigned times for nodes change respectively to maintain uniform spacing per segment.

The easiest implementation of a multiple-segment formulation includes the terminal times of each segment (that is, t_0, the interior knots, and t_f) in the parameter vector. The interior knots contribute n_k parameters,

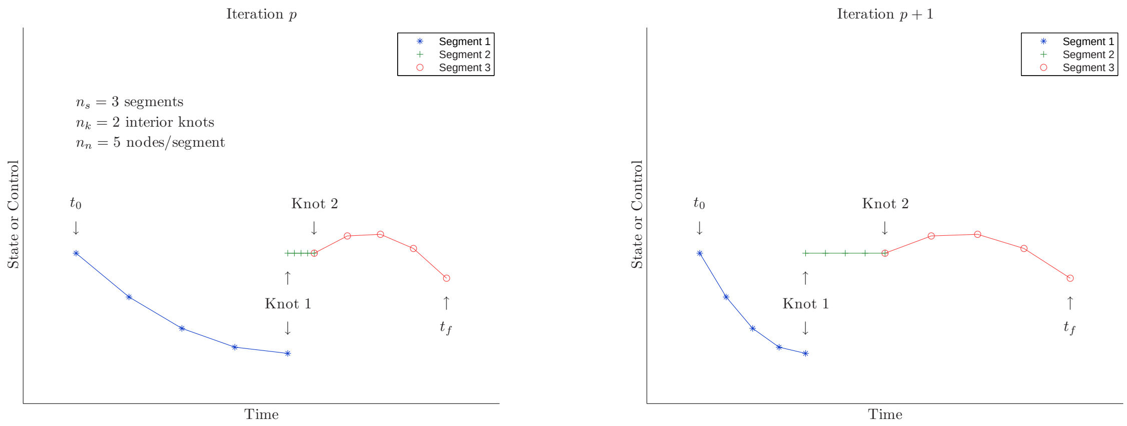

Figure 3. An Example of Segments and Knots at Two Consecutive Iterations

while t_0 and t_f are included in $\boldsymbol{x}$ as before. The parameterization described has n_s segments, each with n_n nodes. Each node is associated with n_y states and n_u controls. The parameter vector is defined as,

$$\boldsymbol{x} = \left[\cdots\ \boldsymbol{y}_{j,k}^T\ \cdots\ \cdots\ \boldsymbol{u}_{j,k}^T\ \cdots\ \cdots\ t_k\ \cdots\ t_0\ t_f\right]^T. \tag{32}$$

In Eq. 32, $\boldsymbol{y}_{j,k} = \boldsymbol{y}\left(t_{j,k}\right)$ and $\boldsymbol{u}_{j,k} = \boldsymbol{u}\left(t_{j,k}\right)$ represent the state and control vectors (respectively) at the j^{th} node of the k^{th} segment,

$$t_{j,k} = t_{k-1} + \frac{j-1}{n_n - 1}\left(t_k - t_{k-1}\right) \quad j = 1, \ldots, n_n,\ k = 1, \ldots, n_s, \tag{33}$$

and $t_{n_s} = t_f$. The parameters $t_k, k = 1, \ldots, n_k$, are the times at which each interior knot occurs. Thus, i identifies members of a vector, j denotes the respective nodes, and k represents the relevant knot or segment. The parameters can also be listed as

$$\boldsymbol{x} \;=\; [\underbrace{\cdots\ y_{i,j,k}\ \cdots}_{n_y n_n n_s}\underbrace{\cdots\ u_{i,j,k}\ \cdots}_{n_u n_n n_s}\underbrace{\cdots\ t_k\ \cdots}_{n_k}\underbrace{t_0\ t_f}_{2}]^T. \tag{34}$$

and the dimension of the parameter vector is $n = (n_y + n_u)n_n n_s + n_k + 2$. The number of variables for multiple segments is approximately n_s times the number of variables for a single segment. However, multiple segments also introduce additional constraints.

$$\boldsymbol{c}(\boldsymbol{x}) = \left[\boldsymbol{c}_{\psi_0}^T(\boldsymbol{x})\ \boldsymbol{c}_{\psi_f}^T(\boldsymbol{x})\ \boldsymbol{c}_{\beta}^T(\boldsymbol{x})\ \boldsymbol{c}_{\dot{y}}^T(\boldsymbol{x})\ \boldsymbol{c}_{s}^T(\boldsymbol{x})\ \tilde{\boldsymbol{c}}_{t}^T(\boldsymbol{x})\right]^T \tag{35}$$

The initial conditions and final conditions remain unchanged. The path constraints, $\boldsymbol{c}_\beta(\boldsymbol{x})$, still impose the n_β conditions of $\boldsymbol{\beta}$ at each of the nodes. That is, within a segment, there are $n_\beta n_n$ elements in $\boldsymbol{c}_\beta$. Since there may be path constraints along each segment, $\boldsymbol{c}_\beta$ becomes a vector of dimension $n_\beta n_n n_s$. Continuity constraints within a segment are similarly addressed using the Hermite-Simpson implicit integration scheme to impose the dynamical conditions. However, since there are n_s segments, $n_y(n_n - 1)n_s$ continuity constraints are included in $\boldsymbol{c}_{\dot{y}}$.

The next set of constraints, $\boldsymbol{c}_s$, establishes conditions between segments (at the knots). These may include intermediate point constraints or continuity conditions. Generally, the vector $\boldsymbol{c}_s$ is used to address continuity between segments. At the knots, time continuity is assumed so that the last node of the k^{th} segment occurs at the same time as the first node of the $(k+1)^{\text{th}}$ segment. Additionally, state continuity is required across segments. Thus, position and velocity states are subject to equality constraints. The behavior illustrated at Knot 2 in Figure 3 is representative of a state continuity constraint at a knot. Thus,

$$\boldsymbol{c}_{s_k} \;=\; \boldsymbol{y}_{1,k+1} - \boldsymbol{y}_{n_n,k}, \quad k = 1, \ldots, n_s - 1, \tag{36}$$

$$\boldsymbol{c}_s \;=\; \left[\boldsymbol{c}_{s_1}^T\ \cdots\ \boldsymbol{c}_{s_{n_s-1}}^T\right]^T, \tag{37}$$

and $\boldsymbol{c}_s$ is a vector of dimension $n_y(n_s - 1)$.

The last set of constraints are time inequality constraints of the form $\tilde{\boldsymbol{c}}_t(\boldsymbol{x}) \leq \boldsymbol{0}$. Although knots are allowed to float freely in time, their order must be chronological. If the k^{th} knot lags behind the $(k-1)^{\text{th}}$ knot, for example, the duration of the associated segment is negative, a nonsensical result. To prevent this behavior, let

$$\begin{aligned} \tilde{c}_{t_k} &= t_{k-1} - t_k, \quad k = 1, \ldots, n_s, && (38) \\ \tilde{\boldsymbol{c}}_t &= \left[\tilde{c}_{t_1} \; \cdots \; \tilde{c}_{t_{n_s}}\right]^T, && (39) \end{aligned}$$

where, again $t_{n_s} = t_f$. The variable segment durations produce n_s time conditions.

The total number of constraints, including initial conditions, final conditions, path constraints, dynamical constraints, knot continuity constraints, and time inequality constraints becomes

$$n_c = n_{\psi_0} + n_{\psi_f} + n_\beta n_n n_s + n_y(n_n - 1)n_s + n_y(n_s - 1) + n_s. \tag{40}$$

The vector $\boldsymbol{c}(\boldsymbol{x})$ now includes both equality and inequality constraints. In setting up the NLP problem, the user must exercise care to properly differentiate between the two types of constraints.

A more convenient parameterization includes the duration of the segments, Δt_k, in the parameter vector $\boldsymbol{x}$, instead of the knot times. Consider the parameter vector,

$$\boldsymbol{x} \;=\; [\underbrace{\cdots \; y_{i,j,k} \; \cdots}_{n_y n_n n_s} \; \underbrace{\cdots \; u_{i,j,k} \; \cdots}_{n_u n_n n_s} \; \underbrace{\cdots \; \Delta t_k \; \cdots}_{n_s} \; \underbrace{t_0 \; t_f}_{2}]^T. \tag{41}$$

The dimension of $\boldsymbol{x}$ is now $n = (n_y + n_u)\, n_n n_s + n_s + 2$, adding one dimension to the previously identified parameterization (n_s durations are needed instead of n_k times). This offers the advantage of converting the inequality constraints of $\tilde{\boldsymbol{c}}_t(\boldsymbol{x})$ into a single equality constraint. Let the knot times be defined as

$$t_k = t_0 + \sum_{\kappa=1}^{k} |\Delta t_\kappa| \,. \tag{42}$$

Then the inequality, $t_{k-1} - t_k \leq 0$, is guaranteed, and need not be included in the constraint vector, $\boldsymbol{c}(\boldsymbol{x})$. Instead $\tilde{\boldsymbol{c}}_t(\boldsymbol{x})$ is replaced by the single time equality constraint,

$$c_t(\boldsymbol{x}) = (t_f - t_0) - \sum_{k=1}^{n_s} |\Delta t_k| \,. \tag{43}$$

Now, the total constraint vector is simply

$$\boldsymbol{c}(\boldsymbol{x}) = \left[\boldsymbol{c}_{\psi_0}^T(\boldsymbol{x}) \; \boldsymbol{c}_{\psi_f}^T(\boldsymbol{x}) \; \boldsymbol{c}_\beta^T(\boldsymbol{x}) \; \boldsymbol{c}_y^T(\boldsymbol{x}) \; \boldsymbol{c}_s^T(\boldsymbol{x}) \; c_t(\boldsymbol{x})\right]^T, \tag{44}$$

which includes

$$n_c = n_{\psi_0} + n_{\psi_f} + n_\beta n_n n_s + n_y(n_n - 1)n_s + n_y(n_s - 1) + 1 \tag{45}$$

equality constraints. An appealing aspect of this alternate formulation is that it is not necessary to track, during the solution process, which constraints are active. In this formulation, all constraints are active at all times. This formulation also enhances the convergence properties of the algorithm.

For instance, constraints may not necessarily be satisfied at a given iteration leading to convergence. The inequality constraints previously mentioned are applied to avoid the occurrence of segments of negative duration. However, in the intermediate step leading to a solution, some segments may violate this constraint. Depending upon the specific nature of the problem, this may prevent the optimizer from converging on a solution. The formulation employing segment durations ensures this situation does not arise.

Furthermore, $|\Delta t_k|$ can be replaced by Δt_k if a simple bound is placed on the variable to limit it to nonnegative values. In other words, the numerical algorithm can be instructed not to search for a solution where Δt_k is negative. This has the same effect as the absolute values formulation.

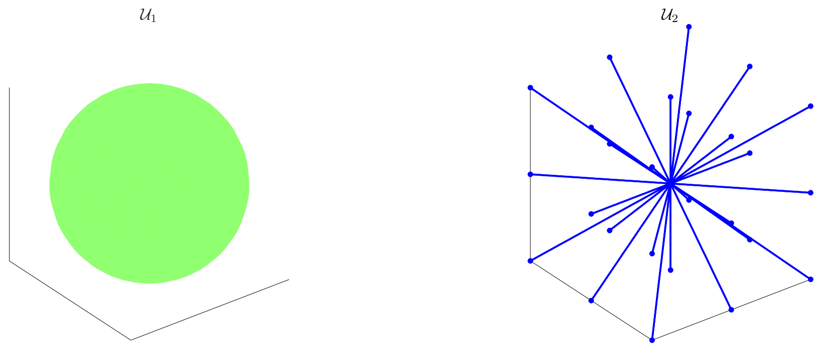

Figure 4. Control Spaces (a) $\mathcal{U}_1$ (Orientation Free), and (b) $\mathcal{U}_2$ (Orientation Fixed)

C. Switching Segments and Time for Spacecraft Orientation Restrictions

The multiple-segment formulation presented facilitates the realization of finite control acceleration solutions for spacecraft problems. For example, a segment can be pre-designated to be either a thrusting arc or a coasting arc, and the optimizer can determine the durations for each segment and the dynamic thrust direction (i.e. spacecraft orientation profile) during each thrusting arc. In this case, the parameters, Δt_k, indicate thrust durations, while $\boldsymbol{u}_{j,k}$ represents the thrust direction at each node. This formulation is relevant when the control pointing direction is unconstrained.

Additional modifications are necessary to handle restrictions on the orientation of a spacecraft. This investigation considers the case when mission requirements dictate the orientation of a spacecraft for arbitrarily long time durations, imposing significant limitations on spacecraft control. An alternative formulation to the transcribed optimal control problem is now considered to address this highly specific restriction. The orientation of the vehicles is assumed to be completely pre-specified and independently controlled. Position control is still possible, however, as long as the spacecraft is able to generate thrust along each axis.

Assume that the orientation of the l^{th} deputy spacecraft is described by a body-fixed coordinate frame $\mathcal{B}_l \equiv \{\hat{\boldsymbol{x}}_{\mathcal{B}_l}, \hat{\boldsymbol{y}}_{\mathcal{B}_l}, \hat{\boldsymbol{z}}_{\mathcal{B}_l}\}$, the thrusters are aligned with the $\mathcal{B}_l$ frame, and they are either off or producing accelerations of magnitude T^* along the principal directions. Thus,

$$u_{i,l}\left(u_{i,l} - T^*\right)\left(u_{i,l} + T^*\right) = 0, \quad i = \hat{\boldsymbol{x}}_{\mathcal{B}_l}, \ldots, \hat{\boldsymbol{z}}_{\mathcal{B}_l}, \quad l = 1, \ldots, n_d. \tag{46}$$

The control histories for all trajectories must meet this constraint. If the control effort is limited to the set $\mathcal{U} = \{-T^*, 0, T^*\}$ along a principal direction, then the optimal solution determines both the time at which a burn is initiated and the duration. The action commanded for each thruster is independent of the commands issued to all other thrusters. This problem creates several additional difficulties in determining feasible, and certainly optimal, solutions.

The first difficulty arises from the fact that there are only a finite number of potential thrust configurations. This is easily demonstrated by comparing the solution space for orientation free versus orientation fixed spacecraft. In Figure 4, the control spaces, $\mathcal{U}_1$ and $\mathcal{U}_2$, are compared, where $\mathcal{U}_1$ and $\mathcal{U}_2$ are the sets of all possible control values in the orientation free and orientation fixed scenarios, respectively. Clearly, in both scenarios, the spacecraft is free to coast (zero thrust). If the attitude of the spacecraft is free to change, for a given thrust magnitude, the thrust vector may be aligned with any vector originating from the center and terminating on the surface of the $\mathcal{U}_1$ sphere of radius T^*. In contrast, the orientation fixed scenario has only a discrete set of possible control values. In addition, when thrusting in a combination of two or three axes directions, the net thrust magnitude is *larger* than T^*. The variable T^* represents the lower limit for feasible thruster performance, which is likely greater than an ideal (unconstrained) control acceleration. When the net magnitude in a non-principal direction is even larger than T^*, spacecraft maneuverability becomes even more constrained.

An additional difficulty arises during the transcription of the optimal control problem. At any time, there are a finite number of thrust options. With $n_u = 3$ thrusting directions (in this case $\hat{\boldsymbol{x}}_{\mathcal{B}_l}$, $\hat{\boldsymbol{y}}_{\mathcal{B}_l}$, and $\hat{\boldsymbol{z}}_{\mathcal{B}_l}$), and three possible control values in each direction $(-T^*, 0, T^*)$, there are 3^3 acceleration options for the control.

In general there are 3^{n_u} possible control combinations for a spacecraft, since n_u may be greater than three if there were thrusters in non-principal directions. The goal of this study is to devise a method that allows the optimizer to choose the optimal switching sequence for each thruster.

One way to accomplish this is to use the constraint in Equation 46. Unfortunately, gradient methods move a point towards the closest root of the constraint, independent of whether it is the *right* root to choose. Once the constraint is met, it is difficult, if not impossible, for an NLP algorithm to seek another root. Essentially, the NLP algorithm is asked to optimize over a finite set of values. However, gradients cannot be defined between "on" and "off" for a thruster. Although the constraint is valid, it cannot be implemented with traditional transcription.

To circumvent this problem, consider the multiple-segment formulation of Section B. Instead of optimizing the control values, *assume* the control values and optimize the *switching times.* In one implementation, a formulation is devised that assigns each segment one of the 3^{n_u} options, and then requires the optimizer to determine the duration of each segment. With this formulation, the predetermined control values are removed from the parameter vector.

For example, a segment k is designated with control $\boldsymbol{u}(t_{j,k}) = [T^* \ -T^* \ 0]^T$ for all nodes $j = 1, \ldots, n_n$ along that segment. If this control combination is not desirable in the optimal solution, the optimizer determines that the duration of that segment, Δt_k, is zero. With $n_s = 3^{n_u}$ segments, it is conceivable that the durations of several, if not a significant number of segments are reduced to zero. Recall that any segment includes parameters for the n_n nodes with n_y states assigned to each node. Since no time elapses over a zero duration segment, no changes are applied to any of the states over that segment. Thus a zero duration segment implies that some to many of the variables in the NLP problem are essentially wasted, taking up space and computational time, but contributing nothing to the solution.

In addition, without having insight into the optimal solution, no information is available regarding the order of the segments. Consider the optimal solution that requires thrusting in the $\hat{\boldsymbol{z}}$-direction followed by thrusting in the $\hat{\boldsymbol{x}}$-direction. If the segments are not prearranged in that order, there is no way to achieve this solution under the formulation with only 3^{n_u} segments. Instead, one might include multiple sets of 3^{n_u} segments. This also allows for multiple firings in any direction. With m sets, there are now $n_s = (3^{n_u})\, m$ segments. The resulting NLP problem is naturally large, with many wasted variables to slow down the algorithm.

The formulation ultimately selected for this investigation minimizes the number of variables while offering the greatest flexibility in the solution process. Knots and segments are employed, but their definitions are slightly altered. This formulation is considerably more complex and non-intuitive in contrast to the traditional multiple-segment formulation. However, these inconveniences are outweighed by the formulation's ability to consistently determine feasible and optimal solutions.

The salient feature of this formulation is that *each of the control axes is treated independently*, in contrast to earlier formulations that consider all control combinations and subsequently increase n_s. Let a knot be defined as any interior point where one member of the control vector switches from one value to another. The fundamental distinction here is that each knot *is associated with a control axis.* Before, the knot times could be described by t_k, an array with n_k values. Likewise, the segment durations were Δt_k, and there were $n_s = n_k + 1$ segments. Now, the knot times are described as $t_{i,k}$, a two-dimensional array with $n_u \times n_k$ values. In other words, there are knots for each control axis. The *axis durations* are also two-dimensional, as $\Delta t_{i,k}$, with $n_u \times (n_k + 1)$ values. These are the time durations between two switches in a given axis. It is important to note that this is not necessarily the duration of the segments. A segment is now defined as the time elapsed between switches in *any* of the controls. For example, the first segment is bounded both by t_0 and the first switch in the control, regardless of the axis in which the switch occurs. With $n_u n_k$ interior knots to separate the segments, the total number of segments becomes $n_s = n_u n_k + 1$.

Consider the concept demonstrated, as an example, with $n_k = 3$ interior knots per each of $n_u = 3$ control axes. Figure 5 demonstrates the concept at two consecutive iterations of the optimization process. The control values are pre-designated in a logical pattern. In each control axis, the control value begins at T^* (a positive thrusting arc), and a coasting arc follows the first knot. At the second knot, the control value switches to $-T^*$ (a negative thrusting arc) until the third knot, where another coasting arc begins. With additional knots, this pattern continues. The control values in each of the n_u control axes is designated similarly. In this example, observe that the total number of segments is $n_s = (3)(3) + 1 = 10$.

Let the time elements of the parameter vector, $\boldsymbol{x}$, be identified as $\boldsymbol{x}_t$. The time elements include all of

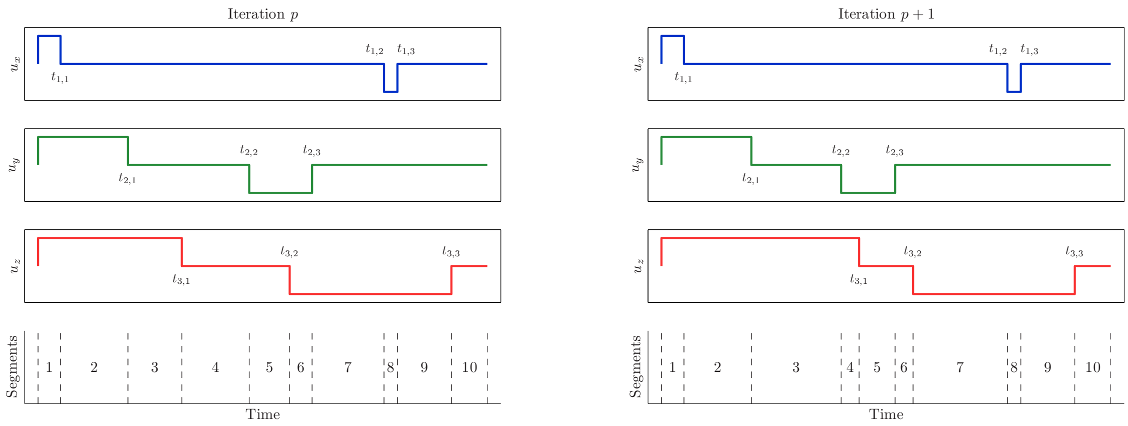

Figure 5. Conceptual Control Profile with Segment Divisions at Two Consecutive Iterations

the axis durations, $\Delta t_{i,k}$, along with the initial and final time. Here, $\boldsymbol{x}_t$ is defined as

$$\boldsymbol{x}_t = \begin{bmatrix} \Delta t_{1,1} & \Delta t_{1,2} & \Delta t_{1,3} & \Delta t_{2,1} & \Delta t_{2,2} & \Delta t_{2,3} & \Delta t_{3,1} & \Delta t_{3,2} & \Delta t_{3,3} & t_0 & t_f \end{bmatrix}^T. \tag{47}$$

The knot times, $t_{i,k}$, are defined according to elements in $\boldsymbol{x}_t$.

$$t_{i,k} = t_0 + \sum_{\kappa=1}^{k} |\Delta t_{i,\kappa}|\,. \tag{48}$$

Thus, $\Delta t_{i,k}$, represents the time duration between the knots located at $t_{i,k-1}$ and $t_{i,k}$. The definition in Equation 48 guarantees that the knot times remain in chronological order for a given control axis. That is,

$$t_{i,k-1} - t_{i,k} \leq 0, \quad k = 1, \ldots, n_k + 1 \tag{49}$$

where $t_{i,0} = t_0$ and $t_{i,n_k+1} = t_f$. For example, in Figure 5, $t_{2,1}$ preceeds $t_{2,2}$, which is followed by $t_{2,3}$. However, there is no relation between knot times in different control axes. Therefore, $t_{2,1}$ need not be after $t_{1,1}$. In general,

$$\iota < i \not\Rightarrow t_{\iota,\kappa} \leq t_{i,k} \;\forall\, i, \iota = 1, \ldots, n_u, \;\forall\, k, \kappa = 1, \ldots, n_k. \tag{50}$$

Compare the arrangement of knot times for iteration p and iteration $p+1$ in the figure. Both illustrations represent valid arrangements for the knots. However, it is clear that the chronological ordering of all the knots may change during the optimization process.

This algorithm defines segment boundaries by the chronological ordering of knot times, including t_0 and t_f at the end points. Let the unordered segment boundaries be defined as

$$\left[t'_0 \cdots t'_\kappa \cdots t'_{n_s}\right] = \left[t_0 \; t_{1,1} \cdots t_{i,k} \cdots t_{n_u,n_k} \; t_f\right]. \tag{51}$$

Because knots are completely free to move on $[t_0 \; t_f]$ and are independent between control axes, the segment boundaries, t'_κ, are not necessarily in ascending order. Thus, a sorting algorithm converts t'_κ into the sequential listing, t_k, of the segment boundaries. Now, t_{k-1} and t_k bound the k^{th} segment.

Observe, for example, that Segment 5 is bounded at the p^{th} iteration by knots located at $t_{2,2}$ and $t_{3,2}$. Between iterations, though, the axis durations $\Delta t_{i,k}$ change values, thereby changing the positions of the knots at the next iteration. At iteration $p+1$, Segment 5 is bounded by $t_{3,1}$ and $t_{2,3}$. Completely different sets of variables now define the segment boundaries! Expressing the segment boundaries for Segment 5 in terms of $\boldsymbol{x}_t$, on the p^{th} iteration,

$$t_4(\boldsymbol{x}_t) \;=\; t_{2,2}(\boldsymbol{x}_t) = t_0 + |\Delta t_{2,1}| + |\Delta t_{2,2}|\,, \tag{52}$$

$$t_5(\boldsymbol{x}_t) \;=\; t_{3,2}(\boldsymbol{x}_t) = t_0 + |\Delta t_{3,1}| + |\Delta t_{3,2}|\,, \tag{53}$$

while on the $(p+1)^{\text{th}}$ iteration,

$$\begin{aligned} t_4(\boldsymbol{x}_t) &= t_{3,1}(\boldsymbol{x}_t) = t_0 + \left|\Delta t_{3,1}\right|, && (54) \\ t_5(\boldsymbol{x}_t) &= t_{2,3}(\boldsymbol{x}_t) = t_0 + \left|\Delta t_{2,1}\right| + \left|\Delta t_{2,2}\right| + \left|\Delta t_{2,3}\right|. && (55) \end{aligned}$$

Thus, the dependencies of segment boundaries continuously change throughout the optimization process. The term *segment-time switching* is used to refer not only to the switching aspect of the control, but more importantly, to the switching dependencies of a segment's time elements demonstrated here. Segment-time switching is the fundamental characteristic of the parameterization presented in this investigation.

1. *Implementation Considerations*

Significant complexities arise in the implementation of the segment-time switching algorithm. The NLP code relies on using gradients to determine a search direction to improve the current set of parameters. However, gradients become considerably difficult to calculate, either analytically or numerically, due to the switching dependencies.

For example, consider the dynamical constraints of Equations 30 and 31, which necessarily rely on the values of segment boundaries, as

$$h = \frac{t_k - t_{k-1}}{n_n - 1}. \tag{56}$$

However, since their dependencies on parameters in $\boldsymbol{x}$ change upon each iteration, the evaluation of the constraint gradients, $\dfrac{\partial \boldsymbol{c}(\boldsymbol{x})}{\partial \boldsymbol{x}}$, becomes nontrivial. If gradients are calculated analytically, the algorithm must identify the current dependencies for segment boundaries. This begins when boundaries are placed in chronological order. The sorting algorithm updates and identifies which value of $t_{i,k}$ corresponds to each value of t_k. The extent of the actual dependencies on parameters in $\boldsymbol{x}$ cascades from there.

Alternatively, gradiently may be calculated numerically using finite differences. Depending on the NLP algorithm employed, this may be an automated process. However, a user must exercise care to ensure that the finite differences are evaluated for *all possible* dependencies on each iteration. With many parameters, this process consumes a significant amount of time on each iteration, thus slowing the solution process dramatically. A solution derived in five minutes using analytical derivatives may require up to twenty-four hours with numerical derivatives.

2. *Transcriptions with Multiple Spacecraft*

The above results are now extended for applications to spacecraft formations. Let n_d designate the number of deputy spacecraft in the formation. In this case, there are n_y states per spacecraft (6, in the case of 3-dimensional dynamics), n_u controls per spacecraft (≥ 3, in the case of 3-axis control), n_n nodes per segment, n_k interior knots per control axis (control switches), $n_s = n_u n_k + 1$ segments over which the dynamics must be satisfied, and n_d deputy spacecraft. The resulting parameter optimization problem has n variables and n_c constraints.

For each spacecraft, and for each node in each segment, a variable that represents each state is required, thus contributing $n_y n_n n_s n_d$ variables. Similarly, a variable representing the duration between control switches in each axis is necessary, contributing $n_u(n_k+1)n_d$ variables to the parameter vector. The initial and final times contribute two additional variables. Thus, the number of variables required for this formulation is, $n = n_y n_n n_s n_d + n_u(n_k+1)n_d + 2$. The parameter vector is subsequently defined as,

$$\boldsymbol{x} = \left[\cdots\ y_{i,j,k,l}\ \cdots\ \cdots\ \Delta t_{i,k,l}\ \cdots\ t_0\ t_f\right]^T, \tag{57}$$

where the state indices are $i = 1, \ldots, n_y$, $j = 1, \ldots, n_n$, $k = 1, \ldots, n_s$, and $l = 1, \ldots, n_d$, and the time indices are $i = 1, \ldots, n_u$, $k = 1, \ldots, n_k + 1$, and $l = 1, \ldots, n_d$. Again in this formulation, the control accelerations are not considered variables in the parameter vector. Instead, the control values are pre-designated within the set $\mathcal{U} = \{-T^*, 0, T^*\}$ and are assigned specific values between switching times. The optimization variables associated with the control vector, are simply the axis durations, $\Delta t_{i,k,l}$, which imply the times (knots) when the control switches occur.

The constraint vector for this formulation is the same as that listed in Equation 44, except there are multiple time constraints, $\boldsymbol{c}_t$. The time restriction must hold for each axis and for each spacecraft, so $n_t = n_u n_d$. The total number of constraints in $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0}$ is

$$n_c = n_{\psi_0} + n_{\psi_f} + n_\beta n_n n_s n_d + n_y(n_n - 1)n_s n_d + n_y(n_s - 1)n_d + n_u n_d. \tag{58}$$

The sizes for the initial and final conditions, $\boldsymbol{c}_{\psi_0}$ and $\boldsymbol{c}_{\psi_f}$, are n_{ψ_0} and n_{ψ_f}, respectively. These dimensions are likely increased relative to the previous formulation since there are now n_d spacecraft. The path constraints, $\boldsymbol{c}_\beta$, are imposed at every node, along each segment, and for each spacecraft. The vector is sized accordingly. The dynamic constraints, $\boldsymbol{c}_{\dot{y}}$, are now of size $n_y(n_n - 1)n_s n_d$ to enforce continuity between the nodes for every state of every spacecraft along each segment. Continuity is enforced at the knots in $\boldsymbol{c}_s$ by $n_y(n_s - 1)n_d$ constraints for every state of every spacecraft. Notice that the number of transitions between segments equals the number of total knots, $n_s - 1 = n_u n_k$. Finally, there are $n_u n_d$ time constraints contained in $\boldsymbol{c}_t$ to ensure that all axis durations sum to $t_f - t_0$.

III. Applications

The switching segment formulation is tailored to determine feasible and locally optimal spacecraft trajectories for highly constrained formations. Considering deep-space imaging formations as the application, it is reasonable to assume that constraints may be imposed on

- the size, shape, and orientation of the formation,
- the orientation of each member of the formation (deputy spacecraft),
- the thruster capability and thruster location on each spacecraft body.

Formationkeeping in the presense of all these constraints is naturally a difficult task. For example, although a specified thrust magnitude can be limiting by itself, when the thrust direction is also restricted by requirements on spacecraft orientation, precision control is *nearly* impossible. Certainly, not all of these constraints can be met simultaneously. Thus, the optimal control problem seeks to minimize deviations when constraints cannot feasibly be enforced.

A. Formation Constraints

Define the scalar value $r^*_{cd_l}$ to be the required radial distance between the chief spacecraft (reference origin) and the l^{th} deputy spacecraft. In addition, the scalar value $r^*_{d_\lambda d_l}$ is defined as the generalized required distance between deputy λ and deputy l. Recall the notation of relative states for the l^{th} deputy as

$$\boldsymbol{y}_l = \begin{bmatrix} \boldsymbol{r}_l \\ \boldsymbol{v}_l \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{d_l} - \boldsymbol{r}_c \\ \boldsymbol{v}_{d_l} - \boldsymbol{v}_c \end{bmatrix}. \tag{59}$$

Let $\boldsymbol{r}_{\lambda l} = \boldsymbol{r}_{d_l} - \boldsymbol{r}_{d_\lambda}$ such that the formation constraints are of the form

$$\boldsymbol{r}_l^T \boldsymbol{r}_l - \left(r^*_{cd_l}\right)^2 = 0, \quad l = 1, ..., n_d \tag{60}$$

$$\boldsymbol{r}_{\lambda l}^T \boldsymbol{r}_{\lambda l} - \left(r^*_{d_\lambda d_l}\right)^2 = 0, \quad l = 1, ..., n_d - 1, \quad \lambda = l + 1, ..., n_d \tag{61}$$

Furthermore, let $\boldsymbol{r}^*_{cs}$ define a vector pointing from the chief to a specified target point in space. If the target is sufficiently distant from the spacecraft, $\boldsymbol{r}^*_{cs}$ essentially points along an inertially fixed direction such that $\boldsymbol{r}^{*\mathcal{I}}_{cs} =$ constant. This is not a necessary restriction, however. In general, $\boldsymbol{r}^*_{cs}$ is written in the rotating frame as a function of time: $\boldsymbol{r}^{*\mathcal{R}}_{cs} = \boldsymbol{r}^{*\mathcal{R}}_{cs}(t)$. For imaging a distant point, the plane of the formation is assumed perpendicular to $\boldsymbol{r}^*_{cs}$, that is,

$$\boldsymbol{r}_{\lambda l}^T \boldsymbol{r}^*_{cs} = 0, \quad l = 1, ..., n_d - 1, \quad \lambda = l + 1, ..., n_d. \tag{62}$$

In Figure 6, both the formation size and orientation requirements are illustrated. Here, $n_d = 3$, and specified formation distances the same for each deputy force the formation to appear as an equilateral triangle. The formation may take on many planar shapes with enough deputy spacecraft and through proper definition of the formation distances.

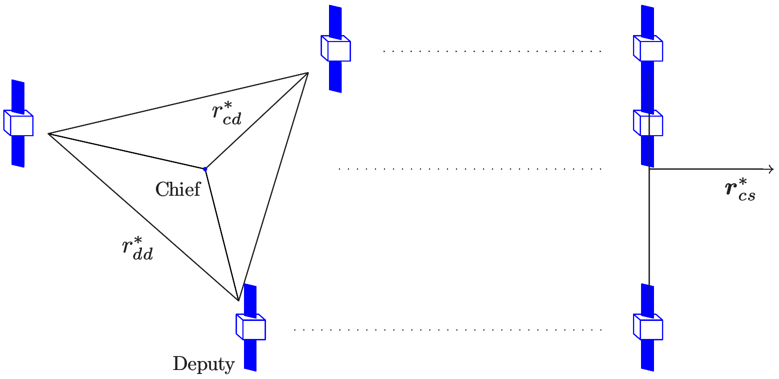

Figure 6. Formation Pointing

B. Control Constraints with Specified Spacecraft Orientation

1. *Spacecraft Orientation*

Let the orientation of the l^{th} deputy spacecraft be defined by the body-fixed coordinate frame,

$$\mathcal{B}_l \equiv \{\hat{\boldsymbol{x}}_{\mathcal{B}_l}, \hat{\boldsymbol{y}}_{\mathcal{B}_l}, \hat{\boldsymbol{z}}_{\mathcal{B}_l}\} \tag{63}$$

where the unit vectors, $\hat{\boldsymbol{x}}_{\mathcal{B}_l}$, $\hat{\boldsymbol{y}}_{\mathcal{B}_l}$, and $\hat{\boldsymbol{z}}_{\mathcal{B}_l}$ can be written in terms of any coordinate system. Most likely, they are expressed in the frame in which the states are integrated, i.e. the CR3BP rotating frame, $\mathcal{R}$.

For the subset of interferometry missions considered here, each deputy spacecraft points along a specified direction for the duration of the mission or phase of interest. Therefore, define the first axis of the body-fixed frame in the direction of the imaged point.

$$\hat{\boldsymbol{x}}_{\mathcal{B}_l} \equiv \frac{\boldsymbol{r}_{cs}}{r_{cs}} \tag{64}$$

Note that the scalar $r_{cs} = |\boldsymbol{r}_{cs}|$ normalizes the unit vector. This axis is normal to the face of the spacecraft containing an imaging payload. Now define the other coordinate axes as,

$$\hat{\boldsymbol{y}}_{\mathcal{B}_l} \equiv \frac{\boldsymbol{r}_l \times \hat{\boldsymbol{x}}_{\mathcal{B}_l}}{|\boldsymbol{r}_l \times \hat{\boldsymbol{x}}_{\mathcal{B}_l}|}, \tag{65}$$

$$\hat{\boldsymbol{z}}_{\mathcal{B}_l} \equiv \hat{\boldsymbol{x}}_{\mathcal{B}_l} \times \hat{\boldsymbol{y}}_{\mathcal{B}_l}. \tag{66}$$

The vector $\hat{\boldsymbol{y}}_{\mathcal{B}_l}$ is roughly parallel to the direction of motion, and $\hat{\boldsymbol{z}}_{\mathcal{B}_l}$ is approximately aligned with the radial distance from the chief. This statement is based on the assumption of equal spacing among deputies. Note, however, that this coordinate system definition is valid even if the requirements specified by r^*_{cd} and r^*_{dd} are not met.

2. *Spacecraft Design*

For simplicity, assume that each deputy has a cube-shaped structure and is equipped with a thruster on each side, where the body frame, $\mathcal{B}_l$, is aligned with the principal axes of the spacecraft. The attitude of each spacecraft is independently controlled to maintain the proper alignment with the target.

It is not the intent of this work to suggest requirements on the structural design of a spacecraft. The assumptions made here are generalized for the problem under consideration. As a result, it is assumed the vehicles are capable of thrusting in all three body-fixed directions (although not necessarily at the same magnitudes). This configuration represents a simple way of implementing the thrusting requirements for the selected example. It is also one of the most limiting implementations. The method used in this work is generalized to consider other structural characteristics, but this implementation is a logical way of demonstrating the methods and the effects that highly constrained spacecraft orientations may have on performance.

3. *Control Constraints*

Consider the vector $\boldsymbol{u}_l$ that defines the control acceleration for the l^{th} spacecraft. In the body-frame,

$$\boldsymbol{u}_l^{\mathcal{B}} \equiv \begin{bmatrix} u_{\hat{\boldsymbol{x}}_{\mathcal{B}_l}} \\ u_{\hat{\boldsymbol{y}}_{\mathcal{B}_l}} \\ u_{\hat{\boldsymbol{z}}_{\mathcal{B}_l}} \end{bmatrix}, \tag{67}$$

and the components of the vector are constrained to the values in the set $\mathcal{U} \equiv \{-T^*, 0, T^*\}$ as in Equation 46. This constraint must be imposed at all times, as this represents the true thrusting capability of the spacecraft.

C. The Objectives

In this investigation, the control constraints of Section B are imposed by the problem formulation. That is, thrusting solutions only include finite burns at the specified magnitude and in the specified directions. Thus, the formation constraints listed in Section A cannot be perfectly met. However, the objective function is selected to penalize deviations along the trajectory arc over $t \in [t_0\ t_f]$. In addition, a cost is assigned to the duration of all burn segments to minimize the effort (thrusting) over the course of the trajectory. In general the cost function weighs three separate sets of costs.

$$J = w_1 J_1 + w_2 J_2 + w_3 J_3 \tag{68}$$

The weights, w_i, are selected to balance the costs of fuel and formation deviations. The cost indices are formulated as integrals of the following form:

$$J_1 = \sum_{l=1}^{n_d} \int_{t_0}^{t_f} \boldsymbol{u}_l^T \boldsymbol{u}_l \, dt \tag{69}$$

$$J_2 = \sum_{l=1}^{n_d} \int_{t_0}^{t_f} \left(\boldsymbol{r}_l^T \boldsymbol{r}_l - \left(r^*_{cd_l}\right)^2\right)^2 dt + \sum_{l=1}^{n_d-1} \sum_{\lambda=l+1}^{n_d} \int_{t_0}^{t_f} \left(\boldsymbol{r}_{\lambda l}^T \boldsymbol{r}_{\lambda l} - \left(r^*_{d_\lambda d_l}\right)^2\right)^2 dt \tag{70}$$

$$J_3 = \sum_{l=1}^{n_d-1} \sum_{\lambda=l+1}^{n_d} \int_{t_0}^{t_f} \left(\boldsymbol{r}_{\lambda l}^T \boldsymbol{r}_{cs}\right)^2 dt \tag{71}$$

$$\tag{72}$$

The cost, J_1, is a quadratic cost on the thrust; J_2 minimizes deviations in formation size and shape; and J_3 minimizes deviations in the formation plane. When input as a function of the parameters $\boldsymbol{x}$, the cost function takes the form,

$$F(\boldsymbol{x}) = w_1 F_1(\boldsymbol{x}) + w_2 F_2(\boldsymbol{x}) + w_3 F_3(\boldsymbol{x}). \tag{73}$$

The implementation of the parameterized subcosts follow.

1. *Thrust*

The control in each axis of each spacecraft is broken into $n_k + 1$ axis segments, whose durations are $\Delta t_{i,k,l}$. In addition the controls are already known for each axis segment: they can be expressed conveniently as

$$u_{i,k,l} = T^* \cos\left(\frac{\pi}{2}(k-1)\right), \tag{74}$$

where $i = 1, \ldots, n_u$, $k = 1, \ldots, n_k + 1$, and $l = 1, \ldots, n_d$. Thus the control cost is simply,

$$F_1(\boldsymbol{x}) = \sum_{l=1}^{n_d} \sum_{k=1}^{n_k+1} \sum_{i=1}^{n_u} \Delta t_{i,k,l} u_{i,k,l}^2. \tag{75}$$

2. Formation Size and Shape Deviation

For the formation costs, added complexities stem from the node distribution from t_0 to t_f. Specifically, the node distribution varies for each spacecraft, yet the deviation in the shape of the formation depends on the position of each spacecraft at a given time. A simple solution is to perform an interpolation of the nodes for each spacecraft. Using interpolated values, a representative number of points over the trajectory is selected to approximate the integrals of Equation 70. Define the interpolated set of states as $\boldsymbol{y}_{j,l}$ where $j = 1, ..., n_g$, and n_g is sufficiently large to generate a uniformly spaced grid of points that describe the trajectories of the $l = 1, ..., n_d$ spacecraft. The interpolated states are distinguished from those in Equation 57 by not having a k subscript, as they are uniformly distributed between t_0 and t_f independent of knot locations. Then

$$\begin{aligned} F_2(\boldsymbol{x}) \quad = \quad & \sum_{l=1}^{n_d} \sum_{j=1}^{n_g-1} \frac{t_{j+1}-t_j}{2} \left\{ \left(\left[\sum_{i=1}^{3} y_{i,j,l}^2 \right] - \left(r_{cd_l}^*\right)^2 \right)^2 + \left(\left[\sum_{i=1}^{3} y_{i,j+1,l}^2 \right] - \left(r_{cd_l}^*\right)^2 \right)^2 \right\} \\ & + \sum_{l=1}^{n_d-1} \sum_{\lambda=l+1}^{n_d} \sum_{j=1}^{n_g-1} \frac{t_{j+1}-t_j}{2} \left\{ \left(\left[\sum_{i=1}^{3} \left(y_{i,j,l} - y_{i,j,\lambda}\right)^2 \right] - \left(r_{d_\lambda d_l}^*\right)^2 \right)^2 \right. \\ & \left. + \left(\left[\sum_{i=1}^{3} \left(y_{i,j+1,l} - y_{i,j+1,\lambda}\right)^2 \right] - \left(r_{d_\lambda d_l}^*\right)^2 \right)^2 \right\} \end{aligned} \tag{76}$$

computes deviations from specified distances between chief-deputy and deputy-deputy pairs.

3. Formation Plane Deviation

The cost associated with deviations in the formation plane exhibits the same complexities as the formation shape cost. Once again, this difficulty is overcome through the use of interpolated points, $\boldsymbol{y}_{j,l}$. The resulting cost index is

$$\begin{aligned} F_3(\boldsymbol{x}) \quad = \quad & \sum_{l=1}^{n_d-1} \sum_{\lambda=l+1}^{n_d} \sum_{j=1}^{n_g-1} \frac{t_{j+1}-t_j}{2} \left\{ \left(\left[\sum_{i=1}^{3} \left(y_{i,j,l} - y_{i,j,\lambda}\right) r_{(cs)_i}^*(t_j) \right] \right)^2 \right. \\ & \left. + \left(\left[\sum_{i=1}^{3} \left(y_{i,j+1,l} - y_{i,j+1,\lambda}\right) r_{(cs)_i}^*(t_{j+1}) \right] \right)^2 \right\}. \end{aligned} \tag{77}$$

This cost function uses a trapezoidal approximation of the integral of Equation 71 to accumulate penalties in plane deviation.

D. Sample Solutions

A candidate formation consists of $n_d = 3$ deputy spacecraft, equally spaced as in Figure 6 with distance $r_{cd}^* = 1$ km between the chief and each deputy and $r_{dd}^* = 1.73$ km between two deputies. The formation seeks to collect data on a distant star located along the inertial pointing direction, $\boldsymbol{r}_{cs}^{*\mathcal{I}} = [1\ 0\ 0]^T$. Solutions are found with several sets of parameters. A baseline solution is found with $n_n = 4$ nodes per segment and $n_k = 10$ interior switch times per control axis, allotting for six thrusting segments per control axis per satellite (three positive, three negative). A trajectory is devised with fixed initial states, and final conditions dictate the formation size and plane constraints be met at the final time. Since the CR3BP is time invariant, the initial time is set as $t_0 = 0$. The formation nominally rotates about the reference with the same period as the Halo orbit. By fixing the final time, t_f, at a third of the reference Halo orbit period (approximately 5.1183×10^6 seconds), the three deputies each complete a third of a rotation about the chief.

1. The Initial Guess

In numerical optimization problems, the initial guess, $\boldsymbol{x}_0$, is often a determining factor in whether an optimal solution is successfully identified. A gradient-based numerical algorithm can only be trusted to find *locally* optimal solutions, most likely in the vicinity of the starting point. Thus, identical problems can easily

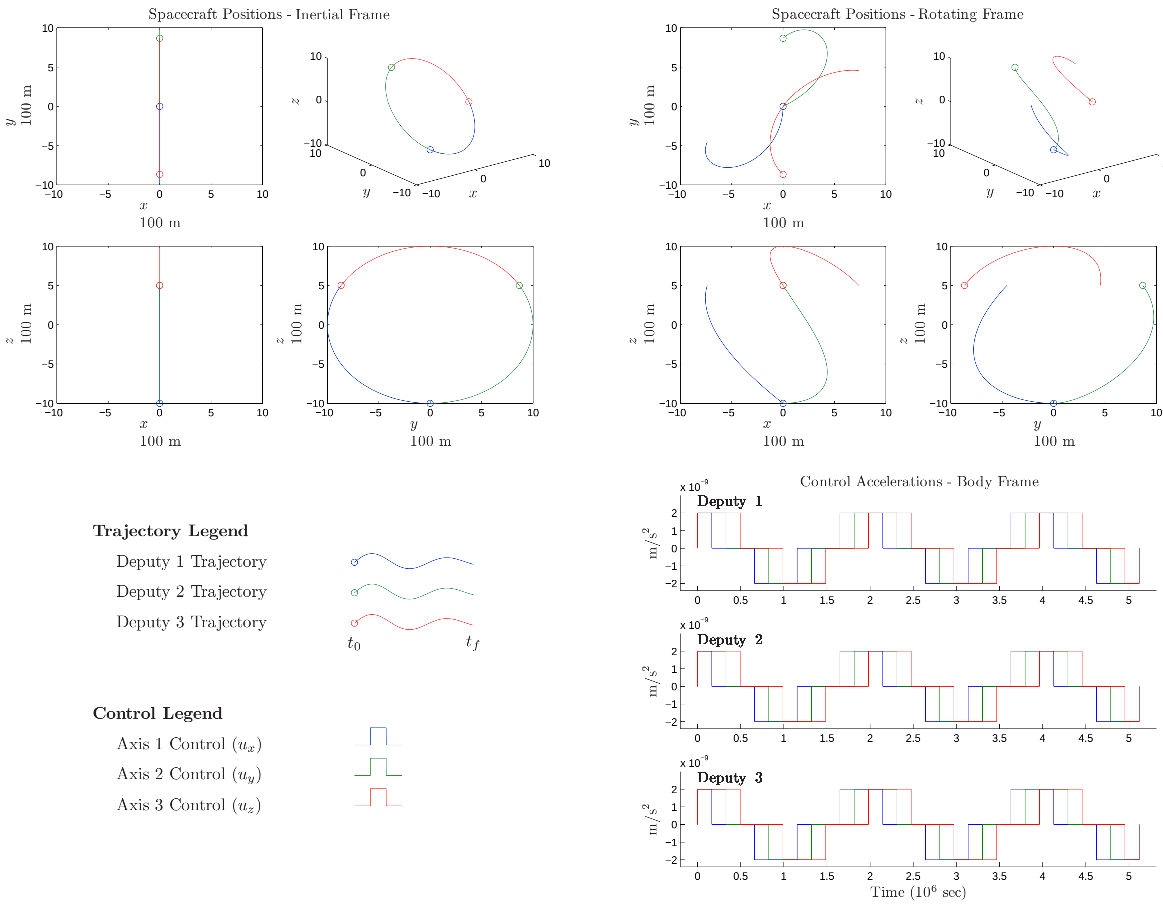

Figure 7. Baseline Initial Guess

converge on different solutions if starting from distant points. One way to manage this characteristic is to use several different values of $\boldsymbol{x}_0$ to find solutions, choosing the minimum of the locally optimal solutions as the globally optimal solution. For this effort, the successful identification of any feasible and locally optimal solution to the highly constrained formation problem demonstrates the capability of the formulation presented in Section II.

Collocation methods, where discretized states are included as parameters in the optimization, allow the user to select an initial set of parameters that *guides* the optimization process towards a particular trajectory. Although $\boldsymbol{x}_0$ does not satisfy the dynamical constraints, its states may represent the vicinity of the desired solution. Through iteration, the optimizer modifies the current point until feasibility and optimality tolerances are achieved. In this mindset, a baseline initial guess for this candidate formation appears in Figure 7, depicting the trajectories of three spacecraft in the inertial and rotating frames, along with a nominal control profile of the three spacecraft in their respective body frames. As seen in the inertial frame, the trajectories satisfy the formation requirements (size and plane orientation) as presented above, maintaining a plane perpendicular to the inertial pointing direction, $\boldsymbol{r}_{cs}^{*\mathcal{I}}$. This set of states represent a perfectly circular planar formation. Ideally, an unconstrained control profile is capable of delivering the associated accelerations. The constrained controls shown here, however, do not produce these states. Instead, the burn durations chosen for the initial guess simply distribute the time of each of the $n_u n_k + 1 = 31$ segments equally. This choice, while somewhat arbitrary, frees the optimizer to search the solution space for the proper durations of burning and coasting segments, while keeping the states fairly close to their ideal placement.

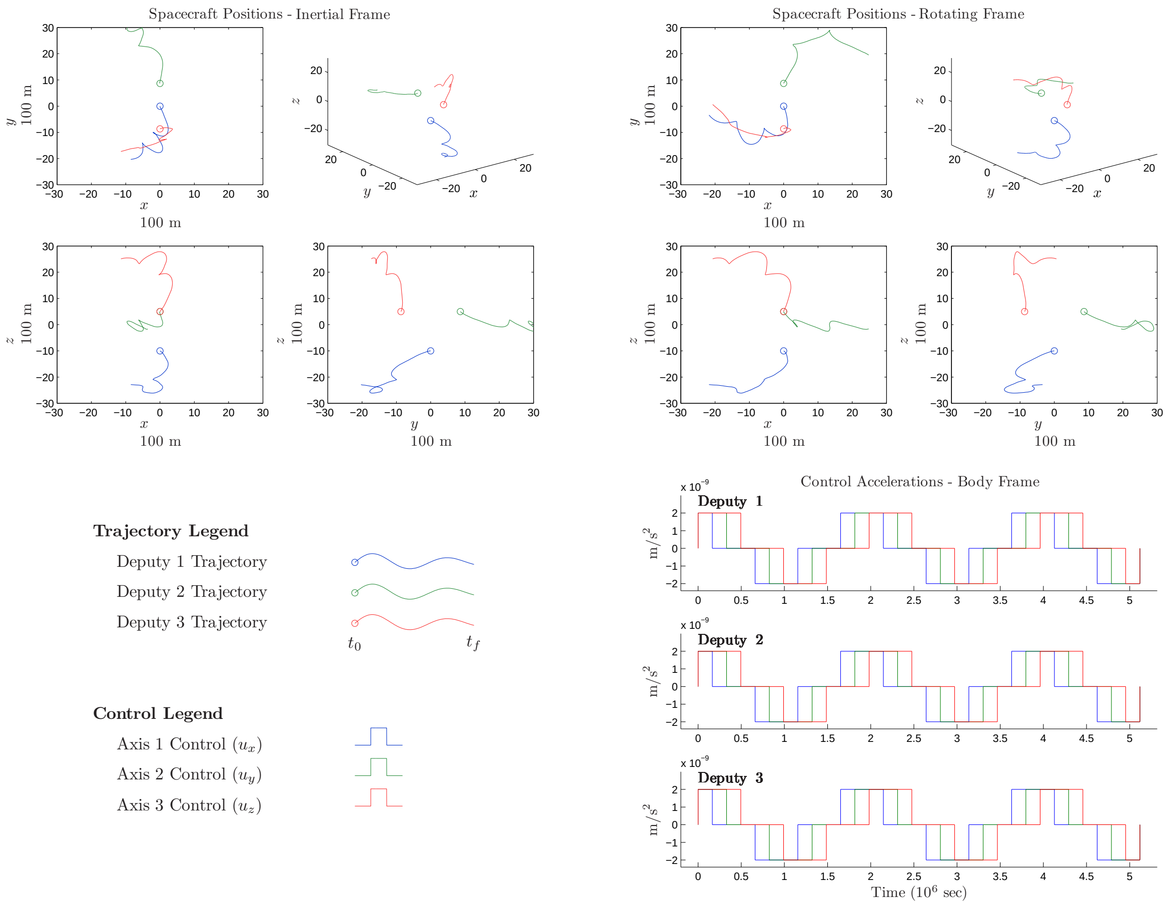

Figure 8. Feasible Initial Guess

As an alternative, an initial guess can be generated that is feasible by propagating a guessed control profile, say, that of the baseline guess. In this case, the initial guess satisfies the dynamical constraints immediately, but the formation constraints are clearly not satisfied. In addition, deviations from the desired formation characteristics are large. Thus, the cost is also large at the first iteration of the optimization procedure. In Figure 8, a set of states is shown that is the true propagation of the arbitrary control profile used in the baseline guess. It is apparent that the trajectories of three spacecraft driven by this control profile are far from meeting any formation requirements.

Although this is a dramatic example, the difference between the baseline and feasible initial guess communicates an advantage to using a collocation method over an indirect or direct shooting method. Shooting methods essentially "shoot" a guess for the control to the terminal time, making corrections on each iteration to meet constraints. However, because of the extreme sensitivities of this dynamic regime, the quality of a shooting guess is critical to the success of the process. A collocation method is far more flexible in the initial guess.

2. Baseline Solution

The baseline initial guess is optimized via the fomulation presented in this work to yield the solution depicted in Figure 9. The trajectories of the three spacecraft are shown in the inertial and rotating reference frames, along with the control profile for each of the spacecraft.

First, it is observed that the optimization has modified the guessed segment durations to specify proper burn times and burn durations. Burn segments are shortened and coast segments are lengthened according to feasibility constraints (driving dynamic constraint residuals to zero) and optimality considerations (minimizing fuel expenditure). Notice also that several burn segments have zero duration, depicted as a single line

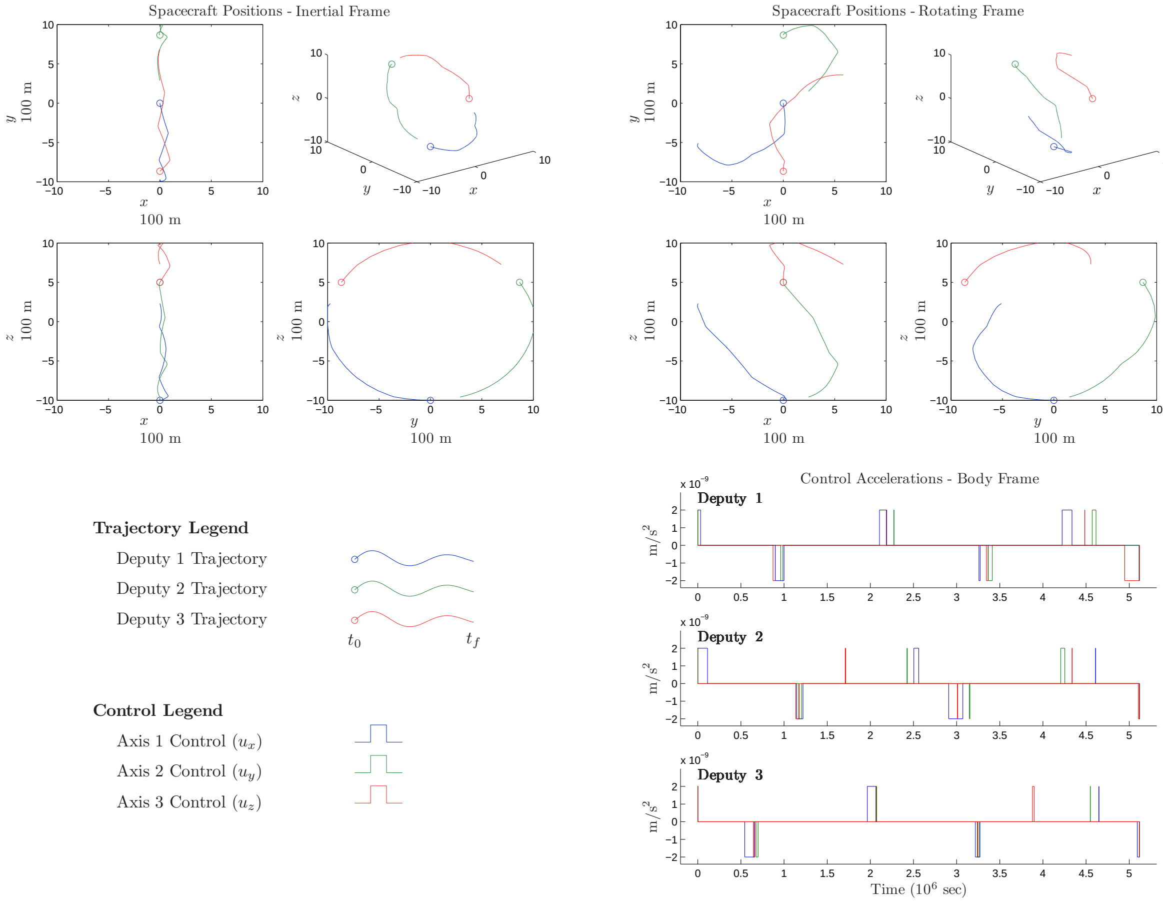

Figure 9. Baseline Solution

separating two coasting arcs. Recall that the formulation pre-specifies thrusting arcs. However, if a solution does not require that thrusting arc, its duration is reduced to zero in the optimization. The structure of the solution does not change, and a segment still exists in that place (occurring over zero time). Additionally, n_n nodes exist at a single point in time. Since the solution implies the given burn segment is superfluous, many of the optimization variables have no effect (these are the states along the zero segment). However, since no feasible control profile was known *a priori*, it is helpful to include *more* segments than ultimately necessary until some insight into a feasible trajectory is extracted.

Next, it is apparent that the paths of each of the deputy spacecraft resemble the initial guess to a significant extent. Certainly, deviations away from the "perfect" trajectory are visible, but the solution implies that those deviations are necessary in order to identify a feasible trajectory given the specified control restrictions. More importantly, the trajectory now coincides with the control history. For example, the control discontinuities between segments result in corners in the velocity states for each spacecraft (not shown).

One can also extract from the baseline solution an optimal rotation rate for the formation. By comparing the inertial trajectories of the baseline guess and solution, it is apparent in the optimal solution that the spacecraft traverse less distance over the time interval. This suggests that for a 1 km (radius) formation, the optimal rotation rate is slightly less than the baseline guess of 2π/period.

The baseline solution demonstates the effectiveness of the solution technique on a highly constrained optimal control problem. The sample mission requires fixed spacecraft orientation, fixed thruster locations, and a specified thrust acceleration magnitude. Under these constraints, trajectories are sought to maintain formation size, shape, and orientation restrictions. An unconstrained control solution that maintains the formation configuration requires acceleration levels below those deliverable by existing technology. In the

Table 1. Comparison of Solutions with Various Parameters

	Baseline	t_f	n_k, t_f	Feasible Guess	Thrust
n_n	4	4	6	4	4
n_k	10	10	20	10	10
t_f (10^6 sec)	5.1183	10.2366	10.2366	5.1183	5.1183
Guess	Baseline	Baseline	Baseline	Feasible	Baseline
w_{Thrust}	$\frac{1}{400}$	$\frac{1}{400}$	$\frac{1}{400}$	$\frac{1}{400}$	$\frac{1}{1600}$
w_{Distance}	0.1	0.1	0.1	0.1	0.1
w_{Plane}	1	1	1	1	1
Thrust (km/s^2)	2.0e-12	2.0e-12	2.0e-12	2.0e-12	4.0e-12
n	2333	2333	6779	2333	2333
# Iterations	45	157	194	95	272
Computational Time (sec)	253.91	956.41	3000.35	575.88	1570.04
Weighted Thrust Cost	1.6233	5.8133	6.6838	10.9247	1.1324
Weighted Formation Cost	16.1818	634.3255	67.7092	35.5675	6.9286
Weighted Plane Cost	0.8591	84.6949	4.6852	4.9035	1.0632
Total Cost	18.6643	724.8338	79.0782	51.3956	9.1242

actuator constrained case, the proposed methodology determines reasonable performance expectations in the presense of these system constraints.

3. Varying Parameters to Obtain Different Solutions

The methodology presented in this formulation serves to establish reasonable expectations for libration point missions under the constraints imposed by the current state of technology. The generalized method can be applied to any number of potential scenarios during the mission planning process. To demonstrate the flexibility of the method, some formulation parameters are changed here to observe their effect on the solution. The final time, the number of nodes and knots, the initial guess, and the thrust magnitude are varied to determine four other solutions to be compared with the baseline in Table 1. The results vary in size, solution convergence, and cost performance.

By doubling the fixed final time (now two thirds of an orbital period, or 10.2366×10^6 seconds) without any other changes, it is evident that the cost is significantly increased from the baseline. Of course, it would be reasonable to expect the integral costs to be proportional to the final time. However, the solution is over an order of magnitude greater in cost. That is because for a longer trajectory, a good solution necessarily requires more maneuvers, and, subseqently, more knots (switching opportunities) in the problem formulation.

The next example maintains the longer final time, now including twice as many knots as the baseline. In addition the number of nodes is increased to improve the fidelity of the solution. As a result, the dimension of the parameter optimization problem increases significantly (from 2333 to 6779 variables). This requires additional computational time for each iteration and an overall longer convergence time. However, including more knots does lead to an improved solution relative to the previous example.

The feasible initial guess is also compared in performance to the baseline initial guess (using the original final time and numbers of nodes and knots). In contrast to the baseline, the feasible guess does not lead to an improved solution. Clearly, the baseline guess is in the vicinity of a smaller locally optimal point than the feasible guess. This is expected since the feasible initial guess does not produce trajectories that meet the formation requirements.

Finally, a solution is identified using a larger thrust magnitude than the baseline. Although the total cost is less than the baseline, this takes into account a different weight on the thrust portion of the cost index. The weight is selected to normalize the cost, dividing away the square of the thrust magnitude (in scaled distance and time units). Each of the thrust acceleration magnitudes and cost function weights in this investigation are selected arbitrarily. However, it is clear that these parameters can be tailored to values that effectively represent a realistic mission scenario.

IV. Conclusions

In this investigation, a method is presented for determining locally optimal control solutions for libration point formations restricted to highly constrained actuation requirements. The process employs a collocation (direct optimization) method with some unique modifications that guarantee solutions attained are physically realizable with existing thruster technology. Modifications are based upon a segmented formulation with switching sets of time dependencies. In this formulation, the optimizer determines precise switching times for thrusters independently, resulting in an optimal actuation schedule.

Typically, collocation solutions are only approximations to continuous solutions. Generally, the numerical solution consists of control values at nodes only, and values in between nodes are interpolated. However, the control solutions presented in this investigation require no interpolation! The value of the control is already known, and the optimizer determines the exact time to switch control values to the accuracy specified by the user. Thus, numerical control solutions are immediately implementable with a continuous system.

The solutions presented demonstrate the method's capability to minimize selected costs when determining a feasible trajectory. A number of optimization parameters are selected to indicate the level of accuracy required in the solution. In addition, a user specifies many relevant inputs, including

- formation configuration, size, orientation, and rotation rate,
- thruster capability and placement,
- dynamic model and reference trajectory, and
- initial and terminal conditions.

The generalized formulation allows this method to be relevant for many future applications. This methodology is an effective mission planning tool for establishing requirements and capabilities for highly constrained formations.

References

[1] Howell, K.C. and Keeter, T.M. "Station-Keeping Strategies for Libration Point Orbits: Target Point and Floquet Mode Approaches." *Advances in the Astronautical Sciences*, 89(2), 1995, pp. 1377-1396.

[2] Scheeres, D.J. and Vinh, N.X. "Dynamics and Control of Relative Motion in an Unstable Orbit," AIAA Paper 2000-4235, Aug 2000.

[3] Luquette, R.J. and Sanner, R.M. "A non-linear approach to spacecraft formation control in the vicinity of a collinear libration point." AAS/AIAA Astrodynamics Conference, 30 Jul-2 Aug 2001 (Quebec, Canada: AAS), AAS Paper 01-330.

[4] Gurfil, P., Idan, M., and Kasdin, N.J. "Adaptive Neural Control of Deep-Space Formation Flying." American Control Conference, 8-10 May 2002 (Anchorage, AK: ACC), pp. 2842-2847.

[5] Scheeres, D.J., Hsiao, F.-Y., and Vinh, N.X. "Stabilizing Motion Relative to an Unstable Orbit: Applications to Spacecraft Formation Flight." *Journal of Guidance, Control, and Dynamics*, 26(1), Jan-Feb 2003, pp. 62-73.

[6] Gurfil, P., Idan, M., and Kasdin, N.J. "Adaptive Neural Control of Deep-Space Formation Flying." *Journal of Guidance, Control, and Dynamics*, 26(3), May-Jun 2003, pp.491-501.

[7] Gurfil, P. and Kasdin, N.J. "Stability and control of spacecraft formation flying in trajectories of the restricted three-body problem." *Acta Astronautica*, 54, 2004, pp. 433-453.

[8] Marchand, B.G. and Howell, K.C. "Formation Flight Near L_1 and L_2 in the Sun-Earth/Moon Ephemeris System Including Solar Radiation Pressure." In *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, Aug, 2003. AAS Paper 03-596.

[9] Howell, K.C. and Marchand, B.G. "Design and Control of Formations Near the Libration Points of the Sun-Earth/Moon Ephemeris System." In *Proceedings of the Space Flight Mechanics Symposium - Goddard Space Flight Center*, Greenbelt, MD, Oct 2003.

[10] Marchand, B.G. and Howell, K.C. "Aspherical Formations Near the Libration Points in the Sun-Earth/Moon Ephemeris System." AAS/AIAA Space Flight Mechanics Meeting, 7-12 Februray 2004 (Maui, HI: AAS), AAS Paper 04-157.

[11] Howell, K.C. and Marchand, B.G. "Formations Near the Libration Points: Design Strategies Using Natural and Non-Natural Arcs." In *Proceedings of GSFC* 2^{nd} *International Symposium on Formation Flying Missions and Technologies*, Greenbelt, MD, Sep 2004.

[12] Marchand, B.G. Spacecraft Formation Keeping Near the Libration Points of the Sun-Earth/Moon System. PhD Dissertation. Purdue University. August 2004.

[13] Howell, K.C. and Marchand, B.G. "Natural and Non-Natural Spacecraft Formations Near the L_1 and L_2 Libration Points in the Sun-Earth/Moon Ephemeris System." *Dynamical Systems: An International Journal, Special Issue*: Dynamical Systems in Dynamical Astronomy and Space Mission Design, 20(1), Mar 2005, pp. 149-173.

[14] Marchand, B.G. and Howell, K.C. "Control Strategies for Formation Flight in the Vicinity of the Libration Points." *Journal of Guidance, Control, and Dynamics*. 28(6), Nov-Dec 2005, pp. 1210-1219.

[15] Marchand, B.G., Howell, K.C., and Betts, J.T. "Discrete Nonlinear Optimal Control of S/C Formations Near the L_1 and L_2 Points of the Sun-Earth/Moon System." AAS/AIAA Astrodynamics Specialists Conference, Lake Tahoe, CA, August, 2005.

[16] Infeld, S.I., Josselyn, S.B., Murray, W., and Ross, I.M. "Design and Control of Libration Point Spacecraft Formations." *Journal of Guidance, Control, and Dynamics*, 30(4), Jul-Aug 2007, pp. 899-909.

[17] Mueller, J. "Thruster Options for Microspacecraft: a Review and Evaluation of Existing Hardware and Emerging Technologies." 33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 6-9 July 1997 (Seattle, WA: AIAA), AIAA Paper 97-3058.

[18] Gonzales, A.D. and Baker, R.P. "Microchip Laser Propulsion for Small Satellites." 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 8-11 July 2001 (Salt Lake City, UT: AIAA), AIAA Paper 2001-3789.

[19] Gonzales, D.A., and Baker, R.P. "Microchip Laser Propulsion for Small Satellites." *Proceedings of the 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Salt Lake City, Utah, Jul 2001. AIAA Paper 2001-3789.

[20] Gonzales, A.D. and Baker, R.P. "Micropropulsion using a Nd:YAG microchip laser." *Proceedings of SPIE - The International Society for Optical Engineering*, 4760(ii), 2002, p 752-765.

[21] Phipps, C. and Luke, J. "Diode Laser-Driven Microthrusters–A New Departure for Micropropulsion. *AIAA Journal*, 40(2), 2002, pp. 310-318.

[22] Phipps, C., Luke, J., and Helgeson, W. "Laser space propulsion overview." *Proceedings of SPIE - The International Society for Optical Engineering*, 6606, *Advanced Laser Technologies 2006, 2007*, p 660602.

[23] Coulter, D. "NASA's Terrestrial Planet Finder Missions." In *Proceedings of SPIE Vol. 5487*, Bellingham, WA, 2004, pp. 1207-1215.

[24] Lay, O.P. et al. "Architecture Trade Study for the Terrestrial Planet Finder Interferometer." In *Proceedings of SPIE Vol. 5905*, Bellingham, WA, 2005, 590502.

[25] Cash, W. and Gendreau, K. "MAXIM Science and Technology." In *Proceedings of SPIE Vol. 5491*, Bellingham, WA, 2004, pp. 199-211.

[26] Gill, Philip E., Murray, Walter, and Saunders, Michael A. "User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming." February 12, 2006.

[27] Howell, K.C. "Three-Dimensional, Periodic, 'Halo' Orbits." *Celestial Mechanics*, 32, pp.53-71, 1984.

[28] Hull, David G. *Optimal Control Theory for Applications.* New York: Springer-Verlag, 2003.

[29] Hull, David G. "Conversion of Optimal Control Problems into Parameter Optimization Problems." *Journal of Guidance, Control, and Dynamics*, 20(1), Jan-Feb 1997, pp. 57-60.

[30] Kirk, Donald E. *Optimal Control Theory: An Introduction.* Prentice Hall, Englewood Cliffs, NJ, 1970.