The Journal of the Astronautical Sciences, Vol. 57, No. 3, July–September 2009, pp. 607–632

Actuator Constrained Optimal Formation Keeping Near the Libration Points

B.G. Marchand[1] and S.A. Stanton[2]

Abstract

Spacecraft formations that evolve near the libration points of the Sun–Earth/Moon system are one option commonly considered for space-based interferometry applications. Although precision tracking is often envisioned as a requirement, earlier studies have already revealed that the dynamically sensitive nature of this region of space presents a number of unique challenges in this respect. For instance, physical hardware limitations, specifically related to the on-board actuators, can limit the attainable tracking accuracy. The goal of this investigation is to devise a numerical process that allows the designer to identify the maximum tracking accuracy achievable in the presence of actuator (e.g., thruster) constraints. This is accomplished through the application of direct optimization methods. A problem formulation is presented that effectively treats, within a nonlinear programming framework, problems with state and control discontinuities and problems involving temporally continuous but spatially discrete control variables. The results of this investigation lay the algorithmic foundation for future explorations involving multiple independent actuators each with its own set of constraints and switching dependencies.

Introduction

Although optimal control applications are common in spacecraft mission design, complex actuator constraints, and their impact on the success of the mission, are not often considered during the mission design process. Spacecraft actuators, such as thrusters, that are subject to stringent physical limitations can adversely affect the attainability of mission goals. This is particularly true for spacecraft that evolve near the libration points of a multi-body system where dynamical sensitivities amplify the overall effect of actuation errors. In incorporating actuator constraints into a control problem, numerical optimization methods are most commonly

[1]Assistant Professor, Department of Aerospace Engineering, The University of Texas at Austin, Austin, TX 78712.
[2]Capt, USAF; Ph.D. Candidate, Department of Aerospace Engineering, The University of Texas at Austin, Austin, TX 78712.

adopted for their generality and relative simplicity in contrast to more classical, albeit analytical, methods.

A classical nonlinear programming (NLP) framework accounts for possible upper and lower bounds on the control input [1]. Earlier studies, in the area of libration point mission design, frequently rely on this type of constraint. For example, Senent, Ocampo, and Capella [2], employing concepts from both indirect and direct optimization, study optimal low thrust transfers to libration point orbits subject to upper and lower bounds on the engine power. However, the power is still treated as a continuous-time variable within those bounds. Other studies that incorporate NLP techniques for libration point mission design, and/or formation flight in that region, include the work of Marchand, Howell, and Betts [3], Millard and Howell [4], Infeld et al. [5], and Hughes [6]. Marchand, Howell, and Betts [3] initially employ Floquet analysis to identify naturally bounded formations and subsequently use NLP to identify the optimal single impulse that achieves a closed relative path near the Sun–Earth/Moon libration points. Millard and Howell [4] employ NLP techniques to optimize both formation imaging quality and fuel consumption. Infeld et al. [5] employ a classical NLP formulation in addressing formation design and control near the libration points. In this case, a Legendre pseudospectral method is implemented through the use of DIDO, an optimization software package that exploits SNOPT [7]. More recently, Hughes [6] presents a generalized NLP formulation that is suitable for a wide range of formation configurations and dynamical regimes. In all of these studies, the most complex actuator constraint considered is, at most, a simple bound on the elements of the control vector. Oftentimes, in fact, these constraints are not meant to imply a physical hardware limitation but, rather, are employed as a means of limiting the search space explored by the optimizer. In either case, the optimizer treats the control variable as both temporally and spatially continuous within the control bounds.

It is important to establish the distinction between a temporally continuous and a spatially continuous function. In the present context, a control variable is a temporally continuous function if it is defined over a continuous domain (i.e., time). In contrast, a bang-off-bang control solution, although temporally continuous, is spatially discontinuous because the range of the function is limited to a finite set of values (i.e., on or off). This investigation considers the preliminaries associated with incorporating control actions that are spatially discontinuous within the NLP framework. The on/off times that define each thrust and coast segment are treated as parameters in the optimization process. These switching times are referred to here as knots. Thus, although this study is still focused on an optimal "control" problem, the control variable is not an optimization parameter. Instead, the control switching times are the parameters to be optimized for a given mission scenario.

Initially, a single-impulse formulation is considered as a means of identifying the minimum requirements of the formulation when a controlled "periodic" orbit is sought. Although a single impulse is determined to be insufficient to achieve controlled periodicity, the example serves to identify the most efficient cost index formulation to minimize cost within the specified constraints. It also serves as a simple way to introduce the basic notation of the formulation employed throughout the document. In addition, a simple way of verifying the optimality of the transcribed solutions is presented that extends beyond the single-impulse case. This then leads to the mathematical preliminaries associated with formulations involv-

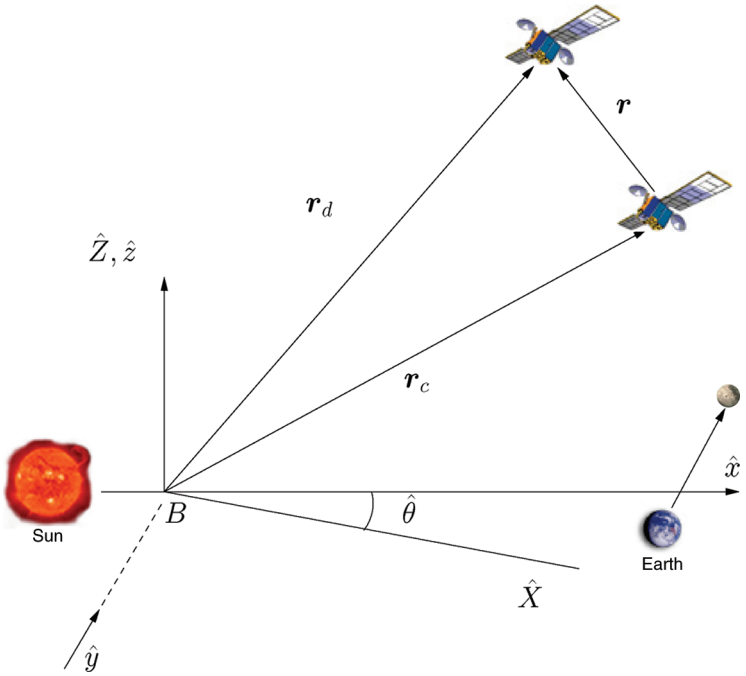

FIG. 1. Two S/C Formation in the Sun–Earth/Moon CR3BP.

ing multiple impulses and multiple fixed-thrust finite-burn sequences that do achieve controlled periodicity. A set of sample results are presented for each of the three formulations discussed. The results of the multiple impulse example are in fact employed as a startup solution to the fixed-thrust finite-burn example. Although the actuator constraints considered in this study are not the most complex that can be imposed, they are sufficient to demonstrate the general concept. That is, the use of knots as an effective means of treating problems that include temporally continuous but spatially discrete control inputs.

Dynamical Model

Circular Restricted Three-Body Model

In this study, the central spacecraft is termed "chief" while all other vehicles in the formation are denoted as "deputies." In the circular restricted three-band problem (CR3BP), the motion of the chief spacecraft is described in terms of rotating coordinates (R) relative to the barycenter (B) of the Sun–Earth/Moon primaries. In this frame, the rotating x-axis is directed from the Sun toward the Earth–Moon barycenter, as illustrated in Fig. 1. The z-axis is normal to the plane of motion of the primaries, and the y-axis completes the right-handed triad.

In Fig. 1, the position vectors $\mathbf{r}_c$ and $\mathbf{r}_d$ locate the chief and deputy vehicles with respect to an inertially fixed reference point, in this case B. Subsequently, the position of the deputy with respect to the chief spacecraft is given by the vector $\mathbf{r} = \mathbf{r}_d - \mathbf{r}_c = [x, y, z]^{\mathrm{T}}$. Consistent with this terminology, the relative equations of motion can generally be summarized as

$$\ddot{\mathbf{r}}(t) = \Delta\mathbf{f}(\mathbf{r}(t), \dot{\mathbf{r}}(t), \mathbf{r}_c(t), \dot{\mathbf{r}}_c(t)) + \Delta\mathbf{u}(t) \tag{1}$$

where

$$\Delta\mathbf{f}(\mathbf{r}(t), \dot{\mathbf{r}}(t), \mathbf{r}_c(t), \dot{\mathbf{r}}_c(t)) = \mathbf{f}(\mathbf{r}_d(t), \dot{\mathbf{r}}_d(t)) - \mathbf{f}(\mathbf{r}_c(t), \dot{\mathbf{r}}_c(t)) \tag{2}$$

and

$$\Delta\mathbf{u}(t) = \mathbf{u}_d(t) - \mathbf{u}_c(t) \tag{3}$$

If the equations of motion are formulated in terms of inertial coordinates, the vector function $\mathbf{f}(\cdot)$ evaluates the sum of the external forces acting on a given vehicle. However, because the working frame selected in this study is not inertial, $\mathbf{f}(\cdot)$ also includes kinematic terms specifically associated with the rotating coordinate system selected, R. Similarly, the vectors $\mathbf{u}_d$ and $\mathbf{u}_c$ in equation (3) denote the associated control accelerations applied by the deputy and chief vehicles, respectively, in the appropriate coordinate system.

In the present investigation, the chief spacecraft is assumed to evolve along a naturally existing solution such that $\mathbf{u}_c(t) = 0$, hence $\Delta\mathbf{u}(t) = \mathbf{u}_d(t)$. Earlier studies [3] indicate that, in optimization applications, it is most efficient to reduce the order of the dynamical model by assuming the path of the chief vehicle to be a known function of time. That is because the absolute state of the chief spacecraft and the relative state of the deputy vehicle differ by several orders of magnitude. These relative scale differences pose numerical difficulties during the solution process. Indeed, variable scaling can empower or hinder the solution of a nonlinear program [1]. Because no closed-form solutions are available in the CR3BP, scaling issues are overcome by treating the state of the chief spacecraft as a known function of time through the use of cubic splines. Consistent with the method employed by Marchand, Howell, and Betts [3], this requires that the reference orbit of the chief spacecraft be numerically identified first, independent of the optimization process. Spline coefficients are then generated and stored based on this numerically determined reference orbit. Because the relative equations of motion are explicitly dependent on the state of the chief spacecraft, the spline coefficients are used to obtain the state of the chief at any point in time during the optimization process. As detailed by Marchand, Howell, and Betts [3], this approach reduces the order of the state vector and eliminates the scaling issues previously described.

Aside from the relative scaling issues identified by Marchand, Howell, and Betts [3], other scaling difficulties arise if the nondimensional form of the relative equations of motion is employed. For example, in the CR3BP, the state vector is often scaled (i.e., nondimensionalized) by defining the characteristic length as the mean distance between the Earth and the Sun and the characteristic time as the inverse of the Earth's mean motion. Because the nominal formations presently under consideration do not exceed distances over 20,000 km from the chief spacecraft, the resulting position, velocity, and control acceleration variables are disproportionately scaled. Thus, the dimensional relative equations of motion are employed but the deputy states are represented in units of meters for position and μm/s for velocity (i.e., 10^{-6} m/s).

Nonlinear Optimal Control for Formation Keeping

In previous investigations, Howell and Marchand [8–12] propose a variety of formation-keeping approaches, continuous and discrete, optimal and nonoptimal. Some of these investigations [9] incorporate nonlinear control methods under the assumption that the spacecraft is capable of unconstrained actuation. Others [12]

focus on impulsive schemes, like targeters and Floquet-based methods, that assume the spacecraft is capable of performing the necessary corrections impulsively and in any direction. The conclusions from these earlier studies indicate that, in the vicinity of the libration points, non-natural configurations require nearly continuous actuation to maintain path tracking within the tolerances specified by the mission. Unfortunately, as presently envisioned, propulsive-based actuation may conflict with the science goals of some interferometry mission concepts. From a science perspective, the data gathered by the interferometer may be corrupted by either plume impingement, structural vibrations induced by thruster firing, or any other source of misalignment or sensor cross-contamination introduced by the proximity of the vehicles in the formation. The impact of these will of course depend on the relative distance and attitude of the vehicles.

Precision tracking either in the absence of actuation or in the presence of inexact actuation is not a generally reasonable expectation in dynamically sensitive regimes. Thus, the next logical step is to establish reasonable expectations regarding tracking accuracy in the presence of both mission and actuator constraints. The present study is an initial step toward that goal. Specifically, the focus here is on addressing, at a basic level, the implementation of actuator constraints that lead to independent sequences of bang-off-bang control solutions. This is consistent with a scenario, for example, where the spacecraft thrusters are unable to deliver variable thrust and are limited, instead, to fixed thrust levels.

Of course, optimization methods are not self-starting. Startup arcs are required to identify neighboring optimal configurations. Earlier studies have established that Floquet analysis [12] is useful in identifying naturally occurring formations that can subsequently serve as startup arcs to an optimization process [3]. However, for the range of sample formation geometries considered here, generally focused on small relative distances, it is not necessary to provide an exceedingly accurate startup arc. In fact, it is sufficient to specify arcs that exhibit some of the salient desired characteristics even if the constraints are not all initially met.

Ultimately, the goal of the optimization process is to minimize the propulsive cost, within the bounds established by the thrusters and the mission constraints, and the deviations from the specified nominal path. The nonlinear programming approach [1] adopted relies on a Hermite-Simpson discretization of the dynamical model. The solution of the resulting sequential quadratic programming problem is identified using SNOPT [7]. Comparative examples are presented to validate the methodology. Also, a method of validating the optimality of the numerically determined solutions is presented.

Transcription of the Optimal Control Problem

The basic approach for solving the optimal control problem by transcription is discussed by Betts [1]. In a transcription approach, the time interval is divided into n_n nodes such that

$$t_0 = t_1 < t_2 < \cdots < t_{n_n} = t_f \tag{4}$$

The associated states and controls along the trajectory are then defined at the nodes, and are included as optimization parameters in the nonlinear programming problem. Let us introduce the notation $\mathbf{y}_j = \mathbf{y}(t_j)$ to indicate the value of the state vector at the j^{th} node. Similarly, the control vector associated with this node is given by $\mathbf{u}_j = \mathbf{u}(t_j)$. In the direct transcription method adopted here, the state

equations are represented by a system of defect equality constraints based on a compressed Hermite-Simpson discretization [1]. The parameter vector that contains the NLP variables is subsequently defined as

$$\mathbf{x} = [\mathbf{y}_0^{\mathrm{T}}, \mathbf{u}_0^{\mathrm{T}}, \cdots, \mathbf{y}_f^{\mathrm{T}}, \mathbf{u}_f^{\mathrm{T}}, \mathbf{p}^{\mathrm{T}}, t_0, t_f]^{\mathrm{T}} \tag{5}$$

where $\mathbf{p}$ represents a generic vector of problem parameters while t_0 and t_f denote the times associated with the initial and terminal nodes along the solution, respectively.

As a result of the transcription process, the differential-algebraic system defining the optimal control problem, for each segment, is replaced by NLP constraints of the form

$$\mathbf{c}_l \leq \mathbf{c}(\mathbf{x}) \leq \mathbf{c}_u \tag{6}$$

where

$$\mathbf{c}(\mathbf{x}) = [\boldsymbol{\zeta}^{\mathrm{T}}, \boldsymbol{\psi}_0^{\mathrm{T}}, \boldsymbol{\psi}_f^{\mathrm{T}}, \boldsymbol{\beta}^{\mathrm{T}}]^{\mathrm{T}} \tag{7}$$

The vectors $\mathbf{c}_l$ and $\mathbf{c}_u$ define the lower and upper bounds of $\mathbf{c}(\mathbf{x})$, respectively. For example, if $\mathbf{c}_j(\mathbf{x})$, the j^{th} element of $\mathbf{c}(\mathbf{x})$, is associated with an equality constraint, then $\mathbf{c}_{l_j}$ and $\mathbf{c}_{u_j}$ are both set to zero. The first elements of $\mathbf{c}(\mathbf{x})$, labeled $\boldsymbol{\zeta}$, are equality constraints associated with the defect equations [1]. For example, if $\boldsymbol{\zeta}^T = [\zeta^T{}_2 \cdots \zeta^T{}_{n_n}]$, then each ζ_j represents a defect constraint of dimension n_y evaluated at any one of the nodes, except for that corresponding to the fixed initial state. The vectors $\boldsymbol{\psi}_0^{\mathrm{T}}$ and $\boldsymbol{\psi}_f^{\mathrm{T}}$, of dimension n_{ψ_0} and n_{ψ_f}, represent boundary conditions imposed at the initial and terminal nodes, respectively. The remaining constraint, $\boldsymbol{\beta}^{\mathrm{T}}$ of dimension n_β, consists of vector elements $[\boldsymbol{\beta}_0^{\mathrm{T}}, \ldots, \boldsymbol{\beta}_f^{\mathrm{T}}]$. Each element of $\boldsymbol{\beta}^{\mathrm{T}}$ represents a nonlinear vector algebraic path constraints imposed at any one of the n_n nodes along the solution.

Solution of the Optimal Control Problem

Once the problem is transcribed, the goal is to identify the optimal $\mathbf{x}$ that minimizes $F(\mathbf{x})$, where $F(\mathbf{x})$ denotes a scalar cost index subject to $\mathbf{c}(\mathbf{x})$ constraints. A necessary condition for a solution to exist is that the number of parameters, i.e., the dimension of $\mathbf{x}$, is greater than the number of constraints, i.e., the dimension of $\mathbf{c}(\mathbf{x})$. Thus, throughout this document, the first step in the setup of any problem is to determine whether this requirement is satisfied and, if not, what changes are necessary to the problem setup to ensure it is.

The size of the NLP problem can be quantified as a function of the number of variables, n, and constraints, n_c. Traditionally, a solution is often transcribed as a single segment with n_c nodes. Each of these nodes is associated with n_y states and n_u control variables. In addition, the entire solution may be dependent on up to n_p parameters that are constant along the solution. Thus, the parameter vector is of dimension

$$n = (n_y + n_u)n_n + n_p + 2 \tag{8}$$

where the "2" is indicative of the initial and final times contained within the parameter vector $\mathbf{x}$. Per the definition in equation (7), the dimension of the constraint vector, $\mathbf{c}(\mathbf{x})$, is given by

$$n_c = n_y(n_n - 1) + n_{\psi_0} + n_{\psi_f} + n_\beta n_n \tag{9}$$

where $n_y(n_n - 1)$ represents the number of defect constraints, n_{ψ_0} and n_{ψ_f}, denote the number of initial and terminal boundary conditions, respectively, and $n_\beta n_n$ is

the total number of nonlinear algebraic constraints imposed across all nodes. This large, sparse NLP can be solved efficiently using either a sequential quadratic programming (SQP) method or an interior point (barrier) method [13]. Thus, the classical approach to solving optimal control problems by transcription involves three primary steps:

1. Transcribe the optimal control problem into an NLP.
2. Solve the sparse NLP using SQP methods.
3. Assess the accuracy of the solution then refine the discretization if needed and repeat the above steps.

The transcription process described above is effective for a large class of optimization problems, though it is most accurate and effective for problems where the control variables are both temporally and spatially continuous. However, for problems that involve spatially discontinuous state or control variables, such as impulsive maneuvers or fixed-thrust finite burns, the effectiveness and accuracy of this classical formulation can be improved with the adjustments presented here.

A control solution is considered "impulsive" if it leads to an instantaneous change in velocity at a given point in time. The resulting discontinuity in the state implies the corresponding solution is temporally but not spatially continuous. Similarly, control solutions that employ fixed-thrust finite burns lead to control accelerations that are temporally though not necessarily spatially continuous. For example, in an optimal unconstrained low-thrust problem, if the engines are engaged over the duration of the trajectory, the control solution is both temporally and spatially continuous. In contrast, the control acceleration associated with a bang-off-bang or thrust-coast-thrust sequence exhibits a discontinuity each time a thrust-coast or coast-thrust switch occurs. Thus, a bang-off-bang sequence is temporally but not spatially continuous.

The following sections are devoted to an alternate formulation that better accommodates spatial discontinuities in the state or control variables within an NLP framework. A single-impulse formulation is presented first to introduce the basic notation within the context of a classical example. This is followed by a multiple impulse and multiple fixed-thrust finite-burn formulations. The resulting formulations are employed in the numerical examples presented later.

Formulation for Single-Segment Single-Impulse Control

The classical transcription formulation discussed earlier easily accommodates a single impulsive maneuver at the start of the trajectory. Here, the start of the path is set to occur at time t_0. The control parameters are given by $\Delta\mathbf{v}(t_0)$ of size $n_u = 3$. For this formulation, n_n still denotes the number of nodes, uniformly spaced in time between t_0 and t_f, and n represents the number of parameters that result from the discretization. The number of nodes, although generally arbitrary, is selected to minimize the computational costs without compromising the fidelity of the resulting optimal arc. The size of the parameter vector for a single-impulse problem is subsequently determined as

$$n = n_y n_n + n_u + 2 \tag{10}$$

This represents the total number of variables that account for the number of states (n_y) per node (n_n), the dimension of the control vector, $\Delta\mathbf{v}(t_0) \in \mathcal{R}^{n_u}$, and the initial and final times, t_0 and t_f.

The initial conditions are completely specified for the n_y states and initial time. For the particular example considered in this study, at least in regards to the single-impulse case, only the terminal position is fixed. Thus, a total of $n_y/2$ terminal constraints are further imposed. An additional $n_y(n_n - 1)$ defect constraints are also specified. Thus, the total number of constraints, n_c, is determined as

$$n_c = n_y(n_n - 1) + (n_y + 1) + n_y/2 \tag{11}$$

The multiple impulse and fixed-thrust finite-burn formulations that follow focus on a slightly different example, one with a fixed terminal state and time to ensure controlled periodicity of the path. As discussed later, a single impulse alone is insufficient in that case, and thus only the terminal position is fixed in the single-impulse examples. However, the single-impulse formulation is employed to present several key concepts and motivate the need for a method that adequately incorporates multiple impulses and multiple bang-off-bang thrust sequences.

Formulation with Multiple Segments for Multiple Impulse and Fixed-Thrust Finite-Burn Control

In the event a solution includes interior state or control discontinuities, incorporating knots in the formulation offers significant improvements over classical formulations. A *knot* is a point that divides two arcs or segments at a point in time where either a state or control discontinuity occurs. A knot may be either fixed or free in time. Knots also offer an effective means of incorporating constraints that vary by segment. By definition, any jump discontinuities along the solution must occur at a knot. If the control variables are defined by impulsive maneuvers, state discontinuities are introduced through the velocities every time a maneuver is executed. In a fixed-thrust finite-burn scheme, in contrast, control discontinuities are introduced every time a switch occurs, from *on* to *off* or vice versa.

Figure 2 illustrates the conceptual relation between knots, nodes, and segments. Consider a formulation with n_s segments. Each segment consists of n_n nodes, and each node is associated with n_y states. Knots define the start and end times of each segment along a trajectory. The knot time that defines the end of the last segment is defined as t_f. The time that defines the start of the first segment, t_0 is always set to zero because the dynamical model selected is time-invariant. All subsequent knots are defined relative to t_0. Thus, the initial time is not considered a true knot in the present formulation. This implies that the number of knots, n_k, is equal to the number of segments, n_s, and that a total of $n_s - 1$ knots exist between t_0 and t_f.

If all knot times are unconstrained, the parameter vector $\mathbf{x}$ is augmented to include the n_k knots. The node spacing in time, although assumed uniform between segments, may vary not only by segment but also from one iteration of the optimizer to the next. Ultimately, the spacing of nodes along a segment is strictly defined by the knots that define the start and end of that segment and the number of nodes per segment initially selected.

In the CR3BP, allowing an impulse at the initial time and each of the $n_s - 1$ interior knots leads to a parameter optimization problem of dimension

$$n = n_s(n_y n_n + n_u) + n_k \tag{12}$$

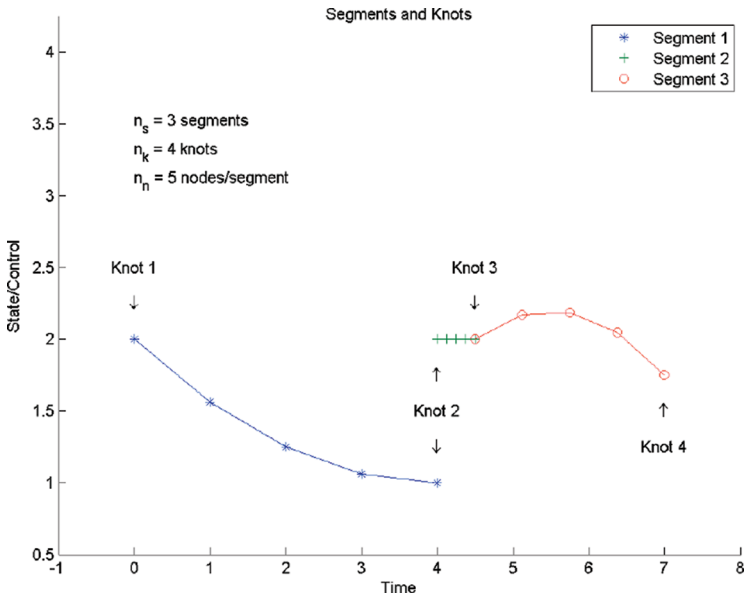

FIG. 2. An Example of Segments and Knots.

That is, a total of n parameters can be adjusted to achieve the optimization goals. These parameters include the n_y states associated with each of the n_n nodes over n_s segments, the control parameters, n_u, associated with each of the n_s segments, and the knot times, n_k, that define the duration of each segment. The existence of optimal solutions, then, hinges on the requirement that the number of constraints, n_c, imposed on the problem be strictly less than n. Although time continuity is assumed between segments, it is not treated here as an explicit constraint. Instead, temporal continuity between segments is inherent to the formulation adopted in this study. That is, the n_n^{th} node of one segment (i.e., the last node) coincides in time with the first node of the next segment.

Among the set of explicit constraints considered in this study, up to n_y state continuity constraints may be imposed at any of the interior knots. Thus, a maximum of $n_y(n_s - 1)$ state continuity constraints are possible on any given problem. For example, consider the illustration in Fig. 2. Here, a discontinuity appears at Knot 2. If this discontinuity is associated with a velocity state, and an impulsive maneuver is allowed, the velocity continuity constraint may be formulated as

$$\mathbf{v}_{j,1,2} - \mathbf{v}_{j,n_n,1} - \Delta\mathbf{v}_{j,2} = \mathbf{0} \tag{13}$$

Following the notation in Fig. 2, $\mathbf{v}_{j,1,2}$ represents the j^{th} element of the velocity vector, $\mathbf{v}$, associated with the first node along the second segment. Similarly, $\mathbf{v}_{j,n_n,1}$ is the j^{th} element of $\mathbf{v}$ associated with the n_n^{th} node of the first segment. Finally, $\Delta\mathbf{v}_{j,2}$ denotes the j^{th} element of an impulsive maneuver, $\Delta\mathbf{v}$, applied at the second knot. If an impulsive maneuver is not allowed at this point, then $\Delta\mathbf{v}_{j,2}$ is strictly equal to zero.

Position discontinuities are not allowed along a solution. Thus, the elements of the state vector that correspond to position are always subject to strict continuity

constraints across segments. The behavior illustrated at Knot 3 in Fig. 2, for instance, is representative of a position continuity constraint at a knot given by

$$\mathbf{r}_{j,1,3} - \mathbf{r}_{j,\mathrm{n_n},2} = \mathbf{0} \tag{14}$$

Consistent with the notation previously introduced, $\mathbf{r}_{j,1,3}$ represents the j^{th} element of the position vector, $\mathbf{r}$, associated with the first node of the third segment, whereas $\mathbf{r}_{j,n_n,2}$ denotes the j^{th} element of position associated with the $n_n{}^{\text{th}}$ node of the second segment.

To preserve the tractability of the intermediate solution arcs, constraints are also imposed on the length of a segment defined by the knots at the start and end of that arc. For example, if left unconstrained, the optimizer may identify intermediate solutions with negative segment lengths by allowing the temporal ordering of knots to switch. To discourage this behavior, inequality constraints are imposed to ensure that the $(i + 1)^{\text{th}}$ knot does not lag behind the i^{th} knot. This leads to $n_k = n_s$ inequality constraints on the knot times, because t_0 is not considered a knot in the time-invariant problem.

Another set of required constraints correspond to the Hermite-Simpson discretization of the dynamical model and leads to a total of $n_s n_y(n_n - 1)$ defect equations. Finally, up to $n_y + 1$ initial state constraints and $n_y + 1$ terminal state constraints may be imposed. Thus, the total number of constraints can add up to a maximum of

$$n_c = n_y(n_s - 1) + n_s + n_s n_y(n_n - 1) + 2(n_y + 1) \tag{15}$$

For n_c constraints and one scalar cost index, a total of n_F function evaluations are performed during each iterative step in the search for an optimal solution. The difference between the number of parameters and the number of function evaluations

$$n - n_F = n_s n_u - n_y - 2 \tag{16}$$

is employed in determining whether the NLP problem is "well posed." Because $n - n_F$ must be positive for the problem to remain underconstrained, the user can rely on equation (16) as a guide to determine the minimum number of control variables, n_u, segments, n_s, and by implication the number of knots, $n_k = n_s$, necessary for a solution to exist.

Aside from the initial single-impulse case, the examples that follow focus on minimizing the cost of establishing controlled relative periodic orbits. Depending on the cost index selected, up to n_y terminal state constraints are imposed, $n_y/2$ in position and $n_y/2$ in velocity, of the form

$$\mathbf{r}_{j,n_n,n_s} - \mathbf{r}_{j,1,1} = \mathbf{0} \tag{17}$$

$$\mathbf{v}_{j,n_n,n_s} - \mathbf{v}_{j,1,1} - \Delta\mathbf{v} = \mathbf{0} \tag{18}$$

For a single-impulse example, only $n_y/2$ constraints are imposed because the terminal velocity cannot be constrained. Instead, the single-impulse cost index is formulated to impose a penalty on the terminal velocity discontinuity. In contrast, the multiple impulse formulation constrains all n_y terminal states. The multiple impulse and fixed-thrust finite-burn examples that follow are specifically concerned with establishing a controlled relative periodic path. Naturally, because a

relative periodic path is not known, *a priori* precision tracking is not a goal in these examples. Rather, these examples are designed as a simple means of illustrating the use of knots to accommodate spatial discontinuities in states or control variables. Note that, for the multiple impulse example, if all $\Delta\mathbf{v}$'s are zero, the optimal solution is considered naturally periodic. Otherwise, a controlled periodic path is the best available solution. Furthermore, it is worth noting that, if a suitable initial guess is provided, the optimal solution should exhibit some of the characteristics of the startup arc. However, no guarantees are implied because precision tracking is not built into the selected cost indeces.

When multiple impulsive maneuvers are employed, n_u represents the dimension of any one of the $\Delta\mathbf{v}$'s applied. An impulsive maneuver here is always applied either at the start of the first segment or any of the interior knots. Thus, in the multiple impulse case, for n_i impulses, the optimization problem is of dimension

$$n = n_s n_n n_y + n_i n_u + n_k \tag{19}$$

In contrast, in a bang-off-bang thrust sequence, n_u represents the dimension of the thrust vector. In this case, each of the n_n nodes along a segment is assigned the same constant-thrust value, be it zero or nonzero. Thus, for finite burns of constant magnitude, the problem is of dimension

$$n = n_s n_n (n_y + n_u) + n_k \tag{20}$$

and the thrust magnitude at each node is prespecified such that

$$0 = \mathbf{u}^{\mathrm{T}}\mathbf{u} - a_s^2 \tag{21}$$

In equation (21), a_s is a constant-thrust acceleration magnitude specified for a given segment. Note that a_s is a nonzero constant only over a burn arc. A coast arc, in contrast, is associated with $a_s = 0$. Because a_s is independently prespecified over each segment, the thrust magnitude is not itself a control parameter. Rather, the $n_s n_n n_u$ terms in equation (20) are associated with the n_u components of the control acceleration vector at each of the $n_n n_k$ nodes. That is, the n_u dimensional thrust direction contributes $n_u n_n n_k$ parameters to $\mathbf{x}$. A unit magnitude constraint is imposed on the thrust direction at each node, regardless of whether or not $a_s = 0$. Finally, it is important to note that, in the examples presented here, the n_u control parameters are strictly associated with translational vehicle control. That is, the present examples assume that sufficient and independent means of attitude control are available to reorient the deputy spacecraft as needed during the execution of either an impulsive or fixed-thrust finite burn.

Optimal control solutions that exhibit a "bang-bang" structure often arise in minimum time problems [14, 15]. Similarly, "bang-off-bang" solutions are not uncommon in minimum effort problems [14, 15]. In spacecraft applications, the term "bang" implies the engines fire at a constant-thrust level until the burn ends at the "off" command. Because the duration of the optimal burn is not known *a priori*, an optimal fixed-thrust finite-burn problem involving multiple "bang" and "off" sequences may be formulated by dividing the solution into segments connected by knots. Thus, a segment is designated as either a "bang" segment or an "off" segment. Because the thrust magnitude is fixed over any given segment, the NLP solver seeks to identify the optimal duration of each of the "bang" and "off" segments over the entire trajectory as well as the optimal thrust

direction at each node. In the present example, the constraints are formulated such that odd-numbered segments correspond to an engine "on" state while the even-numbered segments are considered coast or engine "off" segments. Therefore, for the j^{th} node along the k^{th} segment, the thrust magnitude is specified according to

$$0 = \mathbf{u}_{jk}^{\mathrm{T}}\mathbf{u}_{jk} - a_s^2(\tfrac{1}{2} - \tfrac{1}{2}(-1)^k) \tag{22}$$

The magnitude, $|\mathbf{u}_{jk}|$, alternates between a_s and zero for each k, presenting a bang-off-bang scheme for any formulation with three or more segments. In this case, the knot times and the thrust direction at each node are free parameters in the optimization process. Thus the algorithm determines the optimal times to burn or coast and the optimal direction of the thrust vector. Without prior knowledge of the number of burn arcs required, the user can initially specify an arbitrary number of segments, and any unnecessary burn arcs are removed by the optimization process when two switching points coincide in time. This may occur if the user specified more odd (burn) segments than the optimal solution requires. Thus, a control profile with an instantaneous off-on-off sequence can be interpreted as an unnecessary burn segment.

The fixed-thrust finite-burn example presented next accounts for up to $n_y + 1$ initial time and state constraints, $n_y + 1$ terminal state and time constraints, $n_s n_y(n_n - 1)$ defect constraints, up to $n_y(n_s - 1)$ interior knot state continuity conditions, $n_n n_s$ unit magnitude constraints on the thrust direction, and n_s time inequality constraints to preserve the time ordering of the knots. Thus, the total number of constraints and function evaluations (including the cost index) are given by

$$n_c = 2(n_y + 1) + n_y n_s(n_n - 1) + n_y(n_s - 1) + n_n n_s + n_s \tag{23}$$

$$n_F = n_c + 1 \tag{24}$$

Subtracting equation (24) from equation (20) leads to

$$n - n_F = (n_u - 1)n_n n_s - n_y - 2 \tag{25}$$

Once again, the quantity $n - n_F$ represents the criteria used to determine the minimum number of nodes and knots, for $n_k = n_s$, necessary for the existence of an optimal solution. For example, if $n_u = 3$ and $n_y = 6$, then $n - n_F = 2n_n n_s - 8$, which indicates that the problem is underdetermined as long as $n_n n_s$, the total number of nodes, exceeds four. Of course, an increased number of nodes and segments offers a more diverse solution space in the search for optimality.

Solution of the Nonlinear Program

An SQP approach is selected to solve the NLP problem that results from transcription of the optimal control problem. The basic elements of the NLP formulation are consistent with those presented by Betts [1, 16, 17, 18, 19]. However, the sparse quadratic program is solved using SNOPT [7]. As previously mentioned, a reduced order dynamical model [3] and scaled relative equations of motion are employed to enhance the numerical efficiency and accuracy of the process. The following sections present a brief discussion on the startup arcs employed and the results of three examples consistent with the single impulse, multiple impulse, and fixed-thrust finite-burn formulations previously discussed.

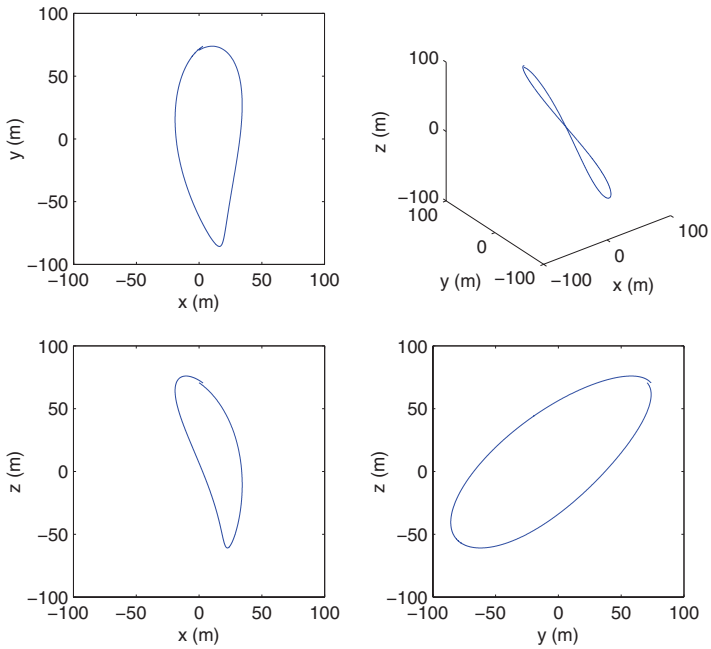

FIG. 3. Floquet Initial Guess [12].

Startup Arcs

Two different startup trajectories are employed here. The first startup arc, illustrated in Fig. 3, originates from Floquet analysis [12]. Marchand, Howell, and Betts [3] present the results of an optimal single-impulse problem using this exact same arc and demonstrate that a local minimum exists very close to this startup solution. To better explore the neighboring solutions, a more arbitrary initial guess is also considered. The arbitrary guess, in Fig. 4, is not considered "feasible" because, initially, the corresponding states do not satisfy the dynamical constraints. Although such an arc is not generally considered the best available, it is reasonably accurate for the optimizer in this case. That is because, in this example, the deputy path is always within 100 meters of the chief, and the long orbital period leads to small relative velocities. Thus, the optimizer is able to compensate for the inadequacies of the startup solution as long as the number of nodes employed is sufficient to adequately represent the geometry of the path. The optimal solutions obtained from these two startup arcs are contrasted to establish the robustness of the optimal solution. If the resulting optimal paths reside near the same local minimum, then the optimality of the solution is considered robust.

The arbitrary guess, in Fig. 4, places the initial state along a circular relative path. The orbital period of the deputy is selected to match that of the chief spacecraft precisely. Because this startup arc does not satisfy the dynamical constraints of the problem, the optimizer has an opportunity to explore a larger area of phase space before converging on an optimal solution. In contrast, the Floquet solution originally identified by Howell and Marchand [12] is already nearly periodic. Thus, the optimizer does not search much farther than the immediate vicinity of the startup arc for an optimal solution. A comparison between the locally optimal solutions obtained from various yet somewhat similar startup arcs,

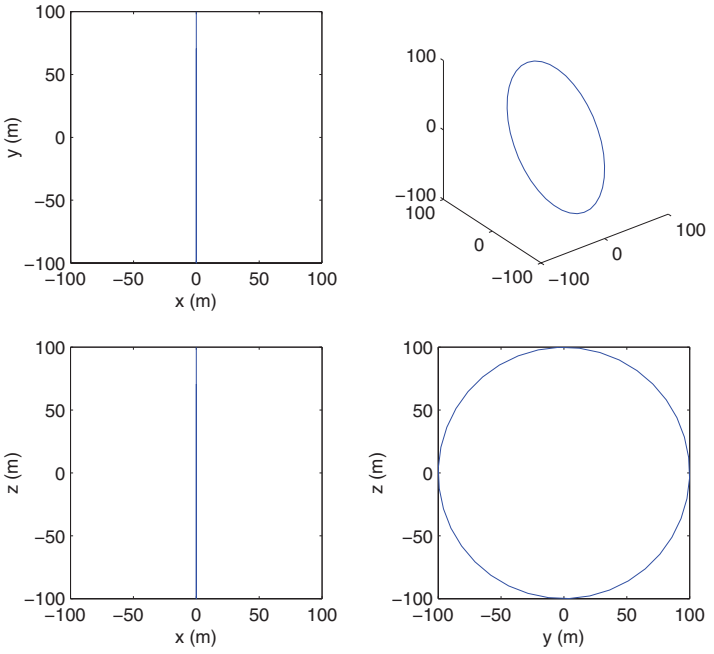

FIG. 4. Arbitrary Initial Guess.

such as that presented in the following section, allows for an assessment of optimizer "inertia." This is an important aspect of the analysis when the neighborhood of interest is rich in locally optimal solutions and one wishes to determine the robustness of the optimal solutions identified.

Single-Impulse Solution

For the four cases considered, the optimal deputy paths are illustrated in Fig. 5. Note that, for most cases, the resulting closed relative path is very similar to the startup arc in Fig. 3. In each case, the initial velocity of the startup arc is arbitrarily set to zero, $\mathbf{v}_0^- = 0$. Subsequently, an initial velocity "update" is necessary to place the vehicle on the desired relative closed path. The updated velocity is given by $\mathbf{v}_0^+ = \mathbf{v}_0^- + \Delta\mathbf{v}$. In essence, the $\Delta\mathbf{v}$ vector represents the initial velocity required for the spacecraft to return to the initial position after one revolution.

Because naturally periodic relative orbits of this type are not available, [12] the terminal velocity, $\mathbf{v}_f^-$, is different from the initial updated velocity, $\mathbf{v}_0^+$. In the CR3BP, if the period of the controlled relative path is fixed to match the orbital period of the reference halo orbit, then $\mathbf{v}_0^+ - \mathbf{v}_f^-$ represents the maintenance maneuver required to continue along the same path over every successive revolution. If, however, t_f is free, then $\mathbf{v}_0^+ - \mathbf{v}_f^-$ is simply a measure of the smoothness of the closed path after one revolution. In this case, when t_f does not match the period of the reference halo orbit, $\mathbf{v}_0^+ - \mathbf{v}_f^-$ will not lead to the exact same path over subsequent revolutions. Even with added control, some orbital drifting is expected relative to the initial startup arc. That is, it is not possible, without continuous control, to maintain a periodic relative orbit of this type. This is supported by the results presented by Howell and Marchand [12], who identified only one type of strictly periodic relative orbits near the chief vehicle. Other

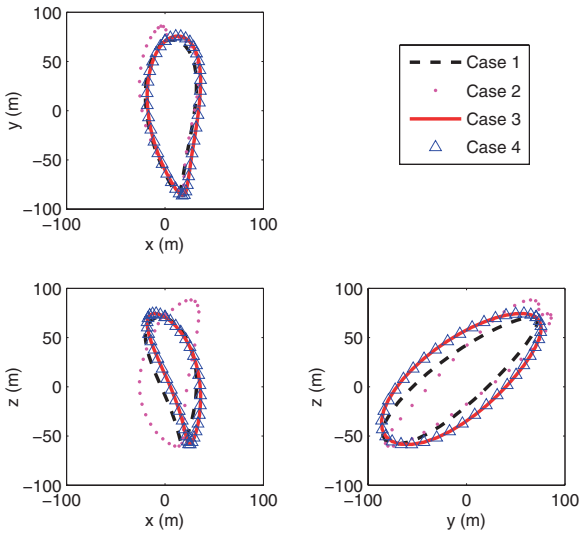

FIG. 5. Closed Relative Path Solutions for Cases in Table 2.

neighboring solutions are, at best, nearly periodic but generally drifting. The arc employed here is the first revolution along one of these drifting orbits.

For either initial guess, consider a single-impulse single-segment example with free terminal time. In this case, a total of $n_c = n_y + n_y(n_n - 1) + n_y/2$ constraints are implemented, n_y initial state constraints, $n_y(n_n - 1)$ Simpson continuity constraints, and $n_y/2$ terminal position constraints. Note that, although the generalized formulation previously presented assumes the initial time may be a free parameter, it is not treated as such in the following examples. That is because the CR3BP is time invariant. Thus, for simplicity, the initial time is always defined as $t_0 = 0$ and all node times are defined relative to this reference.

The total number of function evaluations is given by $n_F = n_c + 1$, n_c constraints and one scalar cost index. For the single-impulse examples presented next, the cost index may be formulated to include a penalty on the magnitude of $\mathbf{v}_0^+ - \mathbf{v}_f^-$. The total number of NLP variables, then, is $n = n_n n_y + n_u + 1$, $n_n n_y$ node states, n_u control variables (i.e., $\Delta\mathbf{v}$ at t_0), and the terminal time, t_f. Thus, $n - n_F = n_u - n_y/2 \geq 0$ is required for a solution to exist. Of course, because $n_u = 3$ and $n_y/2 = 3$, a solution exists only when the terminal time is free. That is, a single impulse alone cannot establish a periodic relative path that matches the orbital period of the reference halo orbit. A multiple impulse example, in contrast, can easily accomplish the goal.

For subsequent examples involving multiple impulses, the terminal position and velocity are both constrained to match the updated initial state. Thus, if n_k impulses are applied, one at t_0 and one at each of the $n_k - 1$ interior knots, then $n_c = n_y + n_k n_y(n_n - 1) + n_k + n_y + 1$, $n_F = n_c + 1$, and $n = n_k n_n n_y + n_k n_u + n_k$. This implies that $n - n_F = n_k(n_y + n_u) - 2(n_y + 1) \geq 0$. Thus, as long as $n_k > 2$, a solution exists. This further suggests that at least two maneuvers are required to identify a controlled periodic relative orbit of the type sought, one at the initial point t_0 and one at an interior knot.

Although it is evident that multiple impulses are required to achieve a controlled periodic relative path, it is still advantageous to study the single-impulse example, with free terminal time, to assess the impact of the problem setup and cost function

TABLE 1. Two Cost Functions

A	$J = \Delta\mathbf{v}^{\mathrm{T}}\Delta\mathbf{v}$
B	$J = \Delta\mathbf{v}^{\mathrm{T}}\Delta\mathbf{v} + w(\mathbf{v}_0^+ - \mathbf{v}_f^-)^{\mathrm{T}}(\mathbf{v}_0^+ - \mathbf{v}_f^-)$

formulation on the smoothness of the path and the initial $\Delta\mathbf{v}$. Tables 1 and 2 summarize the key parameters that result from four different numerical implementations pertaining to the single-impulse case. Table 1 lists the sample cost functions implemented for the analysis. Table 2 presents a summary of the converged optimal solution parameters. The key elements used for comparison are the startup arc and the cost function imposed. The results are assessed in terms of the magnitude of the initial impulse, $|\Delta\mathbf{v}|$ and the smoothness of the closed path, $\mathbf{v}_0^+ - \mathbf{v}_f^-$.

The same Floquet initial guess employed by Marchand, Howell, and Betts [3] is adopted here in deriving the results associated with Solution 1. The arbitrary circular startup arc leads to Solution 2. Note that the magnitude of the impulse in Solution 2 is significantly larger than that of Solution 1. This is not surprising considering the startup arcs were significantly different. Thus, the optimizer simply converged on a different locally optimal solution, one with a higher cost. Of course, it is also not surprising that the path from the arbitrary guess is not as smooth as that from the Floquet solution because the cost index employed placed no penalty on smoothness and no constraints were incorporated to that effect.

Solutions 3 and 4 are based on a multiobjective cost index that places a penalty on the terminal smoothness of the closed path. The weight factor, w, is a user-selected parameter. Naturally, an increased weight on the terminal velocity discontinuity leads to a smoother optimal trajectory.

Solutions 3 and 4 use the same weight factor in the cost function, but the optimization process is based on different startup arcs. It is interesting to note that, although Solutions 1 and 2 arrive at different local minima because of the different starting points, Solutions 3 and 4 arrive at nearly the same optimal point due to the new "smoothing" cost function. Thus having arrived in the same vicinity from different starting points helps to validate the solution. Although additional weight factors were considered, an exhaustive survey was not conducted. However, increasing the weight factor by one order of magnitude improves both the size of the initial impulse and smoothness of the terminal state.

Verification of Optimality. Consider the single-impulse example associated with Solution 3 in Table 2. Recall that the impulsive maneuver, $\Delta\mathbf{v}$, is treated as a constant parameter that is adjusted as needed during the optimization process. In

TABLE 2. Single Impulse Solutions

Solution	Transcription Method	Guess	Cost Function	w	$\lvert\Delta\mathbf{v}\rvert$ [μm/s]	$\lvert\mathbf{v}_0^+ - \mathbf{v}_f^-\rvert$ [μm/s]	t_f [days]
1	Manual	Floquet	A	N/A	17.974	7.808	170.807
2	Manual	Arbitrary	A	N/A	25.602	40.628	214.896
3	Manual	Floquet	B	10	19.352	1.418	176.641
4	Manual	Arbitrary	B	10	19.304	1.484	176.560

verifying optimality, the objective is to determine how well the optimality conditions are satisfied by the transcribed solution. In the example selected here, the cost index is given by $J = \Delta\mathbf{v}^{\mathrm{T}}\Delta\mathbf{v} + w(\mathbf{v}_0^+ - \mathbf{v}_f^-)^{\mathrm{T}}(\mathbf{v}_0^+ - \mathbf{v}_f^-)$, where $w = 10$. The startup arc [3] is characterized by $\mathbf{v}_0^- = \mathbf{0}$ and $\mathbf{v}_0^+ = \mathbf{v}_0^- + \Delta\mathbf{v} = \Delta\mathbf{v}$. Consider an augmented cost index that further incorporates the initial and terminal constraints

$$G = \Delta\mathbf{v}^{\mathrm{T}}\Delta\mathbf{v} + w(\Delta\mathbf{v} - \mathbf{v}_f^-)^{\mathrm{T}}(\Delta\mathbf{v} - \mathbf{v}_f^-) + \mathbf{v}_0^{\mathrm{T}}\psi_0 + \mathbf{v}_f^{\mathrm{T}}\psi_f \tag{26}$$

where

$$\boldsymbol{\psi}_0 = \begin{bmatrix} \mathbf{r}_0 - \mathbf{r}_{0,\mathrm{spec}} = \mathbf{0} \\ \mathbf{v}_0^- - \mathbf{v}_{0,\mathrm{spec}} = \mathbf{0} \end{bmatrix} \tag{27}$$

represents a constraint on the initial state and

$$\boldsymbol{\psi}_f = \mathbf{r}_f - \mathbf{r}_{0,\mathrm{spec}} = \mathbf{0} \tag{28}$$

is a vector of terminal position constraints. The vectors $\mathbf{r}_{0,\mathrm{spec}}$ and $\mathbf{v}_{0,\mathrm{spec}}$ are constant vectors specified by the user and remain unchanged throughout the optimization process. Note that the control parameter vector, $\Delta\mathbf{v}$, appears explicitly in the first and second terms of G. Optimality can be thus presented in one of two forms. In the classical sense, the Euler-Lagrange equations [15] require that

$$G_{\Delta\mathbf{v}} = 2\Delta\mathbf{v}^{\mathrm{T}} + 2\mathrm{w}(\Delta\mathbf{v} - \mathbf{v}_f^-)^{\mathrm{T}} = \mathbf{0} \tag{29}$$

In a parameter optimization sense, [1, 13] where the goal is to minimize a scalar cost index $F(\mathbf{x})$ subject to vector constraints $\mathbf{c}(\mathbf{x})$, optimality ensues when the derivative of $F(\mathbf{x})$ with respect to $\mathbf{x}$ is zero. The Euler-Lagrange conditions, including that in equation (29), are ultimately embedded within the transcribed optimal control problem. For example, in the present formulation, the parameter vector $\mathbf{x}$ is of dimension $n = n_s n_n + n_u + 1$. The first $n_s n_n$ elements correspond to the n_s state elements associated with each of the n_n nodes. The last four elements of $\mathbf{x}$ are the $n_u = 3$ elements of the $\Delta\mathbf{v}$ vector followed by the terminal time, t_f. The SNOPT [7] output includes the gradient of $F(\mathbf{x})$ with respect to $\mathbf{x}$ along the optimal solution. Thus, the vector $G_{\Delta\mathbf{v}}$ may also be deduced by reading the appropriate elements of the gradient of $F(\mathbf{x})$ in the transcribed problem. Subsequently, the gradient in equation (29) is deduced either directly from the optimizer output or by direct substitution of the optimal $\Delta\mathbf{v}$ and $\mathbf{v}_f^-$ into equation (29). If the control history is truly optimal, then equation (29) should be satisfied within some specified tolerance level. With either approach, the gradient is identified as

$$G_{\Delta\mathbf{v}} = [0.000219833 \quad -0.000029742 \quad 0.000043753] \text{ m/s} \tag{30}$$

Because $G_{\Delta\mathbf{v}}$ is reasonably close to zero, the optimality of the solution is verified, at least in the immediate neighborhood of the initial guess. The optimal states and costates are illustrated in Fig. 6. For each state and costate, two curves are illustrated on each plot. The dots correspond to the transcribed solution evaluated strictly at each node. The solid lines are obtained by numerically propagating the node states and costates from one node to the next using a standard Runge-Kutta integrator. It is evident, at least within the specified tolerances, that a subset of the costates extracted from the transcribed solution is directly related to the classical costates from the Euler-Lagrange equations. There are some minor discrepancies that are most apparent in the position costate curves. These differences are

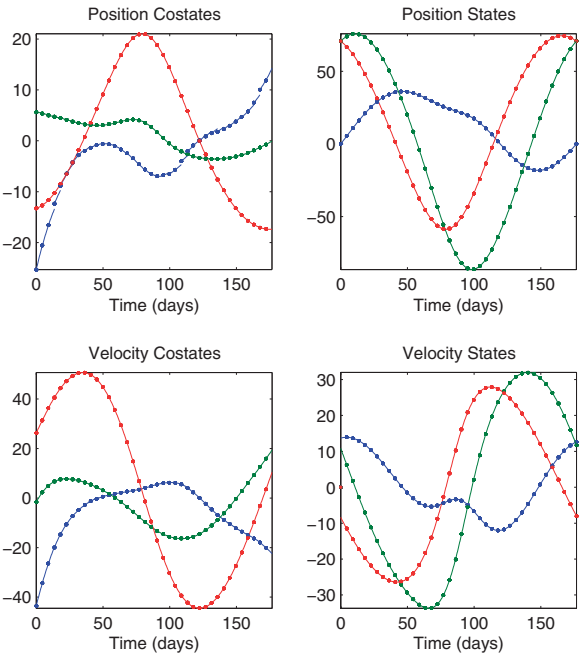

FIG. 6. Transcribed Solution versus Euler-Lagrange Integrated Solution.

ultimately attributed to the specified optimality and feasibility tolerances for the transcribed model. Increased fidelity in both should minimize these discrepancies at the expense of an increase in computational overhead. In general, however, the steps presented here offer a simple way of verifying the optimality of the transcribed solutions.

Multiple Impulse Solution

As established in the previous section, a fixed terminal time problem that seeks to establish a controlled periodic path requires at least two knots. The present example, then considers a three-impulse formulation. The first impulse, at t_0, essentially updates the initial velocity from zero to $\mathbf{v}_0^+$ while the remaining two impulses, $\Delta\mathbf{v}_2$ and $\Delta\mathbf{v}_3$, are applied at the interior knots. In total, the implementation includes three knots, one knot per maneuver performed. The terminal time along the trajectory, t_f, is constrained to match the orbital period of the reference halo orbit along which the chief vehicle in the formation evolves, roughly 180 days. In the end, if any one of the three maneuvers is unnecessary, the optimization process should nullify it.

With the addition of two new interior impulses, the cost function is augmented as follows

$$J = |\Delta\mathbf{v}_1| + w(|\Delta\mathbf{v}_2| + |\Delta\mathbf{v}_3|) + (w-1)|\mathbf{v}_0^+ - \mathbf{v}_f^-| \tag{31}$$

The constant w is not itself a weighting factor. Rather, it represents the number of revolutions over which the spacecraft must maintain the periodic path. The last term in the cost index, involving $\mathbf{v}_0^+ - \mathbf{v}_f^-$, seeks to minimize the velocity discontinuity (i.e., impulsive maneuver) that defines the start of each of the subsequent $w - 1$ revolutions. Thus, after the first revolution is completed, the

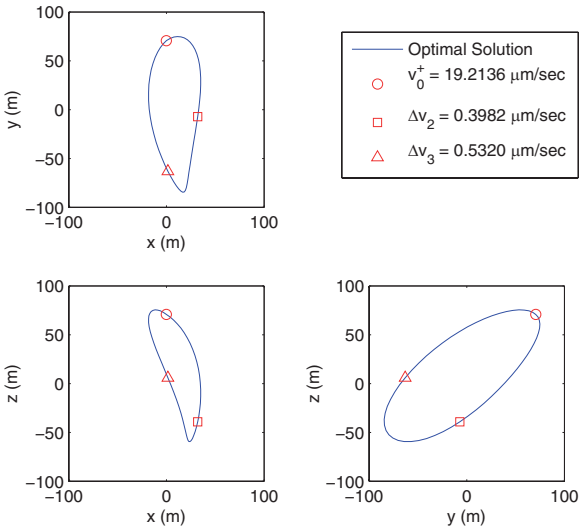

FIG. 7. Closed Relative Path Using Three Impulses.

three maneuvers that maintain the controlled periodic path are $\Delta\mathbf{v}_2$, $\Delta\mathbf{v}_3$, and $\mathbf{v}_0^+ - \mathbf{v}_f^- = \mathbf{v}_0^- + \Delta\mathbf{v}_0 - \mathbf{v}_f^-$. In essence, the cost function represents the sum of all maneuvers performed in completing all w revolutions. For the example in question, when $w = 10$, the optimal solution is shown in Fig. 7. The initial guess employed corresponds to the Floquet startup arc [3] employed in the single-impulse example previously presented. The magnitudes of the three impulses are summarized in Table 3.

The final time, or controlled period, of the closed path is 177.7177 days. Aside from constraining the terminal time, i.e., the orbital period, terminal position and velocity continuity constraints are also imposed; $\mathbf{r}_0 - \mathbf{r}_f$, and $\mathbf{v}_0^- - \mathbf{v}_f^- = \mathbf{0}$. Note that this last velocity continuity constraint is computed as the difference between the initial velocity BEFORE $\Delta\mathbf{v}_0$ is applied, and the terminal velocity at the end of the revolution. In addition, for the stated cost index, a successful optimizer run identifies a solution that minimizes $\mathbf{v}_0^+ - \mathbf{v}_f^- = \mathbf{v}_0^- + \Delta\mathbf{v}_0 - \mathbf{v}_f^-$. In the present example, the optimizer is able to identify a solution where $\mathbf{v}_0^- - \mathbf{v}_f^- \approx \mathbf{0}$. This implies that $\mathbf{v}_0^+ - \mathbf{v}_f^- \approx \Delta\mathbf{v}_0$. Subsequently, the total cost required to maintain a controlled periodic relative path over one of the w revolutions is the sum of $\Delta\mathbf{v}_0$, $\Delta\mathbf{v}_1$, and $\Delta\mathbf{v}_2$. In the present example, where $w = 10$, the sum of these individual impulses is roughly 20.14 μm/s.

Note, though one may be tempted to contrast this total cost with the results of the single-impulse examples previously presented, this is not an equitable or adequate comparison. After all, it is not possible to identify a closed relative path

TABLE 3. Three Impulse Solution

Impulse	$\lvert\Delta\mathbf{v}\rvert$ [μm/s]	Time [days]
1	19.2136	0.0000
2	0.3982	58.9803
3	0.5320	116.6191

with a fixed orbital period via a single impulse. At least two maneuvers are required to fix the period to match that of the chief spacecraft path. Thus, although the paths may appear closed and, at times, even nearly periodic, controlled periodicity is NOT achievable via a single impulse. The multiple impulse example presented here, with fixed terminal time and terminal state continuity constraints, guarantees a precisely periodic controlled relative path.

The multiple impulse solution presented here is employed in the next section as an initial guess to a multiple fixed-thrust finite-burn optimization problem. Aside from providing a suitable startup arc for the fixed-thrust finite-burn case, however, this solution also demonstrates the use of knots as a means of incorporating interior $\Delta\mathbf{v}$'s at variable times. Because the knot times are free, and the number of nodes per segment is held fixed, a heterogeneous node distribution is expected. In fact, the node distribution naturally varies from one iteration to the next during the optimization process. Although this does not lead to any convergence difficulties, it is generally advisable to choose n_n sufficiently large such that the longest segment is always adequately represented.

Multiple Fixed-Thrust Finite Burns of Variable Length

In some cases, depending on the duration of the burn and the engine capabilities, it is not appropriate to model maneuvers as impulsive. Furthermore, although low-thrust engine capabilities continue to improve, unconstrained continuous control solutions are often times infeasible for practical applications. The present example considers the same optimization goal of the last two sections, establishing a controlled periodic arc while minimizing the cost. In this case, however, three fixed-thrust magnitude finite burns are implemented through the use of knots. As before, the duration of a finite-burn arc is determined by the difference between the initial and terminal knot times that define the burn segment. This particular formulation seeks to address the possibility that a mission cannot practically accommodate continuous thrusting at all times.

As previously mentioned, the magnitude of the thrust vector is prespecified along each segment. That is, consistent with equation (22), segments are predesignated as either burn segments or coast segments before the optimization process is initiated. The thrust direction at each node and the knot times, which define the duration of each segment, are the control parameters. It is important to note that modeling engine limitations through constraints, such as that in equation (22), must be addressed on a case-by-case basis. The particular formulation selected in equation (22) is simple by design because it is only intended to demonstrate the use of knots in optimal fixed-thrust finite-burn solutions. Specifically, the use of knots allows for an effective way to treat the spatial discontinuities in acceleration that result from the on/off switching of the thrusters. Furthermore, because the direction of the thrust vector is variable over a given segment, it is assumed that the spacecraft has adequate attitude control to reorient itself as needed during each burn. It is also implied, when applicable, that all thrusters are engaged synchronously. That is, all thrusters coast or burn during the same period of time. Although follow-on studies consider fixed independent thrusters with asynchronous on/off thrusting sequences, that is not the subject of the present investigation. However, all subsequent studies build on the knot-based formulation presented here.

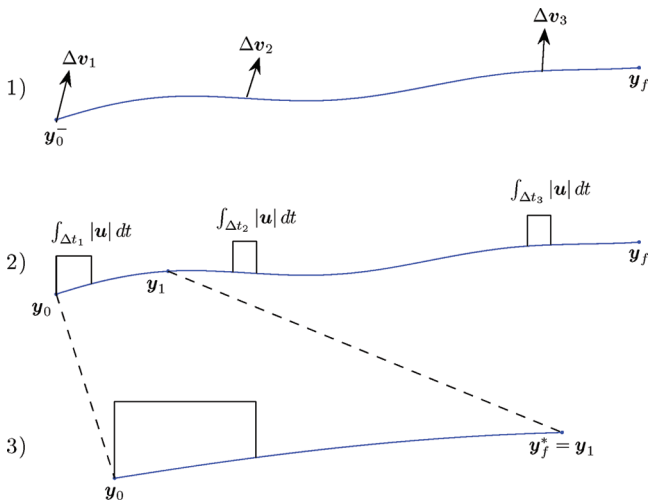

FIG. 8. Process for Finding Fixed-Thrust Finite-Burn Solutions from Impulsive Solutions.

In the present example, the thrust magnitude is constrained to 1 μN for a 500-kg satellite. That is, the specified thrust acceleration becomes

$$a_s = \frac{1\ \mu\text{N}}{500\ \text{kg}} = 2\ \text{nm/s}^2 \tag{32}$$

The control effort is minimized through a cost index of the form

$$J = \int_{t_0}^{t_f} |\mathbf{u}|\,\mathrm{d}t$$

Following the constraint formulation in equation (22), the thrust acceleration magnitude is set to a_s along thrusting arcs (odd-numbered segments) and zero along coasting arcs (even-numbered segments). Subsequently, the cost function reduces to

$$J = \sum_{k=1}^{n_k} |t_k - t_{k-1}| \left(\frac{1}{2} - \frac{1}{2}(-1)^k \right) a_s \tag{33}$$

where $t_{n_k} = t_f$. Notice a quadratic cost function, with integrand $\mathbf{u}^{\mathrm{T}}\mathbf{u}$, would have an equivalent cost function, scaled by a_s^2 instead of a_s.

The optimal impulsive solution presented in Fig. 8 is employed here as a startup arc in the search for the optimal fixed-thrust finite-burn solution. This is accomplished by first extracting the thrust magnitude and direction from each of the individual $\Delta\mathbf{v}$'s listed in Table 3. The magnitude component is used to estimate the duration of an equivalent finite burn. The direction of the impulse is used as an initial guess for the finite burn direction over the segment. The initial guess, then, consists of the number of segments, the duration of each segment, and the direction of thrust during thruster-on segments. Because the impulsive solution has three impulses each followed by a coasting phase, it is logical to set up the fixed-thrust finite-burn problem with $n_s = 6$, with alternating thruster-on and thruster-off segments. To estimate the duration of a burn, it is necessary to establish a relation

between the magnitude of the $\Delta\mathbf{v}$ with respect to the thrust acceleration magnitude, a_s. The relation between these two variables is an integral of the form

$$\Delta v_i \approx \int_{\Delta t_i} |\mathbf{u}|\, dt = \int_{\Delta t_i} a_s\, dt = a_s \Delta t_i$$

Thus, for the startup arc employed here, the initial duration of any thrusting segment is estimated as the ratio of the impulse (i.e., Δv_i) over the thrust acceleration (i.e., a_s). Although this provides an approximation for the initial duration of the burn segments, it is still necessary to identify suitable start times for each fixed-thrust finite-burn segment. Because the optimizer refines the duration of all thrust and coast segments, a suitable initial guess is one that places the start of each burn near the original impulsive counterparts. The initial values for the start and duration of each burn leads to an initial guess for the knot times. Finally, the guesses for the controls at the j^{th} node of the i^{th} thrusting arc are simply

$$\mathbf{u}_j = a_s \frac{\Delta \mathbf{v}_i}{\Delta v_i} \tag{34}$$

That is, the thrust direction along the entire segment (of duration Δt_i) is aligned with $\Delta \mathbf{v}_i$.

This process generates an adequate guess for the second trajectory in Fig. 8, and solutions have been successfully derived through this approach. However, some consideration remains with regard to the scaling of the time variables. Recall, for example, that the magnitude of the first impulse in Table 3 is $\Delta \mathbf{v}_i = 19.2136$ μm/s. The equivalent finite burn duration, for a 2-nm/s^2 control acceleration, is approximately 160 min, or 0.063% of the relative orbital period. Herein lies the problem, the optimization goals require a solution be evaluated over one full orbital period, but the corresponding time scale is significantly disproportionate to the duration of the required thrust arcs. Because the knot times are themselves optimization parameters, and all segments are separated by knots, a burn this short essentially corresponds to two knots that are almost coincident from a numerical perspective. A detailed analysis performed during the course of this study reveals that, without proper variable scaling, the optimizer will generally converge on a local optimal solution with unnecessarily long burn arcs. This naturally leads to more costly solutions due to the extended burn durations.

A method of resolving this inconvenience is to change the scope of the problem, as illustrated in the third trajectory of Fig. 8. Here, a new problem is set up focusing only on one of the burn arcs at a time. From the impulsive solution, the states along the trajectory are known at an arbitrary point, $\mathbf{y}_1$. Let this point become the new final point, $\mathbf{y}_f^*$, with a final time significantly shorter than before. As illustrated, this problem now becomes a two-segment problem only, with a thrusting arc followed by a coasting arc. Note that any more than two segments will result in multiple thrust arcs, and the Δv would be approximated with a series of fixed-thrust finite burns. Although this added flexibility would most likely produce a less costly solution, the objective here is to ensure only one finite burn to account for the Δv.

To demonstrate the optimal fixed-thrust finite-burn approach, the present example focuses on a small section along the first segment of the multi-impulse maneuver solution summarized previously in Fig. 7 and Table 3. Arbitrary states

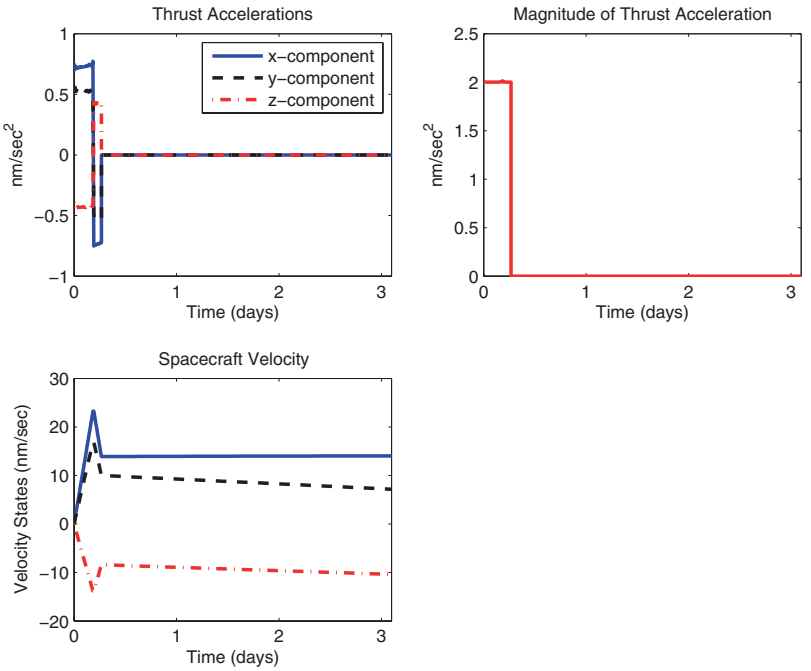

FIG. 9. Fixed-Thrust Finite-Burn Solution with Fixed Terminal State and Time: $\Delta \mathbf{v} = 46.146$ μm/s.

and time are selected and identified as the terminal values. For this example, the states and time selected are associated with the optimal values from the multi-impulse solution, approximately three days after the initial maneuver of magnitude 19.2136 μm/s. The goal of this example is to identify an equivalent fixed-thrust finite burn that converges on this terminal state. Initially, the terminal time is assumed fixed, though this is later determined to be detrimental to the solution process. The control and velocity histories of the converged solution are illustrated in Fig. 9. The position history is not included due to the short time interval considered (three days produces little change in the position).

Some interesting observations are deduced from this result. The control history reveals that the thruster is engaged for 6.409 h. This leads to a total Δv of 46.146 μm/s over that interval. Clearly, this is significantly higher than the cost of the impulsive maneuver in Table 3. A closer look at the individual components of the control acceleration reveals that the thrust vector switches direction near the end of the arc. The spacecraft would have to reorient instantaneously to thrust in the opposite direction, based on the current assumptions. The switch in the thrust direction is manifested in the velocity states by a sharp turn. The source of this behavior originates from the final time constraint. The solution shown is the optimal way of achieving the given final states *within* a specified time limit.

If the final time is free, a burn arc that extends for 2.657 h, with a total Δv of 19.127 μm/s, leads to a solution that is close to the 19.2136 μm/s from the impulsive solution previously presented. The control and velocity histories for this solution are shown in Fig. 10. The lower propulsive cost is naturally attributed to the shorter burn time. Note that the fixed terminal time does not correspond to the engine burn time. The trajectory arc under consideration consists of both an initial burn segment followed by a coast segment. Although the terminal time of the coast

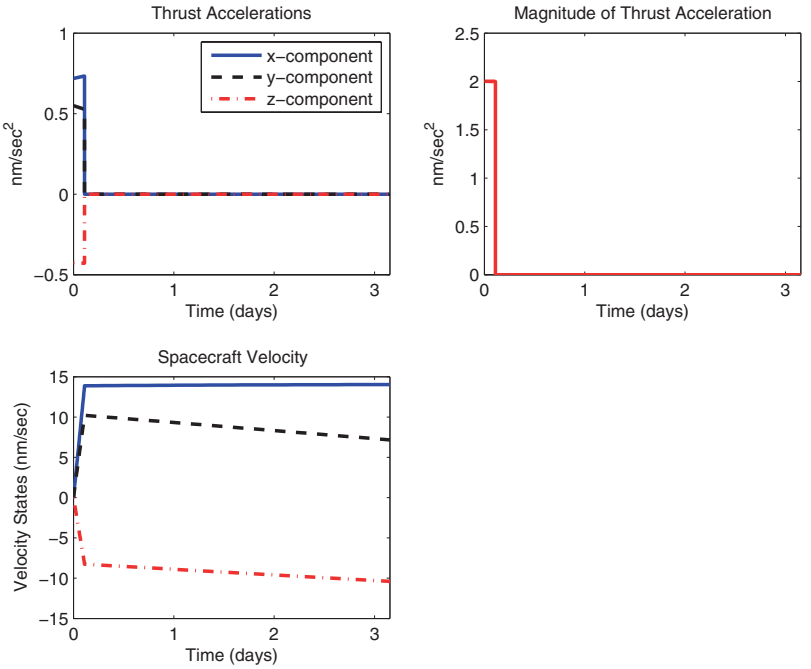

FIG. 10. Fixed-Thrust Finite-Burn Solution with Fixed Terminal State Only: $\Delta \mathbf{v} = 19.127$ μm/sec.

segment is fixed, the interior knot that defines the duration of the burn is still free. Targeting a fixed state and time at the end of the coast segment leads to a higher cost because the spacecraft has to reverse thrusters to "slow down" to meet the terminal time constraint.

The fixed versus free terminal time issue becomes important when the problem is rescoped to consider only one maneuver at a time. Achieving a controlled periodic relative path of the desired geometry entails more than just meeting a terminal state constraint. This is true even in the time-invariant regime of the CR3BP. The computation of the relative path in this example assumes that, at time t_0, the chief starts at a specific point along its orbit. Specifically, the initial state of the chief spacecraft corresponds to a point along the halo where the out-of-plane excursion is at its maximum possible value. The initial state of the deputy is measured relative to this location. After one orbital period, both the chief and the deputy vehicles must return to these fixed initial states for the path to be considered periodic. This is only possible if the period of the relative path of the deputy is equal to that of the halo orbit. This requirement gives rise to the fixed terminal-time constraint. Thus, the proper way to formulate the problem requires, at a minimum, that the multiple fixed-thrust finite-burn scheme be optimized over one fixed orbital period. However, as previously mentioned, some numerical difficulties are encountered with this approach when the duration of the finite burns is on the order of hours while the orbital period is nearly 180 days.

These difficulties are in many ways specific to the sample problem selected to demonstrate the implementation of knots. The initial guess provided corresponds to a natural arc that is nearly periodic but generally drifts with each revolution [12]. Depending on the relative separation of the vehicles, the sensitivity of the dynamical regime leads to maneuvers that are small enough to appear negligible.

However, without the maneuvers, the relative drift can increase quickly over time due to those same sensitivities. In general, and outside any issues specific to the example problems selected here, the formulation presented in this study offers an effective means of incorporating multiple maneuvers, impulsive or constrained finite burns, within a NLP framework.

Conclusions

A formulation is presented that lays the initial algorithmic foundation, within a nonlinear programming framework, for solving optimal control problems involving temporally continuous but spatially discrete control variables. The resulting formulation effectively treats the generalized optimal multiple impulse or multiple bang-off-bang problem for an unspecified number of coast and burn segments. This method is applied to the optimal formation-keeping problem near the libration points of the Sun–Earth/Moon system. Rather than minimizing cost, then, the problems presented are specifically focused on minimizing cost subject to actuator constraints. This is of particular importance for vehicles that evolve near the libration points. Formation flight mission concepts that focus on the region near the libration points typically envision stringent precision tracking requirements due to specified science objectives. Actuation constraints are not typically considered during the conceptual design phase. However, an accurate assessment of the achievable tracking accuracy requires that such constraints be considered during the early design stages. The dynamically sensitive nature of this region of space, combined with physical or mission-imposed limitations on actuation can have a negative impact on the achievable tracking accuracy. The optimal control formulation presented here seeks to address this deficiency by offering analysts a means of easily incorporating reasonably accurate actuator models during the preliminary design phase. Subsequently, analysts can establish—based on the results—reasonable expectations regarding tracking accuracy in the presence of the specified constraints.

References

[1] BETTS, J. *Practical Methods for Optimal Control Using Nonlinear Quadratic Programming*, Philadelphia, Pennsylvania: Society of Industrial and Applied Mathematics, 2001.

[2] SENENT, J., OCAMPO, C., and CAPELLA, A. "Low-Thrust Variable-Specific-Impulse Transfers and Guidance to Unstable Periodic Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 28, March–April 2005, pp. 280–290.

[3] MARCHAND, B.G., HOWELL, K.C., and BETTS, J. "Discrete Nonlinear Optimal Control of S/C Formations Near the L1 and L2 Points of the Sun–Earth/Moon System," Presented as paper AAS 05-341 at the AAS/AIAA Astrodynamics Specialists Conference, Lake Tahoe, California, August 2005.

[4] MILLARD, L. and HOWELL, K. "Control of Interferometric Spacecraft Arrays for (u, v) Plane Coverage inMulti-Body Regimes," *The Journal of the Astronautical Sciences*, Vol. 56, January–March 2008, pp. 71–97.

[5] INFELD, S., JOSSELYN, S., MURRAY, W., and ROSS, I. "Design and Control of Libration Point Spacecraft Formations," *Journal of Guidance, Control, and Dynamics*, Vol. 30, July–August 2007, pp. 899–909.

[6] HUGHES, S. "General Method for Optimal Guidance of Spacecraft Formations," *Journal of Guidance, Control, and Dynamics*, Vol. 31, March–April 2008, pp. 413–423.

[7] GILL, P., MURRAY, W., and SAUNDERS, M. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131.

[8] MARCHAND, B.G., and HOWELL, K.C. "Formation Flight Near L1 and L2 in the Sun–Earth/Moon Ephemeris System Including Solar Radiation Pressure," Presented as paper AAS 03-596 at the AAS/AIAA Astrodynamics Specialists Conference, Big Sky, Montana, August 2003. AAS Paper 03-596.

[9] MARCHAND, B. and HOWELL, K. "Control Strategies for Formation Flight in the Vicinity of the Libration Points," *Journal of Guidance, Control, and Dynamics*, Vol. 28, November–December 2005, pp. 1210–1219.

[10] HOWELL, K. and MARCHAND, B. "Design and Control of Formations Near the Libration Points of the Sun–Earth/Moon Ephemeris System," Goddard Space Flight Center: Space Flight Mechanics Symposium, Greenbelt, Maryland, October 2003.

[11] HOWELL, K. and MARCHAND, B. "Formations Near the Libration Points: Design Strategies Using Natural and Non-Natural Arcs," Goddard Space Flight Center: 2nd International Symposium on Formation Flying Missions and Technologies, Greenbelt, Maryland, September 2004.

[12] HOWELL, K. and MARCHAND, B. "Natural and Non-Natural Spacecraft Formations Near the L1 and L2 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*, Vol. 20, March 2005, pp. 149–173.

[13] BERTSEKAS, D.P. *Nonlinear Programming*. Belmont, Massachusetts: Athena Scientific, 2 ed., 1999.

[14] KIRK, D.E. *Optimal Control Theory: An Introduction*, Englewood cliffs, NJ: Prentice-Hall, Inc. 1970.

[15] BRYSON, A. and HO, Y. *Applied Optimal Control: Optimization, Estimation, and Control*, New York: Hemisphere Pub. Corp., 1975.

[16] BETTS, J.T. and ERB, S.O. "Optimal Low Thrust Trajectories to the Moon," *SIAM Journal on Applied Dynamical Systems*, Vol. 2, May 2003, pp. 144–170.

[17] BETTS, J.T. "Very Low Thrust Trajectory Optimization Using a Direct SQP Method," *Journal of Computational and Applied Mathematics*, Vol. 120, August 2000, pp. 27–40.

[18] BETTS, J.T. "Optimal Interplanetary Orbit Transfers by Direct Transcription," *The Journal of the Astronautical Sciences*, Vol. 42, July–September 1994, pp. 247–268.

[19] BETTS, J.T. "Using Sparse Nonlinear Programming to Compute Low Thrust Orbit Transfers," *The Journal of the Astronautical Sciences*, Vol. 41, July–September 1993, pp. 349–371.