AAS 09-121

# FINITE SET CONTROL TRANSCRIPTION FOR OPTIMAL CONTROL APPLICATIONS

Stuart A. Stanton[*][†] and Belinda G. Marchand [‡]

Previous efforts explore an enhanced collocation method designed to treat optimal control applications in which control variables are constrained to finite sets of values. Presently, the method is applied to several aerospace control problems to demonstrate its utility and capability. On-off actuation schemes are ideally represented with constrained control. The behavior of variable-thrust actuators is modeled by limiting control change rates to a finite space. Solutions derived are characterized as optimal switching schedules between feasible control values. The methodology allows control switches to be determined over a continuous spectrum, overcoming many of the limitations associated with discretized solutions.

## INTRODUCTION

Prevalent in many engineering fields are systems composed of interdependent continuous and discrete components or variables. Although processes exist in nature that are accurately modeled with only continuously-varying dynamics, it is often the case that some level of decision-making occurs in the process. The decision is made by any number of sources, from natural to man-made technologies, but it is clear that the selection is among a discrete number of options. Thus, a hybrid system results, exhibiting continuously-varying and discretely-chosen components. It is observed in the literature that this often takes on a hierachical structure, where continuous or time-driven dynamics exist at the lower levels and discrete or event-driven dynamics exist at the higher levels.[1,2]

The hybrid control problem motivates methods for determining continuous and discrete control variables that affect a hybrid system. The challenge clearly lies in the dual structure of the problem; although strictly continuous and strictly discrete control methods exist, there is room for developing the methods that treat hybrid systems in the sense of stability and optimality. Existing techniques generally extend the theory of either continuous or discrete systems, using, for example, Lyapunov theory,[3,4,5] classical optimal control theory,[2,6] or combinatorial techniques.[1,7]

One of the common limitations in existing methods is that they are only able to consider the case of a single discrete variable. That is, at any given time, there is only one decision to make (from some number of options). Additionally, methods have been developed to treat special cases or specific classes of problems. Ideally, methods may be developed that produce results for a broad range of system scenarios, with any number of separate decision variables.

In earlier studies,[8] a new method for treating a class of hybrid control problems involving discrete control variables is introduced. The implementation for the Finite Set Control Transcription (FSCT) method is presented in detail, and its capability is sampled. The objective of the current investigation is to further explore the capability and utility of the method by demonstrating a range of applications. In so doing, the scope of the method is characterized, ideally inspiring additional applications outside those presented here. The applications presented in this investigation are focused on aerospace systems, as these served to motivate the method's development.

[*] Capt, USAF; Ph.D. Candidate, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, 1 University Station, C0600, Austin, TX 78712-0235.

[†] The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

[‡] Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, 1 University Station, C0600, Austin, TX 78712-0235.

### General Problem Statement

The hybrid system under consideration for this investigation is governed by the dynamics

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{u}), \tag{1}$$

where the vector, $\boldsymbol{y} \in \mathbb{R}^{n_y}$, represents continuous state variables, and $\boldsymbol{u}$ consists of $n_u$ control elements limited to finite values as

$$u_i \in \mathbb{U}_i = \{\tilde{u}_{i,1}, \ldots, \tilde{u}_{i,m_i}\}.$$

The function, $\boldsymbol{f}$, describes the continuous variation of the states in terms of time, $t$, and the present values for each state and control.

At first glance, this formulation appears to limit the method to a specific class of hybrid systems: all states are presented as continuous, all controls discrete. Thus, systems with discrete states or continuous controls are apparently excluded. Previously, the implementation carried this limitation. However, the FSCT method can be tailored to include continuous state and control variables within $\boldsymbol{y}$ and, likewise, discrete states and controls in $\boldsymbol{u}$ to allow for a more general treatment of hybrid systems. The necessary adjustments, however, are specific to the system under consideration, and therefore that aspect is not discussed here. However, it is observed that many control variables traditionally modeled as continuous may be more accurately described by a combination of continuous dynamic states and discrete controls. This characteristic is demonstrated later in this document. Thus, the formulation of Equation 1 is not necessarily restrictive. For continuity and clarity, in this study the term *state* implies a continuous variable, while *control*, a discrete one.

This paper is presented as a series of applications designed to demonstrate the capability of the FSCT method. Throughout, results are compared with those produced using alternative hybrid control methods to articulate particular advantages or ways in which multiple methods can be used in tandem.

- The stability of a switched linear system is considered. In each control mode, the system is stable, but certain switching structures result in instability. The FSCT method is contrasted with a technique involving *Multiple Lypunov Functions*.
- Minimum-time and minimum-fuel optimizations for a simple system in two dimensions is reviewed. An FSCT optimal solution is implemented by a real-time *Model Predictive Control* law.
- Attitude control for a small spacecraft using inexpensive cold-gas thruster technology is explored. The FSCT method is used for trajectory tracking with fixed thrust and variable thrust dynamic schemes.

Background knowledge of the FSCT method is necessary to the following demonstration. Thus, this document proceeds with an overview of the methodology presented previously.[8]

## FSCT METHOD OVERVIEW

The Finite Set Control Transcription is a formulation of the hybrid optimal control problem as a parameter optimization problem that can be solved using a standard Nonlinear Programming (NLP) algorithm, such as SNOPT.[9] The following overview is intended for the reader possessing a general understanding of direct optimization techniques for continuously-varying parameters. Note that, although the method demonstrated here is rooted in direct collocation, alternative formulations exist that capitalize on the structure of indirect or direct shooting methods.

In the most basic sense, the object of a transcription formulation is to convert the optimal control problem formulated as,

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

subject to

$$\begin{aligned}
\dot{\boldsymbol{y}} &= \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{u}), \\
\boldsymbol{0} &= \boldsymbol{\psi}_0(t_0, \boldsymbol{y}_0), \\
\boldsymbol{0} &= \boldsymbol{\psi}_f(t_f, \boldsymbol{y}_f), \\
\boldsymbol{0} &= \boldsymbol{\beta}(t, \boldsymbol{y}, \boldsymbol{u}),
\end{aligned}$$

into an NLP problem of the form,

$$\text{Minimize } F(\boldsymbol{x}) \tag{2}$$

subject to

$$\boldsymbol{c}(\boldsymbol{x}) = \left[\boldsymbol{c}_{\dot{y}}^T(\boldsymbol{x})\ \boldsymbol{c}_{\psi_0}^T(\boldsymbol{x})\ \boldsymbol{c}_{\psi_f}^T(\boldsymbol{x})\ \boldsymbol{c}_{\beta}^T(\boldsymbol{x})\right]^T = \boldsymbol{0}. \tag{3}$$

Ultimately, $\boldsymbol{x}$ must contain the information necessary to express $\boldsymbol{y}(t)$ and $\boldsymbol{u}(t)$ for $t \in [t_0\ t_f]$. In the resulting NLP problem, an initial guess for $\boldsymbol{x}$ is iterated upon until arriving at a feasible and locally optimal set of values. Note that each problem has a cost function to minimize as well as constraints for the dynamics, initial and final conditions, and any path constraints imposed on the system. In the above problem definitions, all constraints are presented as equalities, however, extensions certainly exist for inequality constraints, as well. The nature of the transcription formulation dictates both the definition of the parameter vector, $\boldsymbol{x}$, and the number and forms of the constraint functions in $\boldsymbol{c}(\boldsymbol{x})$ in the resulting parameter optimization problem. The details of how constraint functions are generated is outside the scope of the current development, although this process is articulated in the previous investigation. Let it suffice here to present the definition of $\boldsymbol{x}$ as optimized in the FSCT formulation, knowing that it is possible to devise Equations 2 and 3 to ensure that the original optimal control problem is well represented.

Consider the following definition of the parameter vector used for an optimization with the FSCT method.

$$\boldsymbol{x} = [\cdots\ y_{i,j,k}\ \cdots\ \cdots\ \Delta t_{i,k}\ \cdots\ t_0\ t_f]^T \tag{4}$$

The vector, $\boldsymbol{x}$, contains parameters that represent states, $y_{i,j,k}$, and times $\Delta t_{i,k}$, $t_0$, and $t_f$. One of the key features of this parameterization is that control variables are not among the parameters to be optimized. This is unusual: most collocation and direct shooting methods optimize parameters that directly represent control variables. However, in this case, a unique parameterization is necessary since the controls are discrete variables, while the elements of $\boldsymbol{x}$, by the nature of nonlinear programming, are necessarily treated as continuous variables (although perhaps bounded and subject to constraints). Demonstrated presently, a control history is completely defined by the time elements in the parameter vector.

Let the trajectory defined from initial time, $t_0$, to final time, $t_f$, be broken up into $n_s$ segments. The interior separation times between segments are termed *knots*. These represent instances of time when the discrete control variables switch from one feasible value to another. Suppose each control variable is allowed $n_k$ switches between $t_0$ and $t_f$. The result is that $n_s = n_u n_k + 1$, and each control is held constant over each segment.

Define $n_n$ as the number of nodes per segment. A *node* is a point in time at which the values of the state variables are contained within the parameter vector. Specifically, element $y_{i,j,k}$ in Equation 4 represents the $i^{\text{th}}$ state at the $j^{\text{th}}$ node of the $k^{\text{th}}$ segment. Then $\boldsymbol{x}$ contains $n_y n_n n_s$ elements pertaining to all of the states at each node. These state values are used directly in the cost and constraint Equations 2 and 3.

The elements $\Delta t_{i,k}$ in $\boldsymbol{x}$ indicate the elapsed time between two control switches for a given control variable. Specifically, $\Delta t_{i,k}$ indicates the amount of time that passes between the control switches at the $(k-1)^{\text{th}}$ and $k^{\text{th}}$ knots for the $i^{\text{th}}$ control variable.

The values for each $u_i$ are *pre-specified* between each switching point. Thus, $u_{i,k}^*$ indicates the pre-specified value of the $i^{\text{th}}$ control variable before the $k^{\text{th}}$ knot. With a discrete number of feasible values, it is possible to set $n_k$ large enough such that each possible control value is designated as the actual control value for some duration. During the optimization, the values of $\Delta t_{i,k}$ are determined, indicating the amount of time (possibly, zero) that each control value is maintained.

The transcription definition is best interpreted with a visualization, such as Figure 1. In this conceptualization, consider the hybrid control problem with $n_y = 2$ states and $n_u = 2$ controls, where $\mathbb{U}_1 = \{1, 2, 3\}$ and $\mathbb{U}_2 = \{-1, 1\}$. Next, assume the transcription is selected such that $n_n = 4$ nodes per segment and $n_k = 5$ switching points per control variable. Thus, the number of segments is $n_s = (2)(5) + 1 = 11$ segments.

It is apparent from Figure 1 that each control variable may take up to $n_k + 1 = 6$ different values over the trajectory duration. Arbitrarily, the control values are pre-specified so that each control variable

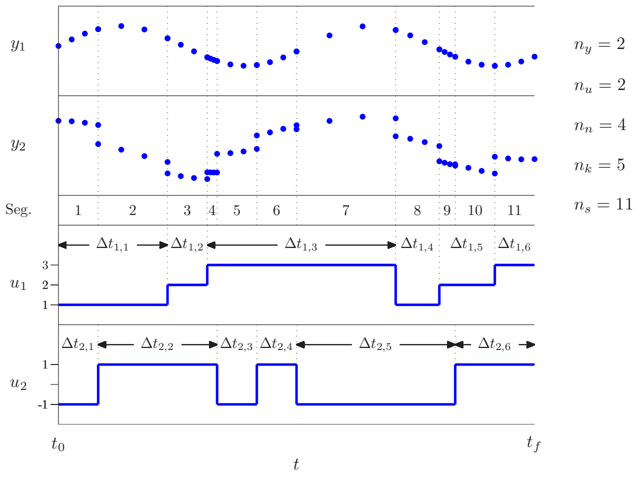


**Figure 1. The Parameters of $\boldsymbol{x}$**

systematically switches between the feasible values for that variable. Note that some feasible control values may not be members of the optimal solution. However, through the NLP algorithm, the time durations between switching points are optimized. If one of the pre-specified control values is unnecessary or non-optimal, then the value of the respective time duration is reduced to zero.

Figure 1 further illustrates that the node distribution is not necessarily uniform over the interval $[t_0 \; t_f]$. The duration of each segment is dictated by the current values of $\Delta t_{i,k}$. The $n_n = 4$ nodes per segment are evenly distributed over a segment, but for shorter segments, this means a closer spacing between nodes. Thus, the state values contained in $\boldsymbol{x}$ may pertain to dense or sparse regions, depending on the time parameters in $\boldsymbol{x}$.

It is also important to note that two nodes are associated with a given knot: the terminal node from the preceding segment and the initial node from the following segment. Therefore, in this parameterization, two sets of state values are contained in $\boldsymbol{x}$ for the times at each knot. For a feasible solution, continuous state variables will have identical values at simultaneous nodes. Constraints in $\boldsymbol{c}(\boldsymbol{x})$ are included to enforce continuity across segments. Of course, these constraints are not always satisfied on intermediate iterations of the solution process. For example, in Figure 1, the states $y_2$ are not continuous. Subsequently, this $\boldsymbol{x}$ does not represent a feasible solution. During the FSCT optimization process, elements of $\boldsymbol{x}$ are updated to ensure that, upon completion, the continuity constraints are satisfied.

Additional constraints are included in $\boldsymbol{c}(\boldsymbol{x})$ to ensure that

$$0 = t_f - t_0 - \sum_{k=1}^{n_k+1} \Delta t_{i,k}, \; i = 1, \ldots, n_u.$$

Also, at all times, $\Delta t_{i,k} \geq 0$ so that there are no negative time intervals.

By pre-specifying the control values, a collocation transcription results in which control switching times are optimized to indicate an optimal control history over all of the feasible control values. Multiple control variables are easily managed and treated completely independently. The control variables for a given segment necessarily affect the hybrid system dynamics, and they are included in appropriate constraint equations for that segment. As the NLP algorithm searches for a feasible and locally optimal set of parameters, the state values are modified at each node so that, upon completion, the state and control histories represent a matching, feasible trajectory.

The total number of feasible values for a control variable, $m_i$, should have a significant effect on the choice of $n_k$, the number of switching points allowed over the trajectory. Clearly, when $n_k \gg \max(m_i)$, it is possible to pre-specify each control value over several time durations, allowing more flexibility in the resulting NLP problem and a greater likelihood to converge on a small local minimum. However, as $n_k$ gets larger, the sizes of $\boldsymbol{x}$ and $\boldsymbol{c}(\boldsymbol{x})$ also increase, which may complicate, or at least slow down, the optimization process. This characteristic indicates the primary limitation of the FSCT method. In order to perform an optimization, a user must specify $n_k$, thus limiting the number of control switches to some maximum value.

In practice, it is useful to *overparameterize* a problem by setting $n_k$ to an arbitrarily high value, allowing for more control switches than are ultimately necessary. Overparameterizing allows the optimizer to demonstrate the optimal number of switches (less than the parameterized number) by driving to zero the duration of superfluous control values. The overparameterization also allows the user additional flexibility to arbitrarily pre-specify control values, knowing that non-optimal control values are ultimately eliminated in the final solution. Indeed, in the examples that follow, the concept of overparameterization is employed. Consequently, ensuing solutions may display features that are ultimately artifacts of the parameterization. For example, two knots may occur simultaneously, appearing as though the control switches from one value to another and then instantaneously to a third. In the parameterization, zero-duration segments are present, indicating that particular pre-specified control values are effectively eliminated from the solution.

## TWO STABLE LINEAR SYSTEMS

Although the FSCT method is especially effective for multiple control variables, consider first a system controlled by only one decision. The system is

$$\begin{aligned} \dot{\boldsymbol{y}} &= \boldsymbol{f}(\boldsymbol{y}, u) = \boldsymbol{A}_u \boldsymbol{y}, \quad (5) \\ u &\in \{1, 2\}, \quad (6) \end{aligned}$$

$$\boldsymbol{A}_1 = \begin{bmatrix} -1 & 10 \\ -100 & -1 \end{bmatrix}, \qquad \boldsymbol{A}_2 = \begin{bmatrix} -1 & 100 \\ -10 & -1 \end{bmatrix}.$$

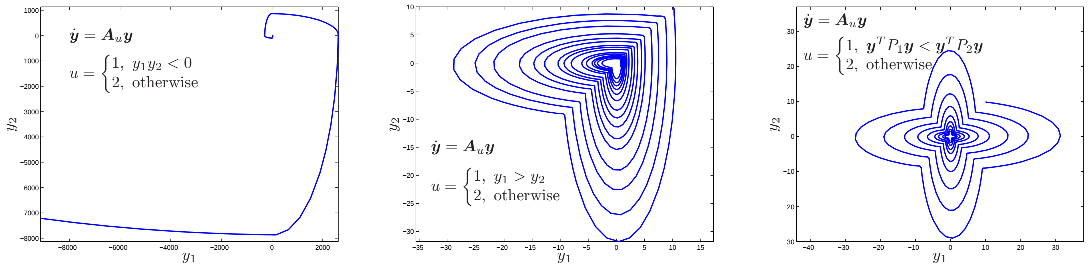
Thus, the system is characterized by two separate dynamical modes, and the decision variable determines which of the two is in play at any given time. Notice that individually, each mode is a linear, time-invariant system guaranteeing exponential stability at the origin, $\boldsymbol{y} = \boldsymbol{0}$.

This example is presented by Branicky[3,4] as a classical demonstration of how multiple Lyapunov functions can be used to develop switching laws for the system. It is intriguing in that, although individually stable, one cannot arbitrarily switch between dynamical modes and guarantee system stability. Branicky shows, for example, a switching law devised such that $u = 1$ when $\boldsymbol{y}$ is in Quadrants 2 and 4, and $u = 2$ when $\boldsymbol{y}$ is in Quadrants 1 and 3. From any point, the trajectory goes to infinity as illustrated in Figure 2(a). However, this is not the case for all switching functions. For example the law that switches modes when $\boldsymbol{y}$ crosses the line $y_2 = y_1$ results in a stable system converging on the origin (Figure 2(b)).

The characteristics of the system can be explained via Lyapunov analysis, which follows. The technique of multiple Lyapunov functions is intuitively applied since the switched system consists of multiple dynamical modes. Subsequently, the FSCT method is applied to demonstrate an alternative analysis technique for determining stable (and optimal) switching laws. This system presented in Equations 5-6 serves as an excellent example, since each method can be exercised in a graceful manner due to the inherent simplicity of the linear system. In addition, this example capitalizes on the familiarity of linear systems and Lyapunov stability theory to the general reader.

### Stability via Multiple Lyapunov Functions

The key feature of the switching law of Figure 2(b) that guarantees stability is that the system remains in both modes for exactly one half of a revolution between each switch. Recall that the two state linear system



(a) (b) (c)

**Figure 2. Three Switching Laws**

with complex eigenvalues $\lambda_{1,2} = \alpha \pm j\omega$ and corresponding eigenvectors $\boldsymbol{v}_{1,2} = \boldsymbol{a} \pm j\boldsymbol{w}$ has solution of the form

$$\boldsymbol{y}(t) = e^{\boldsymbol{A}t}\boldsymbol{y}_0 = e^{\alpha t}\begin{bmatrix} \boldsymbol{a}\ \boldsymbol{w} \end{bmatrix}\begin{bmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{bmatrix}\begin{bmatrix} \boldsymbol{a}\ \boldsymbol{w} \end{bmatrix}^{-1}\boldsymbol{y}_0.$$

Then, one half revolution later from any point,

$$\boldsymbol{y}\left(t + \frac{\pi}{\omega}\right) = e^{\alpha\left(t + \frac{\pi}{\omega}\right)}\begin{bmatrix} \boldsymbol{a}\ \boldsymbol{w} \end{bmatrix}\begin{bmatrix} \cos\omega\left(t + \frac{\pi}{\omega}\right) & \sin\omega\left(t + \frac{\pi}{\omega}\right) \\ -\sin\omega\left(t + \frac{\pi}{\omega}\right) & \cos\omega\left(t + \frac{\pi}{\omega}\right) \end{bmatrix}\begin{bmatrix} \boldsymbol{a}\ \boldsymbol{w} \end{bmatrix}^{-1}\boldsymbol{y}_0 = -e^{\alpha\frac{\pi}{\omega}}\boldsymbol{y}(t),$$

provided that the system remains in the same mode over that time. Thus, for $\alpha < 0$ (a stable system), the function $V = \boldsymbol{y}^T\boldsymbol{y}$, which represents in a sense the energy of the system, is guaranteed to be smaller after one half of a revolution. Consistent switching at intervals of $\frac{\pi}{\omega}$ ensures an incremental decrease in system energy, resulting in convergence to the origin.

Other stable switching structures may also be obtained with a more classical Lyapunov argument. Considering each stable dynamical mode, $\boldsymbol{A}_u$, separately, there exist symmetric positive definite matrix pairs, $\boldsymbol{P}_u$ and $\boldsymbol{Q}_u$, such that

$$\boldsymbol{P}_u\boldsymbol{A}_u + \boldsymbol{A}_u^T\boldsymbol{P}_u = -\boldsymbol{Q}_u. \tag{7}$$

Stability for the mode is demonstrated through the Lyapunov function,

$$V_u = \boldsymbol{y}^T\boldsymbol{P}_u\boldsymbol{y} > 0,$$

with negative time derivative,

$$\dot{V}_u = \boldsymbol{y}^T\boldsymbol{P}_u\dot{\boldsymbol{y}} + \dot{\boldsymbol{y}}^T\boldsymbol{P}_u\boldsymbol{y} = -\boldsymbol{y}^T\boldsymbol{Q}_u\boldsymbol{y} < 0.$$

This standard analysis method offers a way of defining a stable switching law according to the behavior of the Lyapunov functions for each mode. For example, define $\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \boldsymbol{I}$ for simplicity. Then the Lyapunov Equation 7 can be solved uniquely to yield $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, corresponding to their respective modes. In this case, it is observed that regardless of the current mode, the energy of the system decreases according to $-\boldsymbol{y}^T\boldsymbol{Q}_u\boldsymbol{y} = -\boldsymbol{y}^T\boldsymbol{y}$. However, $V_1 \neq V_2$, and a reasonable switching law can be selected such that the Lyapunov function is minimized.[5] Thus,

$$u = \begin{cases} 1, & V_1 \leq V_2 \\ 2, & V_1 > V_2 \end{cases}.$$

A trajectory implementing this switching law is illustrated in Figure 2(c).

### Optimal Switching via FSCT Method

The method of multiple Lyapunov functions demonstrated above can be effective in determining switching strategies between a finite number of system modes identified through a single decision variable. Variations on the theme arise by choosing the minimum $\dot{V}_u$ instead of $V_u$ for some candidate Lyapunov functions, or by minimizing some combination of the two.[5] With an infinite set of stable switching laws, a question remains regarding efficiency and robustness. Although many criteria may be chosen to rank the effectiveness of a switching structure, a simple criterion is presently selected to demonstrate how the FSCT method can aid in the realization of an appropriate switching law. For this example, consider the objective of minimizing the time needed to move a point from its initial position to the vicinity of the origin. Naturally, the trajectory will never go through the origin, as $e^{\alpha t} > 0$ always. However, by choosing a region near the origin, a terminal condition for the optimal control problem is established. Let the final point be subject to

$$\boldsymbol{y}_f^T \boldsymbol{y}_f = 1,$$

such that the objective is to cross the boundary of the unit circle in minimum time, starting from the initial point, $\boldsymbol{y}_0 = [10\ 10]^T$. The optimal control law indicates when to switch between the dynamical modes of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ to most efficiently guide the trajectory to the terminal manifold.

The FSCT method is well equipped to solve this optimal control problem. Actually, many of the unique characteristics of the solution method are not exercised by this example due to the fact that the problem consists of only one decision variable. The total number of segments is exactly the number of pre-specified control values, and consequently, the control characteristics of each segment are known a priori. Thus, the optimization process simply determines appropriate switching times between segments.

To begin the process, a user selects the number of knots, indicating the total allowable control switches over the course of the trajectory. Let $n_k = 20$ knots for an initial optimization, and pre-specify control values such that

$$u_k^* = \frac{3}{2} + \frac{1}{2}(-1)^k,$$

indicating that $u$ begins at the value 1 and alternates between 1 and 2 over each of $n_k + 1 = 21$ segments. Additionally, a user selects a node count that sufficiently captures the state dynamics between control switches when state continuity conditions are satisfied. For this example, $n_n = 100$. Appropriate knot conditions are identified to ensure state continuity across segments, and the optimization function, $\mathcal{J} = F(\boldsymbol{x}) = t_f - t_0$, completes the FSCT formulation.

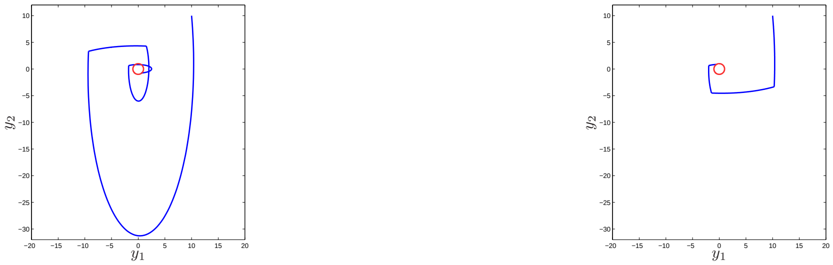
A preliminary guess is necessary to conduct the nonlinear optimization. The initial point, $\boldsymbol{x}_0$, is generated using an interpolation of the trajectory determined by the minimum Lyapunov function switching law of Figure 2(c) over the time interval $t \in [0\ 3]$. Thus, the preliminary cost of the optimization is 3, a reasonable estimate considering this trajectory first crosses the unit circle at time $t = 3.17$. The knots (switching times) between dynamical modes are uniformly spaced in time from 0 to 3. Thus, the preliminary guess does not satisfy the continuity constraints: the guessed control switches do not correspond to the control switches of the interpolated states. This is acceptable, as the optimization process ensures that the final solution is feasible as well as locally optimal.

An FSCT optimization applied for the selected initial guess leads to the trajectory illustrated in Figure 3(a). The final time is $t_f = 0.3782$, significantly smaller than the initial guess. The solution is feasible, and three control switches are clearly observable by the corners in the trajectory. With 20 knots, then, it is apparent that 17 possible control switches are not utilized. Indeed, the solution consists of many knots occurring simultaneously, resulting in zero-duration segments. Thus, the transcription is overparameterized for this solution. Observe that the control switches occur at the following states:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_{k=1} = \begin{bmatrix} -9.3125 \\ 3.3180 \end{bmatrix}, \quad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_{k=2} = \begin{bmatrix} 1.5297 \\ 4.2932 \end{bmatrix}, \quad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_{k=3} = \begin{bmatrix} -1.7921 \\ 0.6385 \end{bmatrix}.$$

Notice that the switching points are related, as

$$-\left(\frac{y_1}{y_2}\right)_{k=1} = \left(\frac{y_2}{y_1}\right)_{k=2} = -\left(\frac{y_1}{y_2}\right)_{k=3} \equiv m.$$



(a)

(b)

**Figure 3. FSCT Locally Optimal Switching Trajectories**

This ratio *implies* the switching law

$$u = \begin{cases} 1, & -\frac{1}{m} \le \frac{y_2}{y_1} \le m \\ 2, & \text{otherwise} \end{cases} \tag{8}$$

where $m = 2.8067$. It is important to observe that the trajectory that resulted from the FSCT optimization shows control switches in only Quadrants 1 and 2, while the control law in Equation 8 observes switches in each of the four quadrants. The difference is explained through the realization that the solution method is capable of determining *locally* optimal solutions, most likely in the vicinity of the initial guess. Obviously, the solution is only a local minimum, as the trajectory actually crossed the terminal radius at one point before the final time. In this case, the initial guess and the parameterization leads to a solution with only three observable control switches. However, the symmetry of the trajectories generated in either dynamical mode imply that there should also exist symmetry in the switching law. This intuition leads to Equation 8. To validate this control law, a second optimization problem is solved, this time with a new initial guess. For the second optimization, the states and control switch times of the initial guess are generated using Equation 8. Optimizing this initial guess, the trajectory of Figure 3(b) is determined. Validating the control law, this second solution corresponds perfectly to the initial guess, *except* that in the final solution, $m = 3.0979$, a slightly larger slope for the switching line. However, the cost is even further improved, with $t_f = 0.0870$.

The fact that the slope value, $m$, changes between the two optimizations is not overly surprising. One reason for this is simply that, in each case, the solution is a local, not global, minimum. Through further analysis, it is apparent that $m$ is a factor of the initial point and the size of the terminal radius, as well. Indeed, a change to the terminal radius such that $\boldsymbol{y}_f^T \boldsymbol{y}_f = 0.5$ yields $m = 3.7786$ in the optimal solution.

The intent of this example is to demonstrate how a classical problem, which can be solved using traditional control techniques, can also be analyzed using the FSCT method. One advantage of the latter is the ability to optimize a control solution or control law according to a specified objective. In this case, the final time is minimized, however it might be equally useful to minimize the integral of the system energy over a fixed time, for example. Both costs capture, in a sense, the sentiment to drive a trajectory to the origin in an efficient manner, although both undoubtedly yield different solutions. It is observed, after all, that the trajectories of Figure 3 reach the unit circle quickly, but their control law does not guarantee that the trajectory will remain within that circle for all future time (it may escape the region and re-enter). Thus, the FSCT method can only guarantee optimality over the range of time considered, not beyond.

## LUNAR LANDER

In a second example, it is useful to revisit the classical Lunar Lander problem explored in the prequel and in many sources on optimal control theory.[10,11,12] The system has 4 states and 2 control variables, in a sense doubling the complexity of the previous example. Specifically, by implementing multiple control values, the unique segment-switching characteristics of the FSCT method can be observed while maintaining problem simplicity and familiarity. The objective of the problem is to transfer a rocket from a lunar orbit to the lunar surface in minimum time or by using minimum fuel. The dynamics are constructed in two dimensions, where separate fixed-magnitude thrusters are pointed in the principal directions. The dynamics are described by

$$\dot{\boldsymbol{y}} = \begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ u_1 \\ -g + u_2 \end{bmatrix},$$

where $r$, $v$, and $u$ represent position, velocity, and control acceleration, respectively, and the subscripts indicate the horizontal and vertical dimensions. A gravitational constant of $g = 1.6231$ m/s$^2$ is utilized. With initial conditions, $\boldsymbol{r}_0 = [200\ 15]^T$ km and $\boldsymbol{v}_0 = [-1.7\ 0]^T$ km/s and final conditions $\boldsymbol{r}_f = \boldsymbol{v}_f = \boldsymbol{0}$, the lander must achieve a soft landing on a specified target from a completely specified initial state. Both minimum-time and minimum-fuel optimizations are realized with the finite set control constraints, $u_1 \in \{-\tilde{u}_1,\ 0,\ \tilde{u}_1\}$ and $u_2 \in \{-\tilde{u}_2,\ 0,\ \tilde{u}_2\}$, where $\tilde{u}_1 = 50$ m/s$^2$ and $\tilde{u}_2 = 20$ m/s$^2$. The control constraints ensure constant thrust acceleration during thrusting arcs.

### Optimal Minimum-Time and Minimum-Fuel Solutions

Optimal solutions are now demonstrated via the FSCT method. For this example, let $n_n = 5$ nodes per segment and $n_k = 14$ knots per control axis. In addition, let the pre-specified controls be identified as

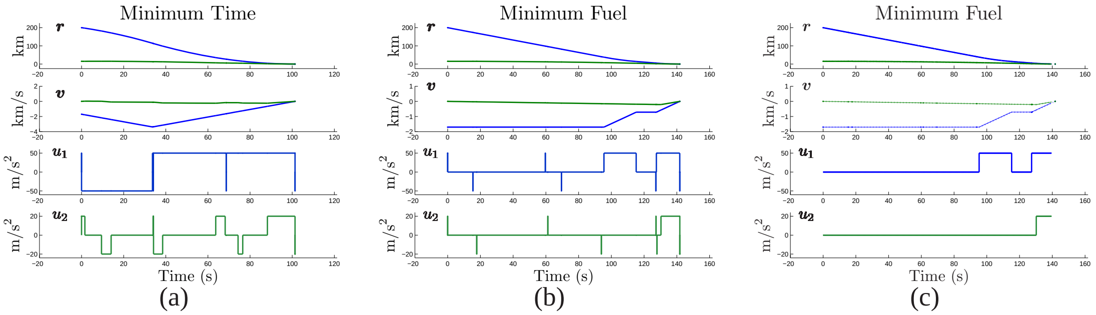
$$u^*_{i,k} = \tilde{u}_i \cos\left(\frac{\pi}{2}(k-1)\right).$$

Thus, it is assumed in the control sequence that the vehicle thrusts initially in the positive directions (uprange and up), then coasts, then thrusts in the negative directions (downrange and down). The resulting optimizations determine the appropriate times for all control switches, indicating the durations for each thrusting and coasting arc.

An initial guess is devised with $t_0 = 0$, $t_f = 300$ seconds, and all knot times are evenly distributed over the interval such that each segment duration is identical. The state parameters in $\boldsymbol{x}$ are constructed to create a linear progression in each state from its initial value to its final value. Initial, final, and knot condition constraints are satisfied by the $\boldsymbol{x}$ supplied to the optimizer before the first iteration, but continuity constraints are not immediately satisfied. During the optimization process, $\boldsymbol{x}$ is improved such that *all* constraints are satisfied. In addition, the final $\boldsymbol{x}$ minimizes the objective function, representing $\mathcal{J} = t_f - t_0$ for minimum time or $\mathcal{J} = \int_{t_0}^{t_f} \boldsymbol{u}^T \boldsymbol{u}\, dt$ for minimum fuel.

Figure 4 displays the solutions of both the minimum-time and minimum-fuel problem. Vehicle positions, velocities, and controls are plotted for both minimizations. Notice the control history $u_1$ for the minimum-time solution. In essence, this solution represents bang-bang control in the first axis, with $u_1(t) = -\tilde{u}_1$ on $t \in [0\ 33.66]$ seconds, and $u_1(t) = \tilde{u}_1$ for the remaining time until $t_f$, at 101.32 seconds. Of course, this control behavior is expected for a minimum-time optimization. Recall, however, that the pre-specified initial value for $u_1$ is $\tilde{u}_1$. As the illustration demonstrates, there is an instantaneous switch in the control at $t_0 = 0$ from $\tilde{u}_1$ to $0$ and then from $0$ to $-\tilde{u}_1$. The solution exhibits that $\Delta t_{1,1} = \Delta t_{1,2} = 0$ in order to accomplish this. In addition, there are instantaneous switches at $t =$ 34.06, 68.65, and 101.32 seconds. At each of these times there exist time durations $\Delta t_{1,k}$ for coasting and negative-direction thrusting, and each has been optimized to be identically zero. This behavior is a common artifact of the FSCT formulation. It does not indicate that control switches should occur at these times; rather it indicates that the problem has been *overparameterized* with more knots than necessary. However, since control values are pre-specified in the optimization, it is useful to overparameterize the problem, allowing for more control switches than needed. Overparameterizing allows the optimizer to demonstrate the optimal number of switches (less than the parameterized number) by driving to zero superfluous control axis durations. The overparameterization also allows the user additional flexibility to arbitrarily pre-specify control values, knowing that non-optimal control values are eliminated in the final solution.

This same behavior is observed for the minimum-fuel optimization displayed in Figure 4(b). One may easily observe that most thrusting arcs are reduced exactly to zero by the optimizer for both control axes.



(a) (b) (c)

**Figure 4 Lunar Lander Problem: Minimum-Time (a) and Minimum-Fuel (b) FSCT Solutions; Model Predictive Controller Simulation (c)**

This indicates that far fewer switches were necessary to identify this local minimum, and it provides confidence that the formulation has not underparameterized the problem by providing too few control switching opportunities.

An important discovery from the Lunar Lander example is the extent by which the FSCT method results in implementable control solutions. First, it is clear that the solution requires some “interpretation.” Superfluous control switches must be discounted before implementing the control history. Actuators with minimum on-times do not support thrust durations approaching zero; however, within the tolerance of the optimization, zero or near-zero burn durations actually indicate that the respective actuation is not desirable. Clearly, an optimization must be scaled properly in time to differentiate short actuation times from non-optimal control sequences. Secondly, once a control solution is adequately interpreted, the performance of the solution in a continuous time setting can be nearly identical. Although this collocation technique does rely on a time discretization along each segment, the switching times between control values are optimized over a continuous spectrum. Therefore, the control solution represents exact switching times within the tolerance of the optimization.

## A Model Predictive Controller for Real-Time Implementation

One potential drawback of the FSCT method is that, while capable of producing optimal control histories for the finite set control problem, optimal control laws for real-time implementation are not immediately available. For the general dynamical problem, there is no guarantee that an optimal control solution will imply a real-time law, $\boldsymbol{u} = \boldsymbol{u}(t, \boldsymbol{y})$. To compensate for this limitation, a process is now considered by which FSCT solutions may be implemented in conjunction with a model predictive control design for real-time implementation of finite control. To begin, a simple model predictive controller is introduced. More complicated, and perhaps more robust, control designs are beyond the scope of this work, although developed in the theory on Model Predictive Control.[13]

*Linear Discrete-Time Model.* Model Predictive Control (MPC) offers a method for tracking an arbitrary reference trajectory while optimizing a performance index over a finite time horizon. Specifically, MPC techniques can be used for tracking (i.e. implementing) an a priori discovered FSCT solution. A basic model predictive controller is derived using a discrete-time, linear dynamic model of the form,

$$\begin{aligned} \boldsymbol{y}(t+\Delta t) &= \boldsymbol{A}(t)\boldsymbol{y}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t), && (9) \\ \boldsymbol{z}(t) &= \boldsymbol{C}\boldsymbol{y}(t). && (10) \end{aligned}$$

Here, $\boldsymbol{z}(t)$ is the measured output from the linear system. In general, then, the nonlinear continuous dynamics of Equation 1 must be transformed into the form of Equations 9-10 through appropriate definitions of $\boldsymbol{A}(t)$, $\boldsymbol{B}(t)$, and $\boldsymbol{C}$. It is beyond the scope of this document to demonstrate this transformation.

*MPC Control Law.* The MPC law exploits the linear discrete-time model to develop estimates for the observation variables, $\boldsymbol{z}(t)$, at future time intervals, given the current state values. The output *predictions* are determined for a finite horizon of future times, and current control values are chosen so that these output estimates are as close as possible to desired values of the nominal trajectory (the FSCT solution). In the traditional sense, the linear discrete formulation allows controls to be determined by solving a linear equation, holding constant the control values over the increment $\Delta t$. The finite set control nature of the hybrid system motivates a minor modification to this design, featured presently.

Let the estimate on the output at time $t + j\Delta t$, given the states at time $t$, be denoted as $\hat{\boldsymbol{z}}(t + j\Delta t|t)$ such that

$$\begin{aligned} \hat{\boldsymbol{z}}(t + j\Delta t|t) &= \boldsymbol{C}\hat{\boldsymbol{y}}(t + j\Delta t|t) \qquad (11) \\ &= \boldsymbol{C}\left[\hat{\boldsymbol{A}}(t + (j-1)\Delta t|t)\hat{\boldsymbol{y}}(t + (j-1)\Delta t|t) + \hat{\boldsymbol{B}}(t + (j-1)\Delta t|t)\boldsymbol{u}(t + (j-1)\Delta t|t)\right] \end{aligned}$$

In this notation, the symbol, $\hat{\cdot}$, indicates that the variable is estimated for a future time value. Equation 11 can be manipulated to show that any future estimate may be expressed as a function of $\boldsymbol{y}(t)$, the controls $\boldsymbol{u}(t)$ through $\boldsymbol{u}(t + (j-1)\Delta t)$, and the estimated values of the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices through time $t + (j-1)\Delta t$. Let the predicted future outputs at discrete time intervals be stored in the vector, $\boldsymbol{Z}$, where

$$\boldsymbol{Z} = \begin{bmatrix} \hat{\boldsymbol{z}}^T(t + \Delta t|t) & \cdots & \hat{\boldsymbol{z}}^T(t + j\Delta t|t) & \cdots & \hat{\boldsymbol{z}}^T(t + p\Delta t|t) \end{bmatrix}^T,$$

and $p$ is the *prediction interval* over which observation variables are stored. Using the simplifying assumption that $\boldsymbol{u}(t + j\Delta t) = \boldsymbol{u}(t)$ over the entire prediction interval, the output can be expressed as

$$\boldsymbol{Z} = \boldsymbol{G}\boldsymbol{y}(t) + \boldsymbol{K}\boldsymbol{u}(t) \qquad (12)$$

with appropriate definitions for the matrices $\boldsymbol{G}$ and $\boldsymbol{K}$. In Equation 12, a linear relationship is defined between the current states, the current control values, and estimates of the future output. Let the nominal output (corresponding to the FSCT trajectory) be expressed at the same discrete time intervals in the vector, $\boldsymbol{Z}_n$. A cost function of the form

$$\mathcal{J} = \frac{1}{2}(\boldsymbol{Z}_n - \boldsymbol{Z})^T\boldsymbol{Q}(\boldsymbol{Z}_n - \boldsymbol{Z}) + \frac{1}{2}(\boldsymbol{u}_n - \boldsymbol{u})^T\boldsymbol{R}(\boldsymbol{u}_n - \boldsymbol{u}) \qquad (13)$$

may be employed to penalize deviations in both states and controls away from the nominal. Thus, $\mathcal{J}$ can then be minimized according to user defined weight matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ to produce a tracking trajectory that also minimizes control over the prediction interval. If $\mathcal{J}$ is minimized at each interval, then $\boldsymbol{u}$ may still take on different values at each interval, even though the prediction equation assumes otherwise. A model predictive control law for the hybrid system is defined according to

$$\boldsymbol{u} = \arg \min_{\boldsymbol{u} \in \mathbb{U}^{n_u}} \mathcal{J}.$$

Simply stated, the implemented control at the current time is the feasible control combination that minimizes the cost function. Thus, if each control variable, $u_i$, may assume $m_i$ possible values, then the control must consider each of the $\bar{m}$ possible control combinations, implementing the minimizing choice. Notice that $\bar{m} = \prod_{i=1}^{n_u} m_i$, and the control law is most effective for $n_u$ and $m_i$ (and therefore $\bar{m}$) reasonably small to reduce the number of computations per interval.

*A Comparison of MPC Performance.* The hybrid system model predictive controller is easily demonstrated in conjunction with the FSCT method using the minimum-fuel solution of the lunar lander problem. With $\bar{m} = (3)(3) = 9$ possible control combinations, it is reasonable to assume that there exists a time interval, $\Delta t$, such that $\bar{m}$ evaluations of $\mathcal{J}$ can be compared to determine the minimizing control per interval. For this simulation, $\Delta t$ is chosen such that there are 500 intervals between $t_0$ and $t_f$ (less that four intervals per second for the minimum-fuel trajectory). In addition, a prediction horizon is selected where $p = 10$, indicating that the controller calculates the estimates for the next 10 intervals of the output at each time step. Using Equation 13, the objective is to mimic the FSCT solution as close as possible via a real-time implementation. Since the a priori discovered optimal solution includes state and control values, it is logical to

**Table 1. Relevant Quantities for a Micro-Satellite**

| Assumed Quantities | | | | |
|---|---|---|---|---|
| Dimensions | | $l_1$ | 1.00 | m |
| | | $l_2$ | 1.25 | m |
| | | $l_3$ | 1.50 | m |
| | | $r$ | 0.25 | m |
| Masses | Dry Mass | $m_d$ | 15.00 | kg |
| | Propellant Mass (at $t_0$) | $m_p$ | 5.00 | kg |
| Cold Gas Propulsion | Specific Gas Constant ($N_2$) | $R$ | 296.80 | $N \cdot m/(kg \cdot K)$ |
| | Specific Heat ($N_2$) | $\gamma$ | 1.4 | |
| | Storage Temperature | $T$ | 298.15 | K |
| | Maximum Thrust | $F_t$ | 2.00 | N |
| | Nozzle Throat Radius | $r_t$ | 2.50 | mm |

| Derived Quantities | | | | |
|---|---|---|---|---|
| Volumes | Cuboid Volume | $V_c$ | $l_1 l_2 l_3$ | |
| | Propellant Volume | $V_p$ | $\frac{4}{3}\pi r^3$ | |
| | Dry Volume | $V_d$ | $V_c - V_p$ | |
| Masses | Total Mass | $m_t$ | $m_d + m_p$ | |
| | Cuboid Mass | $m_c$ | $m_d \frac{V_c}{V_d}$ | |
| | Extra Sphere Mass | $m_s$ | $m_t - m_c$ | |
| Cold Gas Propulsion | Characteristic Velocity | $c^*$ | 434.439 | m/s |
| | Exhaust Velocity | $c$ | 787.042 | m/s |
| | Throat Density × Velocity | $\rho_t v_t$ | 129.42 | $kg/(m^2 s)$ |

use all of this information in the controller. The weight matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$, are proportioned, however, to emphasize position state tracking over velocity or control tracking.

The results of a simulation implementing the real-time controller are depicted in Figure 4(c), displaying positions, velocities, and control values of the lunar lander. For positions and velocities, both FSCT solution and MPC simulation states are plotted to demonstrate minimal deviations between the two. It is especially interesting to compare the control histories of Figure 4(b) and Figure 4(c): they are nearly identical. The primary observable difference between the MPC simulation and the FSCT solution is that the simulation has removed the instantaneous control switches that resulted from overparameterization in the FSCT formulation. Thus, with the FSCT solution in hand, it is possible to derive a real-time control law that very closely recreates the optimal trajectory.

The consistency between the FSCT minimum-fuel solution and the hybrid system MPC simulation suggest the effectiveness of using the two methodologies in tandem. It is observed that the FSCT method offers control histories instead of implementable control laws. On the other hand, an MPC-derived controller may only be as good as the nominal trajectory selected for tracking. As a pair, however, it is possible to derive optimal trajectories and control histories, *and* implement them in a real-time context, where perturbations, modeling errors, and other unknowns are likely to arise. This example is intended to further illustrate the utility of the FSCT method when a control law, rather than a control history, is desired.

## SMALL SPACECRAFT ATTITUDE CONTROL

In a final example, the FSCT method is applied to determine finite set control schedules for tracking an arbitary spacecraft orientation. The FSCT method is well-suited for the problem when only on-off actuation is available. This example is specifically motivated by spacecraft limited to commercial off-the-shelf actuator technologies that are inexpensive and readily available. The available literature[14,15,16] indicates a range of new thruster technologies for small spacecraft that are currently under development. Although these may offer wide ranges of thrust magnitudes and performance efficiencies, it is interesting to explore how the capability of traditional technologies can be stretched to maximize performance. The attitude control problem offers an exciting dynamic environment along with conceivable control limitations which make the FSCT method quite relevant.

Consider a low-cost micro-satellite employing a basic nitrogen cold gas propulsion system[17,18] for attitude control. Two scenarios are now investigated for this small spacecraft attitude control problem. In both of the scenarios, the spacecraft is equipped with six thruster-pairs situated on a cuboid spacecraft body to provide purely rotational control in each of the body's principal directions, positive and negative. The propulsion system is supplied by a single $N_2$ propellant tank, centrally located in the spacecraft for simplicity. Temperature and pressure are regulated at each thruster nozzle to allow for constant thrust of 2 N. Pertinent statistics for the spacecraft and propulsion system are listed in Table 1.

The first scenario demonstrates the simplest control system to conceive. Each thruster pair is controlled by an on/off valve, and thrust magnitudes are limited to two values (2 N when on, 0 N when off). The second scenario explores a variable-thrust cold gas propulsion system in which the effective throat size of each thruster nozzle varies to alter propellant mass flow. However, the new problem can still be modeled in a finite set control formulation so that the FSCT method can be used. In transitioning between the two scenario formulations, it is suggested that *many* variable control problems are actually, at some level, finite control problems with an extended dynamic description.

For the dynamical relations that follow, it is necessary to identify the principal moments of inertia for the spacecraft. Assume a constant mass density within the propellant tank, and further assume a constant mass density in the remaining dry space of volume $V_d$. Using the quantities derived in Table 1, the spacecraft moment in the first principal direction is

$$J_1 = \frac{1}{12} m_c \left(l_2^2 + l_3^2\right) + \frac{2}{5} m_s r^2,$$

with similar definitions for the second and third.

In each scenario, the dynamics are described using quaternion elements. Recall that the quaternion, $\boldsymbol{q} = \left[q_0 \ \boldsymbol{q}_v^T\right]^T$, is a 4-dimensional quantity, consisting of one scalar element and one 3-dimensional vector, that describes the rotation between two coordinate frames. The $3 \times 3$ coordinate transformation matrix used to rotate between two frames may be written as a function of the quaternion, as

$$\boldsymbol{C}(\boldsymbol{q}) = (q_0^2 - \boldsymbol{q}_v^T \boldsymbol{q}_v)\boldsymbol{I} + 2\boldsymbol{q}_v \boldsymbol{q}_v^T - 2q_0[\boldsymbol{q}_v \times]$$

where $[\boldsymbol{q}_v \times]$ is the skew-symmetric cross product matrix that operates on a vector in the same way as a cross product. Thus, the quaternion and the coordinate transformation matrix are directly linked to describe the relationship between two reference frames. With no singularities, the quaternion is an effective means of representing attitude dynamics. For clarity, superscripts are used to describe the reference frames that a quaternion relates. For example, for the reference frames $\{\hat{\boldsymbol{a}}\}$ and $\{\hat{\boldsymbol{b}}\}$, the relating quaternion is ${}^b\boldsymbol{q}^a$, and

$$\{\hat{\boldsymbol{b}}\} = \boldsymbol{C}({}^b\boldsymbol{q}^a)\{\hat{\boldsymbol{a}}\}.$$

### Low-Cost Cold Gas Thrusters: Fixed Thrust Attitude Control

Consider a micro-satellite with on-off actuation for its six attitude control thruster pairs. Each thruster delivers either 0 N or 2 N of thrust, depending on the state of each on/off valve. Using the existing propulsion system, the objective in this scenario is to *track an arbitrary reference trajectory as well as possible, while minimizing fuel expenditure*.

Let the reference trajectory be described by the reference quaternion, ${}^r\boldsymbol{q}^i$, relating the inertial $\{\hat{\boldsymbol{i}}\}$ frame to the reference $\{\hat{\boldsymbol{r}}\}$ frame, while ${}^r\boldsymbol{\omega}^i$ indicates the angular velocity vector between the same frames, written in the reference frame. With the initial quaternion, ${}^r\boldsymbol{q}^i{}_0 = [1 \ 0 \ 0 \ 0]^T$, and angular velocity explicitly defined with respect to time as

$${}^r\boldsymbol{\omega}^i(t) = \begin{bmatrix} 0.3\cos t\left(1 - e^{-0.01t^2}\right) + (0.08\pi + 0.006\sin t)\, te^{-0.01t^2} \\ 0.3\sin t\left(1 - e^{-0.01t^2}\right) + (0.08\pi + 0.006\cos t)\, te^{-0.01t^2} \\ 1 \end{bmatrix} \text{rad/s}, \tag{14}$$

the reference trajectory is completely specified, indicating the ideal attitude for the spacecraft at all times $t$. The reference angular velocity description is purely arbitrary, but it is observed that this reference trajectory offers interesting movement in position and velocity states over the fixed time $t \in [0 \ 20]$ seconds and is an excellent test case for an FSCT optimization.

Define the state vector for this dynamical system according to

$$\boldsymbol{y} = \left[ \ ({}^b\boldsymbol{q}^i)^T \quad ({}^b\boldsymbol{\omega}^i)^T \quad m_p \quad ({}^r\boldsymbol{q}^i)^T \quad p \ \right]^T,$$

where ${}^b\boldsymbol{q}^i$ indicates the quaternion relating the inertial frame and the spacecraft body frame, ${}^b\boldsymbol{\omega}^i$ is the corresponding angular velocity vector, and $m_p$ is the depleting propellant mass when thrusters are activated.

These first three components of $\boldsymbol{y}$ completely define the relevant states of the spacecraft. The remaining elements included in the state vector are strictly for computational convenience as these are useful for cost evaluation and control determination. The formulation allows the reference quaternion, ${}^r\boldsymbol{q}^i$, to be determined at each relevant instant in time by the FSCT method. The scalar state, $p$, measures an integral cost for deviations between reference and actual trajectories.

In the next step, the control vector is defined. Ultimately, $\boldsymbol{u}$ must indicate the position of the on/off valve for each of the twelve thrusters. Since thrusters, at a minimum, are assumed to act in pairs, it is logical to allow each control variable to indicate the valve position for at least two thrusters. However, it is also observed that each thruster pair has a corresponding thruster pair which acts in an opposing fashion such that their effects are cancelled when both pairs are on. Thus, consider the contol vector, $\boldsymbol{u} \in \mathbb{U}^3$ where $\mathbb{U} = \{-1, 0, 1\}$. Thus, $n_u = 3$, and each control variable is limited to three values, indicating for each principal axis, whether the positive-thrusting pair, the negative-thrusting pair, or neither is in the on position.

The state dynamics for the system are described by the following relations,

$$\dot{\boldsymbol{y}} = \begin{bmatrix} {}^b\dot{\boldsymbol{q}}^i \\ {}^b\dot{\boldsymbol{\omega}}^i \\ \dot{m}_p \\ {}^r\dot{\boldsymbol{q}}^i \\ \dot{p} \end{bmatrix} = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{u}) = \begin{bmatrix} \frac{1}{2}\boldsymbol{E}({}^b\boldsymbol{q}^i){}^b\boldsymbol{\omega}^i \\ -\boldsymbol{J}^{-1}\,{}^b\boldsymbol{\omega}^i \times \boldsymbol{J}{}^b\boldsymbol{\omega}^i + F_t\boldsymbol{J}^{-1}\boldsymbol{L}\boldsymbol{u} \\ -2\frac{F_t}{c}\sum_{i=1}^{3}|u_i| \\ \frac{1}{2}\boldsymbol{E}({}^r\boldsymbol{q}^i){}^r\boldsymbol{\omega}^i(t) \\ \left({}^b\boldsymbol{q}^i\right)^T\boldsymbol{H}^T\left({}^r\boldsymbol{q}^i\right)\boldsymbol{H}\left({}^r\boldsymbol{q}^i\right)\left({}^b\boldsymbol{q}^i\right) \end{bmatrix},$$

where $\boldsymbol{J} = \text{diag}(J_1, J_2, J_3)$ is the inertia tensor, $\boldsymbol{L} = \text{diag}(l_1, l_2, l_3)$ contains the spacecraft dimensions, and

$$\boldsymbol{E}(\boldsymbol{q}) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}, \qquad \boldsymbol{H}(\boldsymbol{q}) = \begin{bmatrix} q_1 & -q_0 & -q_3 & q_2 \\ q_2 & q_3 & -q_0 & -q_1 \\ q_3 & -q_2 & q_1 & -q_0 \end{bmatrix} = -\boldsymbol{E}^T(\boldsymbol{q}).$$

First, note that the quaternion dynamics, $\dot{\boldsymbol{q}}$, are a function of both $\boldsymbol{q}$ and $\boldsymbol{\omega}$. When relating $\{\hat{\boldsymbol{i}}\}$ to $\{\hat{\boldsymbol{b}}\}$, both vectors are contained within $\boldsymbol{y}$. When relating $\{\hat{\boldsymbol{i}}\}$ to $\{\hat{\boldsymbol{r}}\}$, however, the angular velocities are evaluated using Equation 14. Next, observe that the dynamics, ${}^b\dot{\boldsymbol{\omega}}^i$, are dramatically simplified by expressing angular velocities in the principal body frame, where the inertia tensor is an easily invertible diagonal matrix. In addition, one may see that the mass flow dynamic, $\dot{m}_p$, is nonzero only when one or more thruster pairs is on.

The cost dynamics, $\dot{p}$, are used for evaluating an integral cost. Here, it is desired to minimize deviations between the actual and reference coordinate frames. Consider the following relations:

$$\begin{aligned} \boldsymbol{C}({}^r\boldsymbol{q}^b) &= \boldsymbol{C}({}^r\boldsymbol{q}^i)\boldsymbol{C}^T({}^b\boldsymbol{q}^i), \\ {}^r\boldsymbol{q}_v{}^b &= \boldsymbol{H}\left({}^r\boldsymbol{q}^i\right)\left({}^b\boldsymbol{q}^i\right). \end{aligned}$$

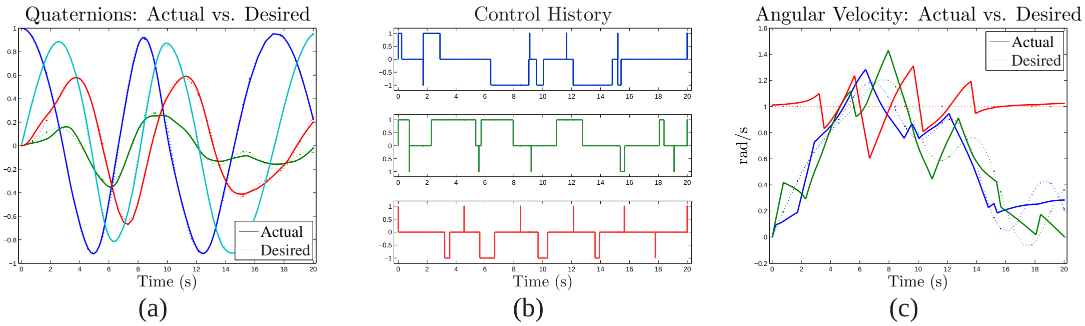
Then if $\{\hat{\boldsymbol{b}}\}$ and $\{\hat{\boldsymbol{r}}\}$ are identical, ${}^r\boldsymbol{q}_v{}^b = \boldsymbol{0}$. A cost function that penalizes $\left({}^r\boldsymbol{q}_v{}^b\right)^T\left({}^r\boldsymbol{q}_v{}^b\right) > 0$ ensures minimal deviations between body and reference coordinate frames. This is equivalent to setting $p_0 = 0$ and minimizing $p_f$ since

$$p_f - p_0 = \int_{t_0}^{t_f} \dot{p}\, dt = \int_{t_0}^{t_f} \left({}^r\boldsymbol{q}_v{}^b\right)^T\left({}^r\boldsymbol{q}_v{}^b\right)\, dt.$$

The complete cost function weighs penalties on trajectory tracking deviations with the amount of propellant mass expelled in tracking the reference. Minimizing the total cost function,

$$\mathcal{J} = \beta_1 p_f - \beta_2 m_{p_f} \tag{15}$$

is equivalent to minimizing tracking deviations and maximizing the final propellant mass when $\beta_1 > 0$ and $\beta_2 > 0$.



(a) (b) (c)

**Figure 5. Fixed Thrust Attitude Control**

The problem is completely defined by identifying the remaining initial states for the optimization. Let the spacecraft begin along the reference trajectory. In this case, ${}^{b}\boldsymbol{q}_0^i = {}^{r}\boldsymbol{q}_0^i = [1\ 0\ 0\ 0]^T$ and ${}^{b}\boldsymbol{\omega}_0^i = {}^{r}\boldsymbol{\omega}^i(t_0)$. In addition, assume the initial propellant mass is $m_{p_0} = 5$ kg. These assumptions imply there is sufficient propellant available to achieve reasonable trajectory tracking for the interval from $t_0 = 0$ to $t_f = 20$ seconds.

*FSCT Solution.* The fixed time optimal control problem detailed above is solved using the FSCT method to yield a feasible and locally optimal trajectory and control switching schedule. For this sample solution, the selected transcription parameters are $n_n = 5$ nodes per segment and $n_k = 20$ knots, allowing 20 control switches in each $u_i$ over the time interval from $t_0$ to $t_f$. The pre-specified control values are selected based on the following law:

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

This control law alternates between positive-, zero- and negative-torque for each control variable. Clearly, the control sequence selection resembles that of the Lunar Lander problem, as this seems to allow substantial flexibility to solve the underlying NLP problem.

The FSCT solution is depicted in Figure 5 when the cost function is set with equal penalty weights, $\beta_1 = \beta_2$. In Figure 5(a), the trajectory position (quaternion) histories for ${}^{b}\boldsymbol{q}^i$ and ${}^{r}\boldsymbol{q}^i$ are shown as the ‘actual’ and ‘desired’ trajectories, respectively. This illustration gives a visual sense of how well the trajectory can be tracked given finite value control limitations. In Figure 5(b), the resulting control history (switching schedule) is depicted. For each control variable, note that the durations of arcs associated with both $u_i = 1$ and $u_i = -1$ are reduced to zero. This indicates that the transcription formulation is not underparameterized.

Finally, Figure 5(c) depicts the actual and desired angular velocities, ${}^{b}\boldsymbol{\omega}^i$ and ${}^{r}\boldsymbol{\omega}^i$. It is not unexpected that significantly more deviation is observable in this plot. Control restrictions clearly reduce the way in which velocity variables can change with time. More importantly, deviations in angular velocities are not penalized in the cost function, so the FSCT method does not attempt directly to drive velocities to match.

## Low-Cost Cold Gas Thrusters: Variable Thrust Attitude Control

The performance of the system above is clearly limited by on-off control actuation. Of course, fixed thrust control is an obvious choice when the intent is to apply the FSCT method to a real example. Indeed, the simplest, and perhaps least expensive, propulsion systems can benefit from the methodology for determining control strategies for reference tracking. A hybrid system model predictive controller, used in conjunction with FSCT solutions may be as successful in this case as it was for the Lunar Lander presented earlier. Here, however, another scenario is presented to expand the class of applications available to the FSCT methodology.

The most straightforward way of improving upon the solutions of the first attitude control scenario is to expand the solution space to include variable magnitude control inputs. This improves performance through better tracking, less fuel expenditure, or both. Thus, this scenario explores the possibility of a variable amplitude controller with a modified cold gas propulsion system. The purpose of the development that follows is to demonstrate that the variable control problem can *still* be interpreted, on a higher level, as a

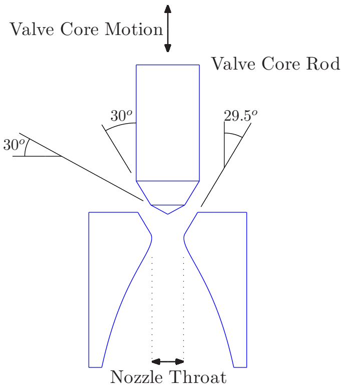


**Figure 6. A Variable Amplitude Thruster Nozzle**

finite set control problem. Extrapolating further, many, if not most, dynamical systems with variable control inputs can be extended to reveal discrete components. Consider, for example, a control system whose varying inputs are determined by a digital computer. At the highest level, everything is reduced to a binary description, '0's and '1's, not unlike the discrete control inputs shown in the examples so far.

A previously developed variable amplitude cold gas propulsion system[19] serves as the inspiration for the following development. Here, the nitrogen propellant system is modified to allow variation in the effective dimension of the nozzle throat and, subsequently, the propellant mass flow. Consider the illustration of Figure 6. A valve core rod lies near the throat of the thruster nozzle, and has controlled motion up and down. Let the variable, $d_i$, indicate the position of the valve core rod for the $i^{\text{th}}$ thruster. In addition, define $r_t$ as the radius of the nozzle throat. The effective throat area is a function of the rod position, expressed as

$$A_t(d_i) = \pi r_t^2 - \pi \left( r_t - \frac{1}{2} d_i \right)^2, \tag{16}$$

where $0 \leq d_i \leq 2r_t$. Note that if the rod position is such that $d_i > 2r_t$, no effect is expected on thruster performance, and $(A_t)_{\text{max}} = \pi r_t^2$. Because the throat area directly affects the mass flow through the nozzle (assuming constant propellant density and velocity), it has a direct effect on the magnitude of thrust. Assuming, as before, that the maximum thrust available is $(F_t)_{\text{max}} = 2$ N, then

$$\rho_t v_t = \frac{(F_t)_{\text{max}}}{(A_t)_{\text{max}}\, c},$$

which can be evaluated using the constants in Table 1. Now, the amplitude of control for each thruster is a function of one discrete variable, indicating the position of the on/off valve, and one continuous variable, indicating the valve core rod position. To describe the dynamical system, it is necessary to understand how the rod position, $d_i$, is controlled. Surely, there are many ways of doing this, all affecting the nature of the dynamics. Assume then, for the sake of this argument, that each rod is driven by a constant-acceleration motor. Thus, the rod position and its velocity, $v_i$, are continuous variables, while its acceleration, $a_i$, may take only a discrete number of values.

If the valve core rod positions and velocities are included as state variables, a hybrid system ensues consistent with the formulation in Equation 1, with only continuous states and discrete controls. While this is not the only formulation for the variable amplitude control problem, this formulation demonstrates that it is possible to extend a variable amplitude control device into a combination of continuous states and discrete decision variables. In this case, states are defined for the physical elements that allow for thrust amplitude variations.

The state vector for this scenario, then, includes the same quantities as the previous scenario, now adding core rod positions and velocities to the set. Recall that, at a minimum, the twelve thrusters of the microsatellite are combined into six thruster pairs. In the first scenario, two pairs providing torque along the same axis of rotation were considered together. In this scenario, the dynamic relations dictate that only thruster pairs can be considered to act in harmony. Let the vectors, $\boldsymbol{d}$ and $\boldsymbol{v}$, which contain the individual rod dynamics, have six components each. Thus, a thruster pair shares the same rod positions and velocities to ensure that translational accelerations cancel at all times. The state vector for the dynamical system takes the form,

$$\boldsymbol{y} = \left[ \begin{array}{ccccccc} ({}^b\boldsymbol{q}^i)^T & ({}^b\boldsymbol{\omega}^i)^T & m_p^T & \boldsymbol{d}^T & \boldsymbol{v}^T & ({}^r\boldsymbol{q}^i)^T & p^T \end{array} \right]^T .$$

The control vector is now

$$\boldsymbol{u} = \left[ \begin{array}{cc} \boldsymbol{w}^T & \boldsymbol{a}^T \end{array} \right]^T ,$$

where $\boldsymbol{w}$ and $\boldsymbol{a}$ are each vectors composed of 6 elements (corresponding to the number of thruster pairs), $w_i \in \{0, 1\}$ indicates whether the $i^{\text{th}}$ thruster pair is on or off, and $a_i \in \{-1, 0, 1\}$ indicates the acceleration of the valve core rods of the $i^{\text{th}}$ thruster pair, which can be negative, zero, or positive. The dynamics of the system are described by

$$\dot{\boldsymbol{y}} = \begin{bmatrix} {}^b\dot{\boldsymbol{q}}^i \\ {}^b\dot{\boldsymbol{\omega}}^i \\ \dot{m}_p \\ \dot{\boldsymbol{d}} \\ \dot{\boldsymbol{v}} \\ {}^r\dot{\boldsymbol{q}}^i \\ \dot{p} \end{bmatrix} = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{u}) = \begin{bmatrix} \frac{1}{2}\boldsymbol{E}({}^b\boldsymbol{q}^i){}^b\boldsymbol{\omega}^i \\ -\boldsymbol{J}^{-1}\, {}^b\boldsymbol{\omega}^i \times \boldsymbol{J}{}^b\boldsymbol{\omega}^i + \rho_t v_t c \boldsymbol{J}^{-1}\boldsymbol{L}\boldsymbol{A}(\boldsymbol{d})\boldsymbol{w} \\ -2\rho_t v_t \tilde{\boldsymbol{A}}^T(\boldsymbol{d})\boldsymbol{w} \\ \alpha_2 \boldsymbol{v} \\ \alpha_3 \boldsymbol{a} \\ \frac{1}{2}\boldsymbol{E}({}^r\boldsymbol{q}^i){}^r\boldsymbol{\omega}^i(t) \\ \left({}^b\boldsymbol{q}^i\right)^T \boldsymbol{H}^T \left({}^r\boldsymbol{q}^i\right) \boldsymbol{H} \left({}^r\boldsymbol{q}^i\right) \left({}^b\boldsymbol{q}^i\right) \end{bmatrix} ,$$

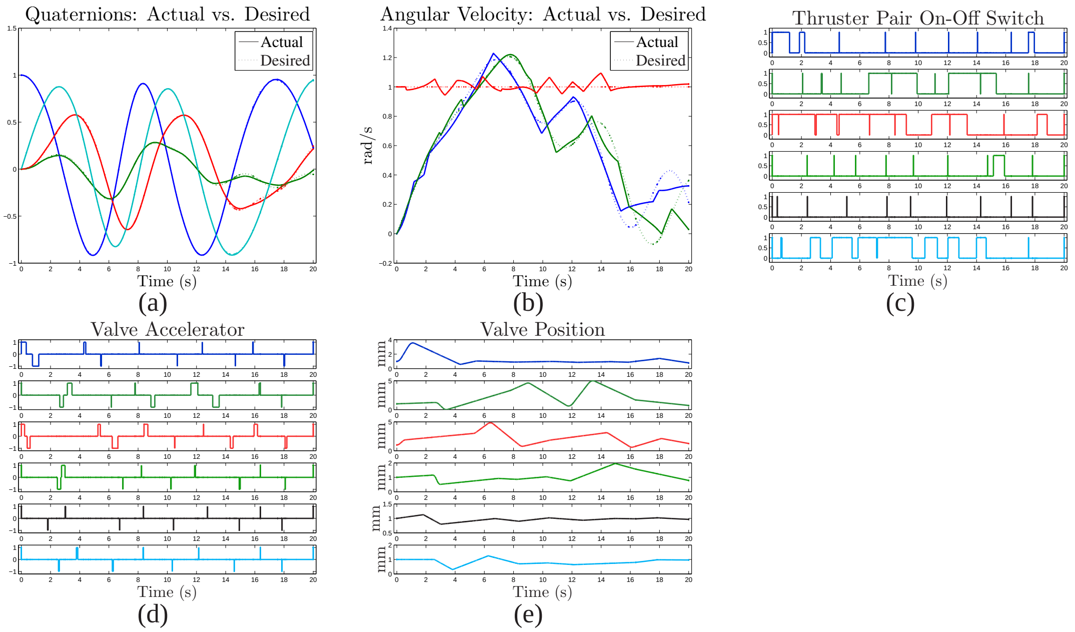
where the previously defined quantities $\boldsymbol{J}$, $\boldsymbol{L}$, $\boldsymbol{E}$, and $\boldsymbol{H}$ are unchanged, and

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{d}) &= \begin{bmatrix} A_t(d_1) & -A_t(d_2) & 0 & 0 & 0 & 0 \\ 0 & 0 & A_t(d_3) & -A_t(d_4) & 0 & 0 \\ 0 & 0 & 0 & 0 & A_t(d_5) & -A_t(d_6) \end{bmatrix} , \\ \tilde{\boldsymbol{A}}(\boldsymbol{d}) &= \left[ \begin{array}{cccccc} A_t(d_1) & A_t(d_2) & A_t(d_3) & A_t(d_4) & A_t(d_5) & A_t(d_6) \end{array} \right]^T , \\ A_t(d_i) &= \frac{1}{\alpha_1^2}\left[\pi r_t^2 - \pi\left(r_t - \frac{1}{2}d_i\right)^2\right] . \end{aligned} \tag{17}$$

Notice immediately that Equation 17 differs from Equation 16 by the scaling factor, $\alpha_1$. Additional scaling factors, $\alpha_2$ and $\alpha_3$, are present in the valve core rod dynamics, as well, so that all state variables remain $\mathcal{O}(10^0)$ to improve the convergence of the underlying NLP problem. In this case, $\alpha_1 = 10^3$ so that $r_t$ and $d_i$ are presented in mm. Likewise, $\alpha_2 = 10^1$ so $\boldsymbol{v}$ is in $10^{-4}$ m/s, and $\alpha_3 = 10^0$ so that $a_i = 1$ indicates that the $i^{\text{th}}$ rod is accelerating by $10^{-4}$ m/s$^2$.

Clearly, $\boldsymbol{A}(\boldsymbol{d})$ and $\tilde{\boldsymbol{A}}(\boldsymbol{d})$ represent the effective throat cross-sectional area for each thruster pair, listed in matrix form and vector form, respectively. These facilitate the new definitions for ${}^b\dot{\boldsymbol{\omega}}^i$ and $\dot{m}_p$. Thus, the effective control torque, evaluated by $\rho_t v_t c \boldsymbol{J}^{-1}\boldsymbol{L}\boldsymbol{A}(\boldsymbol{d})\boldsymbol{w}$ and measured in rad/s$^2$, as well as the total mass flow defined by $\dot{m}_p$, are determined by the current throat area and the state of the on/off valve.

*FSCT Solution.* In as many ways as possible, the optimization of the variable-thrust attitude control scenario is set up identically to the fixed-thrust scenario. The cost function is that of Equation 15. Again, $\beta_1 = \beta_2$ to allow for a direct comparison between results. In this transcription formulation, $n_n = 5$ and $n_k = 20$ again, but with additional control variables ($n_u = 12$, instead of 3), the total number of segments, and thereby nodes, is significantly increased.



(a) (b) (c) (d) (e)

**Figure 7. Variable Thrust Attitude Control**

The pre-specified controls for the formulation follow a standard structure.

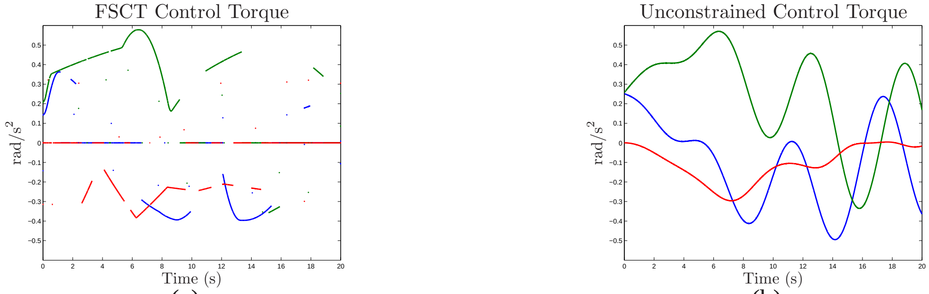
$$
\begin{aligned}
w^*_{i,k} &= \frac{1}{2} + \frac{1}{2}(-1)^{k-1}, \\
a^*_{i,k} &= \cos\left(\frac{\pi}{2}(k-1)\right)
\end{aligned}
$$

Notice that each $w_i$ alternates between the values 1 and 0, while $a_i$ alternates between $1$, $0$, and $-1$.

The results of the FSCT optimization are presented in full in Figures 7 and 8. Immediately, Figure 7(a-b) can be compared to Figure 5 to show how quaternions and angular velocities match the reference trajectories in each scenario. As expected, the variable thrust formulation offers more flexibility, and consequently better tracking, for the attitude control problem.

Figure 7(c-d) record the control histories that produce this trajectory. For each thruster pair, the controls indicate whether the thruster switch is on or off, and whether the motors driving the valve core rod are accelerating the rod. For completeness, Figure 7(e) shows the position history of the valve core rods for each thruster pair. Notice that the positions remain within the bounds $0 \leq d_i \leq 5$ mm, where the rod position has an effect on the resulting mass flow through the nozzle.

Figure 8 examines the effective control torque history for the system. When the effects of all of the finite value control variables are considered along with new dynamic states ($\boldsymbol{d}$ and $\boldsymbol{v}$), one can extract the actual control torque, measured in rad/s$^2$, that is applied to the spacecraft at any time. Figure 8 (a) illustrates this. Note that the zero-duration ‘dots’ contained in the figure are artifacts of zero-duration segments that naturally result in an FSCT solution. For the sake of this discussion, they can effectively be ignored. As a comparison to this solution, Figure 8 (b) illustrates the control torque for an *unconstrained control* system which tracks the reference trajectory perfectly. The unconstrained control torque is derived using a continuous Lyapunov-based control law which guarantees perfect tracking of quaternions and angular velocities (since initial states are along the reference). Some distinct similarities are easy to observe by looking at the plots together. Certainly, control torque magnitudes are similar, but there are also points at which the derived control torque from the finite set formulation very closely mimics the behavior of the purely continuous control.



(a) (b)

**Figure 8 FSCT Variable Thrust Attitude Control Torque vs. Unconstrained Attitude Control Torque**

This is viewed as a significant result which demonstrates how a detailed hybrid system formulation can approach a continuous formulation. While it is a step backwards to use a finite control formulation if control inputs are truly continuous, perhaps it is reasonable to argue that *many* systems, if not most, are truly hybrid systems, modeled as continuous. Often, it is easier to model the continuous system, as numerous methodologies exist for treating such systems. However, if a system has discrete components, it is ideal to treat them as such. Thus, the FSCT method offers an avenue for modeling maybe inevitably present discrete components, at whatever level they appear in the dynamics.

## CONCLUSIONS

The Finite Set Control Transcription method is demonstrated for the determination of optimal solutions to hybrid control problems. The intent of the present investigation is to explore the range of applications of the FSCT method. Although many of the applications in this investigation are particularly relevant to aerospace engineering, the applicability of the method extends to all engineering disciplines.

Three example applications are used to provide context for the method. First, the FSCT method is used on a simple two-state system with two individually-stable dynamical modes. In a number of different sources, the method of multiple Lyapunov functions is utilized to treat the hybrid system with one decision variable. Here, results from the FSCT method are analyzed to demonstrate how optimal control laws may be extracted whose performance exceeds those derived using a Lyapunov argument.

Next a simple lunar lander problem is addressed by the FSCT method. A primary feature of the FSCT method is its ability to manage multiple independent decision inputs simultaneously. In this two-dimensional example, two control variables are included in the optimal control formulation to illustrate how the FSCT method can be applied when multiple independent control variables are considered. Solutions derived via the FSCT method are further utilized in conjunction with a hybrid system model predictive control scheme. For the hybrid system with a reasonable number of possible decision inputs at any given time, the MPC formulation offers real-time decision-making for the hybrid system. When the two methods are used in tandem, optimized control schedules can be realized in the context of potential perturbations or other unknowns.

Finally, the FSCT method tackles an attitude control problem presented in two different formulations. In the first, a small spacecraft is assumed to be limited to finite thrust magnitudes for a cold gas propulsion attitude control system. The second scenario explores a variable-thrust propulsion system, still modeled as a hybrid system. This investigation argues that even a system traditionally modeled with continuous control inputs may be more accurately described as a system ultimately relying on discrete decision variables. Continuous control variables may often be extended into a set of continuous state variables and discrete inputs. The scenario considers *how* an actuator may actually vary the control magnitude it applies. This process generates additional dynamics that can be modeled within the system. It is conceivable that, at some level, many systems can be thought of as hybrid systems with completely continuous states, and completely discrete control variables.

## REFERENCES

[1] M.S. Branicky, V.S. Borkar, and S.K. Mitter, "A unified framework for hybrid control: Model and optimal control theory", *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 31–45, Jan 1998.

[2] Kagan Gokbayrak and Christos G. Cassandras, "Hybrid controllers for hierarchically decomposed systems", *Hybrid Systems: Computation and Control*, vol. 1790, pp. 117–129, Mar 2000.

[3] M.S. Branicky, "Stability of switched and hybrid systems", in *Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, FL, Dec 1994, pp. 3498–3503.

[4] M.S. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems", *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, Apr 1998.

[5] Stefan Pettersson and Bengt Lennartson, "Controller design of hybrid systems", *Hybrid and Real-Time Systems*, vol. 1201, pp. 240–254, Mar 1997.

[6] Vadim Azhmyakov, Sid Ahmed Attia, Dmitry Gromov, and Jörg Raisch, "Necessary optimality conditions for a class of hybrid optimal control problems", *Hybrid Systems: Computation and Control*, vol. 4416, pp. 637–640, Apr 2007.

[7] J.A. Ball, J. Chudoung, and M.V. Day, "Robust optimal switching control for nonlinear systems", *SIAM Journal on Control and Optimization*, vol. 41, no. 3, pp. 900–931, 2003.

[8] Stuart A. Stanton and Belinda G. Marchand, "An enhanced collocation method for dynamical systems subject to finite set control", *Journal of Guidance, Control, and Dynamics*, 2008, (Submitted for Review).

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

[10] Arthur E. Bryson and Yu-Chi Ho, *Applied Optimal Control: Optimization, Estimation, and Control*, Taylor & Francis, New York, 1975.

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

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

[13] Huang Sunan, Tan Kok Kiong, and Lee Tong Heng, *Applied Predictive Control*, Springer-Verlag, London, 2002.

[14] D.A. Gonzales and R.P. Baker, "Microchip laser propulsion for small satellites", in *37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Salt Lake City, UT, Jul 8-11, 2001, AIAA Paper 2001-3789.

[15] J. Mueller, "Thruster options for microspacecraft: a review and evaluation of existing hardware and emerging technologies", in *33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Seattle, WA, Jul 6-9, 1997, AIAA Paper 97-3058.

[16] C. Phipps, J. Luke, and W. Helgeson, "Laser space propulsion overview", in *Proceedings of SPIE - The International Society for Optical Engineering*, 2002, vol. 6606, p. 660602.

[17] S. Adler, A. Warshavsky, and A. Peretz, "Low-cost cold-gas reaction control system for sloshsat flevo small satellite", *Journal of Spacecraft and Rockets*, vol. 42, no. 2, pp. 345–351, Mar-Apr 2005.

[18] R.W. Humble, G.N. Henry, and W.J. Larson, Eds., *Space Propulsion Analysis and Design*, McGraw-Hill, New York, 1995.

[19] W.C. Stone, "Fast variable-amplitude gold gas thruster", *Journal of Spacecraft and Rockets*, vol. 32, no. 2, pp. 335–343, Mar-Apr 1995.