JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 35, No. 5, September–October 2012

Finite-Burn Linear Targeting Algorithm for Autonomous Path Planning and Guidance

S. K. Scarritt* and B. G. Marchand†
University of Texas at Austin, Austin, Texas 78712
and
A. J. Brown,‡ W. H. Tracy,§ and M. W. Weeks¶
NASA Johnson Space Center, Houston, Texas 77058

DOI: 10.2514/1.54249

In path planning and guidance applications, linear targeting through differential corrections is a classical approach for identifying feasible solutions that meet specified mission and trajectory constraints. However, to date, these methods relied on the assumption that the associated correction maneuvers were impulsive in nature. This impulsive assumption is generally reasonable when the duration of the engine burn is small. However, this approximation breaks down when lower thrust engines are employed as the duration of the burn becomes more significant. In these cases, an impulsive linear targeting algorithm is inadequate. Often times, low-thrust problems of this type are solved from the perspective of optimal trajectory design and depend on numerical methods like nonlinear programming. These methods, however, are generally considered prohibitive for autonomous flight applications, where computational resources are limited and optimality is not always as important as feasibility. The present study focuses on the theoretical development and numerical validation of a linear targeting algorithm capable of accommodating finite burn maneuvers. Examples are presented to contrast the performance of this new targeting process against more classical impulsive targeting methods. The examples presented focus largely on precision entry applications, but the finite burn targeting process itself is applicable across a broad set of scenarios and fields.

I. Introduction

IN THE context of this investigation, autonomy refers to the ability to 1) automatically identify a suitable startup arc [1–3] and 2) use that solution to successfully target the specified entry constraints within the fuel budget available at the time [4]. The first step, the identification of the startup arc, can be accomplished in one of two ways. The simplest and most common approach is to generate a database of optimal solutions over a time interval of interest and use those as nominal departure scenarios at the desired time [1,2]. The targeting process then reconverges the solution as needed to account for discrepancies in the timing and state. More recent methods [3] consider the use of infeasible solutions (e.g., with state and time discontinuities) based on a series of two-body approximations. Both methods are suitable for the generation of an initial guess, in this case. However, from a historical perspective, the database method has been successfully employed since the Apollo era, though more commonly from a ground-operations perspective. In an onboard determination scenario, the database method allows for reduced computation time when the database includes sample optimal solutions at an adequate rate. Problems that are time-sensitive require an increased number of samples, where the necessary sample frequency depends on the dynamics involved. The examples presented here employ the database approach to extract an initial guess for the subsequent targeting process. The initial guess supplied to the targeting process does not meet the specified path constraints, and sometimes, the solution may not meet the cost constraint (e.g., fuel available). The solutions supplied also use impulsive maneuvers and, therefore, assume the availability of the main engine. In a main-engine-failure scenario, however, the duration of the burns can increase drastically. Because of these extended burn durations, it is no longer accurate to approximate each maneuver as impulsive. Thus, the quality of the initial guess supplied degrades significantly. The present study is strictly focused on the second stage of the autonomous targeting process, retargeting the entry interface state using only the resources available onboard at the time (i.e., fuel left and operational engine) based on the initial guess supplied.

Significant research has been done on the subject of optimal finite thrust guidance [5–8]. Among these methods, nonlinear programming is commonly employed in solving optimal and nonlinear targeting problems [7,8]. The process of identifying, numerically, optimal or feasible solutions via nonlinear programming is basically the same. The main difference is that optimization problems require a cost index be specified, and feasibility problems, such as constrained nonlinear targeting, do not. Of course, the identification of feasible solutions that meet all the specified constraints is also accomplished through linear targeting methods [4,9]. These classical methods employ the state transition matrix to compute the necessary constraint gradients during the corrections process. More recent studies [10] employ a similar approach to compute analytic derivatives for implementation in a nonlinear programming process for trajectory optimization. Naturally, a nonlinear process is preferred when the computational resources are available. However, for onboard determination, the optimality of a solution is not as critical as the availability of a feasible solution. In this case, the inherent simplicity of linear targeting algorithms leads to a reduced cost in flight-software development and validation.

Earlier studies consider optimization methods for use during onboard targeting processes. These include the use of a simplified adaptive guidance law for targeting relative to a predetermined

Presented as Paper 2009-6104 at the AIAA Guidance, Navigation, and Control Conference, Chicago, Illinois, 10–13 August 2009; received 14 March 2011; revision received 3 October 2011; accepted for publication 12 November 2011.

*Graduate Student, Department of Aerospace Engineering, 210 E. 24th Street.

†Assistant Professor, Department of Aerospace Engineering, 210 E. 24th Street; marchand@mail.utexas.edu. Associate Fellow AIAA (Corresponding Author).

‡Engineer, DM34/Rendezvous Guidance and Procedures, 2101 NASA Parkway.

§Engineer, DM32/Orbit Flight Dynamics, 2101 NASA Parkway.

¶Engineer, EG-6/Aeroscience and Flight Mechanics Division, 2101 NASA Parkway.

nominal trajectory [11] or implementation of an efficient sequential gradient-restoration algorithm employing multiple subarcs [12]. These studies, though, are tested for orbital transfer and rendezvous, which do not have the third-body effects that so greatly impact the mission in this study. The solution process behind the algorithm presented is partly modeled after the two-level targeter [4,9,13–16] employed during the design of the Genesis trajectory [9,14,15]. However, unlike this earlier development, the present algorithm allows for the incorporation of finite burn maneuvers.

A two-level targeter (or corrector) is primarily based on linear system theory; it uses a time-varying linearized dynamical model and a minimum norm solution to compute solution updates. These linear updates are implemented in the nonlinear system in an iterative corrections process that repeats until a feasible solution is identified in the vicinity of the startup arc. Specifics of the classical two-level corrector will not be reiterated here, but a detailed description of the process is available in previous work [4]. The two-level process offers several advantages: because the updates are based on the linearized model, it is numerically simple and computationally efficient. It does not require knowledge of a nominal solution, relying instead solely on the current path of the vehicle. The two-level corrections process also allows for straightforward addition of path constraints, both those at specific points (e.g., entry interface) [4,9,15,16] and those applied over the trajectory as a whole [16]. However, it was originally designed to use impulsive maneuvers as control variables. In this investigation, the classical impulsive two-level corrections process [4] is modified to incorporate accurate thruster models to allow for burns of finite duration while still retaining the structure and simplicity of the original algorithm so that it is suitable for onboard calculations. The theoretical elements of the formulation are presented next, followed by a series of performance comparisons between the impulsive and finite burn targeters.

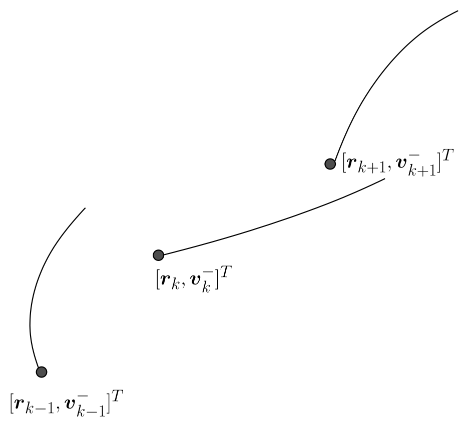
II. Finite Burn Targeting Algorithm

The basic structure of a two-level targeter that incorporates finite burns is generally similar to that of the impulsive two-level targeter [4]. The two-level framework offers significant advantages in terms of computational efficiency, flexibility, and robustness [4]. The design of the finite burn targeting algorithm is intended to retain the advantages of the two-level structure while extending its range of application to include problems with continuous control actuation. Both algorithms use a linearized dynamical model and employ a minimum norm solution in computing the updates to the control variables. The differences, which subsequently lead to added complexity and computational overhead, stem from the increased dimensionality of the state vector associated with any burn arc. Because of the interdependency between these state variables, the partial derivatives are also more complex in nature than those of the impulsive targeter.

Traditionally, an impulsive two-level targeter requires a startup arc represented by a series of patch states. These states, also termed patch points, are selected by the user as representative waypoints along the trajectory. Consider a dynamical system described by

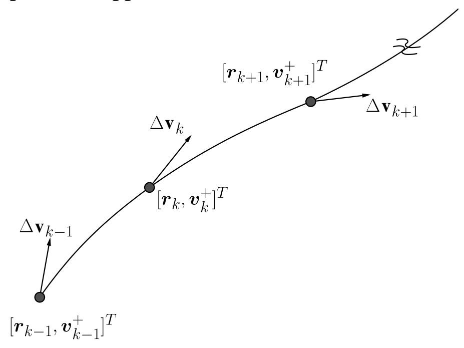
$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t) \tag{1}$$

where $\mathbf{x}(t) = [\,\mathbf{r}(t)^T \quad \mathbf{v}(t)^T\,]^T$. The user supplies the time and state at each patch point, t_{k-1} and $\mathbf{x}_{k-1}^{+} = [\,\mathbf{r}_{k-1}^{T} \quad \mathbf{v}_{k-1}^{+T}\,]^T$ for $k = 2, \cdots, N+1$, respectively. Each state $\mathbf{x}_{k-1}^{+}$ is then numerically integrated forward over an interval $[t_{k-1}, t_k]$ for $k = 2, \cdots, N$. The integrated state, at time t_k, is recorded as $\mathbf{x}_k^{-}$. This is to allow for the possibility that the user-supplied velocity at that point, $\mathbf{v}_k^{+}$, may not coincide with that identified during the propagation, $\mathbf{v}_k^{-}$. Such differences may arise due to a previously scheduled impulsive maneuver at that point or to differences in the models used (two- vs three-body). This is graphically illustrated in Figs. 1a and 1b. Thus, a level I process leads to a trajectory that is continuous in position but, potentially, discontinuous in velocity at certain points. This is rectified by incorporating a level II correction.



$[\boldsymbol{r}_{k+1}, \boldsymbol{v}_{k+1}^{-}]^T$
$[\boldsymbol{r}_{k}, \boldsymbol{v}_{k}^{-}]^T$
$[\boldsymbol{r}_{k-1}, \boldsymbol{v}_{k-1}^{-}]^T$

a) Impulsive two-level targeter: before level I process is applied



$[\boldsymbol{r}_{k+1}, \boldsymbol{v}_{k+1}^{+}]^T$
$\Delta\mathbf{v}_{k+1}$
$\Delta\mathbf{v}_{k}$
$[\boldsymbol{r}_{k}, \boldsymbol{v}_{k}^{+}]^T$
$\Delta\mathbf{v}_{k-1}$
$[\boldsymbol{r}_{k-1}, \boldsymbol{v}_{k-1}^{+}]^T$

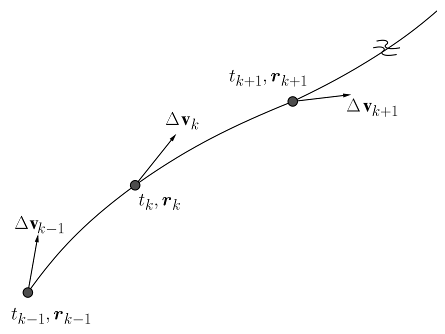
b) Impulsive two-level targeter: after level I process is applied

Fig. 1 Level I process.

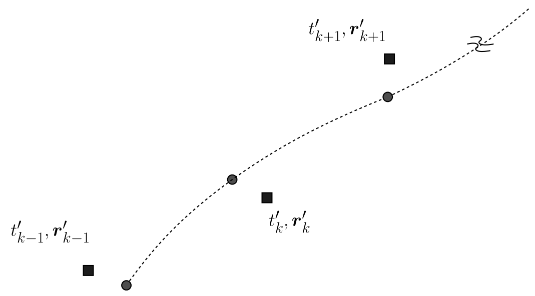
The level II process adjusts the positions and times of each free patch state to drive any of the interior velocity discontinuities to zero, as well as to meet any additional user-specified constraints. Figures 2a–2c show this process graphically. Figure 2a is representative of the scenario in Fig. 1b, where the trajectory is continuous in position but not in velocity. Figure 2b illustrates how the patch state positions, and potentially the associated times, have been adjusted by the level II process. Because the corrections are linear in nature [4], propagation of the updated patch states in the nonlinear system can lead to a trajectory that is, once again, discontinuous in position. The level I process is subsequently applied again to generate an updated trajectory that is continuous in position. The combined level I and level II processes are generally repeated until the user-specified tolerances are met for position and velocity continuity, as well as any additional constraints specified. Additional constraints may include velocity continuity at all patch states, except where maneuvers are allowed, and interior or boundary constraints, among others [4].

It is important to note that the initial guess need not be feasible. That is, position/velocity/time continuity is not necessarily required for the targeter to successfully converge. However, because the overall process is based on linear systems theory, the initial discontinuities can impact the computation time. An initial guess with large discontinuities leads to an increased number of iterations. Naturally, an initial guess with absurdly large discontinuities can lead to nonconvergence. Of course, a low-quality initial guess can have a negative impact on both linear and nonlinear targeting algorithms. However, linear targeters will naturally be more sensitive to large errors. Developing a good initial guess is a problem within itself and is highly dependent on the particular application of interest.

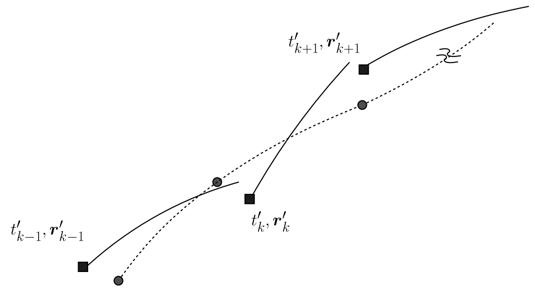
Provided a suitable initial guess is available, the formulation of the impulsive two-level targeter [4] is generalized in nature. As such, it can be applied to any problem that employs impulsive corrections. However, problems that employ continuous control of any kind cannot benefit from this approach, at least not in its original form. The



a) Impulsive two-level targeter: before level II process is applied



b) Impulsive two-level targeter: corrections suggested by level II minimum norm solution



c) Impulsive two-level targeter: after propagating the level II updated patch states

Fig. 2 Level II process.

key to transitioning the methodology to address problems that include segments of continuous control is to formulate the control variables in terms of constant parameters that can be adjusted. For example, if the thrust vector is inertially fixed and the engine only allows fixed thrust or acceleration levels, the control variables become the time of ignition and the direction and duration of the burn. Under similar conditions, if linear steering is allowed, the control variables become the time of ignition, the duration of the burn, the initial burn direction, and the rate of change of the burn direction.

In the classical impulsive two-level targeter, the level I process employs Δ**v**s at the start of each segment to achieve position continuity. These Δ**v**s and, if desired, the time at which the maneuvers are executed are control variables in that case. In a finite burn process, the level I control variables include the ignition time, burn time, and thrust vector parameters. The structure of the finite burn two-level targeter is subsequently developed and presented here.

A. Level I Process

As previously discussed, the application of a level I process [4] to the orbital transfer problem typically involves the identification of an arc that spatially connects two points in space. This is the n-body analog to a two-body Lambert targeter, except the time of flight is not necessarily fixed or prespecified. This ultimately reduces to a form of

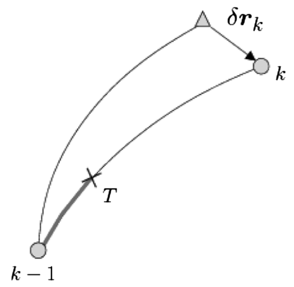


Fig. 3 Level I process.

linear differential correction, where Δ**v**s are adjusted to meet the specified goals. In the present study, however, impulsive maneuvers do not adequately model the true nature of the burn implementation. Thus, the level I process traditionally employed in the two-level targeter [4] requires some modification to incorporate finite burn arcs. Consider a segment defined by patch points $k-1$ and k, as shown in Fig. 3. In a level I process that employs finite burns rather than impulsive maneuvers, the burn arc is treated as a subsegment of the total arc between patch points $k-1$ and k. Point T in Fig. 3 denotes the termination of the burn subsegment.

In the impulsive case, the time derivative $\dot{\mathbf{x}}(t) = [\dot{\mathbf{r}}(t)^T \quad \dot{\mathbf{v}}(t)^T]^T$ is a function only of the state $\mathbf{x}(t)$ and the time t. With the addition of finite burn subarcs, however, the contribution of the thrust acceleration must also be taken into account:

$$\dot{\mathbf{x}}(t) = \tilde{\mathbf{f}}[\mathbf{x}(t), t, m(t), \tilde{\mathbf{u}}(t)] \tag{2}$$

where $m(t)$ is the mass of the vehicle, and $\tilde{\mathbf{u}}(t)$ is the full thrust vector (direction and magnitude). In this investigation, $\tilde{\mathbf{u}}(t)$ is assumed to be constant for $t_j \le t < t_{T_j}$ and zero for $t_{T_j} < t \le t_{j+1}$. Note that the j subscripts indicate the point at which the jth finite burn maneuver occurs; they do not necessarily represent sequential patch points. The equation for $\tilde{\mathbf{u}}(t)$ over the entire trajectory is

$$\tilde{\mathbf{u}}(t) = \sum_{j=1}^{n_{\Delta \mathbf{v}}} \tilde{\mathbf{u}}_j [1(t-t_j) - 1(t-t_{T_j})] \tag{3}$$

where $\tilde{\mathbf{u}}_j$ is the thrust vector at maneuver point j, $1(t)$ is the unit step function, and $n_{\Delta \mathbf{v}}$ represents the total number of finite burn maneuvers along the entire trajectory from start to finish. A plot of the thrust magnitude for a three-burn sequence is shown in Fig. 4.

For the system to remain consistent with Eq. (1), it is necessary to consider an augmented state vector $\mathbf{x}^+_{k-1} = [\mathbf{r}^T_{k-1} \quad \mathbf{v}^{+T}_{k-1} \quad m_{k-1} \quad \dot{m}_{g_{k-1}} \quad \mathbf{u}^T_{k-1}]^T$, where m_{k-1} and $\dot{m}_{g_{k-1}}$ represent the spacecraft mass and the propellant flow rate associated with patch point $k-1$, respectively. The variable $\mathbf{u}_{k-1}$ can be defined as either the full thrust vector $\tilde{\mathbf{u}}_{k-1}$ or as a vector describing only the direction of the thrust. This will be discussed in more detail in the following section. In

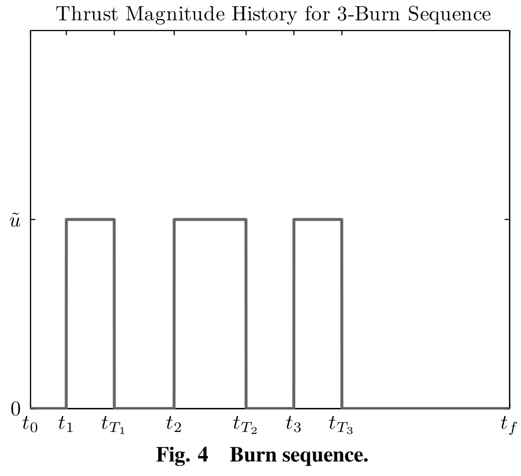


Fig. 4 Burn sequence.

determining the level I algorithm, the goal is to identify a relation between the target, which is the terminal position vector at point k ($\mathbf{r}_k$), and the control variables. The control variables are the vector $\mathbf{u}_{k-1}$, representing either the full thrust or thrust direction, and the cutoff time of the burn (t_T). The variational equation for the burn subsegment is

$$\begin{bmatrix} \delta\mathbf{r}_T - \mathbf{v}_T^-\delta t_T \\ \delta\mathbf{v}_T^- - \mathbf{a}_T^-\delta t_T \\ \delta m_T^- + \dot{m}_{g_T}^-\delta t_T \\ \delta\dot{m}_{g_T}^- - \ddot{m}_{g_T}^-\delta t_T \\ \delta\mathbf{u}_T^- - \dot{\mathbf{u}}_T^-\delta t_T \end{bmatrix} = \Phi(T, k-1)\begin{bmatrix} \delta\mathbf{r}_{k-1} - \mathbf{v}_{k-1}^+\delta t_{k-1} \\ \delta\mathbf{v}_{k-1}^+ - \mathbf{a}_{k-1}^+\delta t_{k-1} \\ \delta m_{k-1}^+ + \dot{m}_{g_{k-1}}^+\delta t_{k-1} \\ \delta\dot{m}_{g_{k-1}}^+ - \ddot{m}_{g_{k-1}}^+\delta t_{k-1} \\ \delta\mathbf{u}_{k-1}^+ - \dot{\mathbf{u}}_{k-1}^+\delta t_{k-1} \end{bmatrix} \tag{4}$$

where $\Phi(T, k-1)$ is the state transition matrix between patch point k and point T. As in the impulsive formulation, the state transition matrix is partitioned into submatrices corresponding to each state:

$$\Phi(T, k-1) = \begin{bmatrix} A_{T,k-1} & B_{T,k-1} & E_{T,k-1} & F_{T,k-1} & G_{T,k-1} \\ C_{T,k-1} & D_{T,k-1} & H_{T,k-1} & I_{T,k-1} & J_{T,k-1} \\ K_{T,k-1} & L_{T,k-1} & M_{T,k-1} & N_{T,k-1} & O_{T,k-1} \\ P_{T,k-1} & Q_{T,k-1} & R_{T,k-1} & S_{T,k-1} & T_{T,k-1} \\ U_{T,k-1} & V_{T,k-1} & W_{T,k-1} & X_{T,k-1} & Y_{T,k-1} \end{bmatrix} \tag{5}$$

For the subsequent coasting subsegment, the variational equation, with partitioned state transition matrix, takes the same form as in the impulsive formulation [4]:

$$\begin{bmatrix} \delta\mathbf{r}_k - \mathbf{v}_k^-\delta t_k \\ \delta\mathbf{v}_k^- - \mathbf{a}_k^-\delta t_k \end{bmatrix} = \begin{bmatrix} A_{k,T} & B_{k,T} \\ C_{k,T} & D_{k,T} \end{bmatrix}\begin{bmatrix} \delta\mathbf{r}_T - \mathbf{v}_T^+\delta t_T \\ \delta\mathbf{v}_T^+ - \mathbf{a}_T^+\delta t_T \end{bmatrix} \tag{6}$$

For this formulation, both the initial and final times of the arc (t_{k-1} and t_k, respectively) are fixed, though that is not a requirement. The initial position $\mathbf{r}_{k-1}$, velocity $\mathbf{v}_{k-1}^+$, and mass m_{k-1}^+ are also fixed, as well as the mass flow rate, $\dot{m}_{g_{k-1}}^+$. It is important to note that $\mathbf{v}_T^+ = \mathbf{v}_T^-$ (and, therefore, $\delta\mathbf{v}_T^+ = \delta\mathbf{v}_T^-$). Furthermore, $\delta\mathbf{v}_T^+ - \mathbf{a}_T^+\delta t_T = \delta\mathbf{v}_T^- - \mathbf{a}_T^-\delta t_T + (\mathbf{a}_T^- - \mathbf{a}_T^+)\delta t_T$. Incorporating these substitutions, the first two vector variational equations from Eqs. (4) and (6) can be combined to give an expression for $\delta\mathbf{r}_k$:

$$\delta\mathbf{r}_k = [\,(A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) \quad B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+)\,]\begin{bmatrix} \delta\mathbf{u}_{k-1}^+ \\ \delta t_T \end{bmatrix} \tag{7}$$

As in the impulsive level I method [4], a minimum norm solution is selected to obtain the desired change in the control variables:

$$\begin{bmatrix} \delta\mathbf{u}_{k-1}^+ \\ \delta t_T \end{bmatrix} = \tilde{M}^T(\tilde{M}\tilde{M}^T)^{-1}\delta\mathbf{r}_k \tag{8}$$

where

$$\tilde{M} = [\,(A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) \quad B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+)\,] \tag{9}$$

A minimum norm solution identifies the smallest change in the control parameters, in this case $\delta\mathbf{u}_{k-1}^+$ and δt_T, that lead to the desired changes in the constraint errors. Of course, these corrections are linear in nature, and as such, an iterative process is required to converge on the specified constraints in the nonlinear system.

A potential issue that can occur with this formulation is that the converged burn cutoff time t_T may fall after the terminal segment time t_k. To resolve this, t_k becomes a control parameter (and is thus allowed to vary), and an additional constraint is appended to enforce $t_T \le t_k$. This constraint, $\alpha_t = t_T - t_k$, would only be active if t_T is greater than t_k. The level I constraint equation, then, becomes

$$\begin{bmatrix} \delta\mathbf{r}_k \\ \delta\alpha_t \end{bmatrix} = \begin{bmatrix} \frac{\partial\mathbf{r}_k}{\partial\mathbf{u}_{k-1}^+} & \frac{\partial\mathbf{r}_k}{\partial t_T} & \frac{\partial\mathbf{r}_k}{\partial t_k} \\ \frac{\partial\alpha_t}{\partial\mathbf{u}_{k-1}^+} & \frac{\partial\alpha_t}{\partial t_T} & \frac{\partial\alpha_t}{\partial t_k} \end{bmatrix}\begin{bmatrix} \delta\mathbf{u}_{k-1}^+ \\ \delta t_T \\ \delta t_k \end{bmatrix} \tag{10}$$

and the three control parameters are, again, found using the minimum norm solution. In practice, this situation can be avoided through proper patch-point selection; the duration of a segment containing a burn arc should always be significantly greater than the expected burn time.

To determine an initial guess for the finite burn parameters (i.e., thrust direction, $\mathbf{u}_{k-1}$, and burn cutoff time, t_T), the impulsive level I process is first used to compute an impulsive correction. The impulsive $\Delta\mathbf{v}$ direction is used as an initial guess for the thrust direction. If $\mathbf{u}_{k-1}$ is defined as the full thrust vector, then the initial guess for the thrust magnitude is the given thrust of the engine. Finally, an initial guess for the burn duration is deduced using the rocket equation:

$$\Delta\mathbf{v}_{k-1} = -I_{\text{sp}}g_0 \ln\left(1 - \frac{\dot{m}_{g_{k-1}}\Delta t_{\text{burn}}}{m_{k-1}}\right) \tag{11}$$

and rearranging to obtain

$$\Delta t_{\text{burn}} = \frac{m_{k-1}}{\dot{m}_{g_{k-1}}}\left(1 - e^{\frac{-\Delta\mathbf{v}_{k-1}\dot{m}_{g_{k-1}}}{u_{k-1}}}\right) = \frac{m_{k-1}}{\dot{m}_{g_{k-1}}}\left(1 - e^{\frac{-\Delta\mathbf{v}_{k-1}}{I_{\text{sp}}g_0}}\right) \tag{12}$$

From the burn duration, the cutoff time is calculated as $\delta t_T = \delta t_{k-1} + \Delta t_{\text{burn}}$.

The impulsive burn approximation becomes less valid as the finite burn time increases. Therefore, the terminal error after the first iteration can be very large when the burn duration is long. The burn direction is assumed to be constant throughout the entire maneuver, and so, small errors in direction can be greatly magnified by the end of a long burn.

1. *Controlling Thrust Magnitude in the Level I Process*

The state $\mathbf{u}_{k-1}$ can be defined as either the full thrust vector $\tilde{\mathbf{u}}_{k-1}$ or as a thrust direction vector that does not affect the thrust magnitude. Clearly, for cases in which variable thrust is permissible, the choice of $\mathbf{u}_{k-1}$ as the full thrust vector $\tilde{\mathbf{u}}_{k-1}$ is appropriate. Most applications of the two-level targeter, however, assume a constant thrust engine. In these cases, if the full thrust vector definition is used for $\mathbf{u}_{k-1}$, changes in the thrust magnitude resulting from $\delta\mathbf{u}_{k-1}$ must be controlled to ensure that the thrust magnitude in the converged solution is equal to the given (constant) thrust of the engine.

If one defines $\mathbf{u}_{k-1}$ as the full thrust vector $\tilde{\mathbf{u}}_{k-1}$, changes in the thrust magnitude can be controlled using a thrust biasing technique. This technique is similar to target position biasing commonly used in perturbed Lambert targeting. Because $\mathbf{u}_{k-1}$ is defined as $\tilde{\mathbf{u}}_{k-1}$, the correction $\delta\mathbf{u}_{k-1}$ implies the correction $\delta\tilde{\mathbf{u}}_{k-1}$, thus changing both the thrust direction and magnitude in Eq. (3). (Because $\dot{m}_{g_{k-1}}$ is assumed constant, in effect, this creates a fictitious variable I_{sp} engine because $\dot{m}_{g_{k-1}} = \tilde{u}_{k-1}/I_{\text{sp}}g_0$ but $\tilde{u}_{k-1}$ is changing after each iteration. This is only temporary, however, because the thrust biasing technique ultimately adjusts the thrust magnitude back to the desired value; hence, I_{sp} ultimately returns to its assumed value as well.) When the level I process converges, $\delta\mathbf{r}_k = \mathbf{0}$, but the converged thrust magnitude will be different from the desired value. Let $\Delta\tilde{u}_{k-1} = \tilde{u}_{\text{engine}} - \tilde{u}_{\text{converged}}$. The level I process is then repeated with the same initial guess for thrust direction, but the initial guess for the thrust magnitude is biased such that $\tilde{u}_{\text{initial(new)}} = \tilde{u}_{\text{initial(old)}} + \Delta\tilde{u}_{k-1}$. The initial guess for the burn time is also updated using Eq. (12) to reflect this change in thrust magnitude. A graphical representation of this process is shown in Fig. 5. When the level 1 process reconverges, the thrust magnitude will be much closer to the desired value. This process of biasing and reconverging is repeated (typically three to five iterations) until $\Delta\tilde{u}_{k-1}$ is within tolerance. For the simulations in Sec. III.B, $\mathbf{u}_{k-1}$ is defined as the full thrust vector $\tilde{\mathbf{u}}_{k-1}$, and the thrust biasing technique is used to control the thrust magnitude. Note that using the thrust biasing technique introduces an additional layer of iteration within the level I process, which can potentially increase the runtime of the algorithm.

If, instead, one defines $\mathbf{u}_{k-1}$ as a thrust direction vector, the thrust magnitude will remain unaffected in the level I process and can be fixed at the given (constant) value. Let $\mathbf{u}_{k-1}$ be a vector that describes the thrust direction but is not required to be a unit vector. The full thrust vector in Eq. (3) is then given by

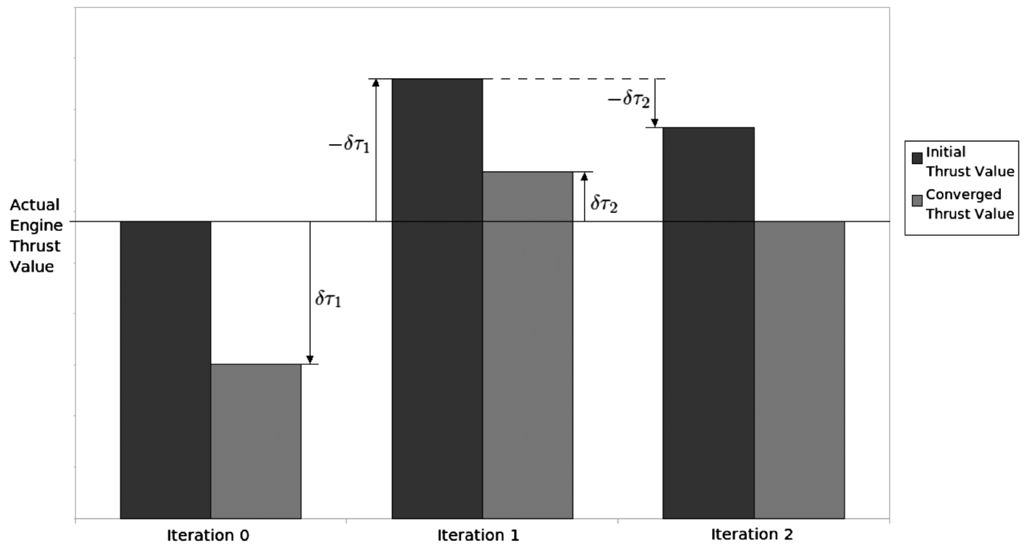


Fig. 5 Thrust biasing in the level I process.

$\tilde{\mathbf{u}}_{k-1} = \tilde{u}_{\text{engine}}[(\mathbf{u}_{k-1})/(\|\mathbf{u}_{k-1}\|)]$. Although the correction $\delta\mathbf{u}_{k-1}$ can change the magnitude of $\mathbf{u}_{k-1}$, the magnitude of the full thrust vector $\tilde{\mathbf{u}}_{k-1}$ remains fixed at $\tilde{u}_{\text{engine}}$. For the simulations in Sec. III.A, $\mathbf{u}_{k-1}$ is defined as a thrust direction vector.

2. *Variable Scaling*

A well-known tool for aiding (and sometimes enabling) the convergence of iterative processes, such as nonlinear targeting, is the scaling of both the control variables and constraints so that the partial derivatives of the constraints with respect to the control variables are $O(1)$. This enables the targeter to evenly adjust control variables and meet constraints of varying orders of magnitude.

To implement scaling in the level I process, first note that Eq. (7) relates variations in the constraints ($\delta\mathbf{r}_k$) to variations in the controls ($\delta\mathbf{u}^+_{k-1}$ and δt_T) through partial derivatives as

$$\delta\mathbf{r}_k = \begin{bmatrix} \frac{\partial\mathbf{r}_k}{\partial\mathbf{u}^+_{k-1}} & \frac{\partial\mathbf{r}_k}{\partial t_T} \end{bmatrix} \begin{bmatrix} \delta\mathbf{u}^+_{k-1} \\ \delta t_T \end{bmatrix} \tag{13}$$

Before forming the minimum norm solution, the controls, constraints, and partial derivatives are scaled by appropriate factors so that

$$\mathbf{u}^+_{k-1_{\text{scl}}} = \frac{\mathbf{u}^+_{k-1}}{u_{\text{scl}}}, \qquad t_{T_{\text{scl}}} = \frac{t_T}{t_{\text{scl}}}, \qquad \delta\mathbf{r}_{k_{\text{scl}}} = \frac{\delta\mathbf{r}_k}{r_{\text{scl}}}$$

$$\frac{\partial\mathbf{r}_{k_{\text{scl}}}}{\partial\mathbf{u}^+_{k-1_{\text{scl}}}} = \frac{\partial\mathbf{r}_k}{\partial\mathbf{u}^+_{k-1}}\cdot\frac{u_{\text{scl}}}{r_{\text{scl}}}, \qquad \frac{\partial\mathbf{r}_{k_{\text{scl}}}}{\partial t_{T_{\text{scl}}}} = \frac{\partial\mathbf{r}_k}{\partial t_T}\cdot\frac{t_{\text{scl}}}{r_{\text{scl}}}$$

therefore,

$$\delta\mathbf{r}_{k_{\text{scl}}} = \begin{bmatrix} \frac{\partial\mathbf{r}_{k_{\text{scl}}}}{\partial\mathbf{u}^+_{k-1_{\text{scl}}}} & \frac{\partial\mathbf{r}_{k_{\text{scl}}}}{\partial t_{T_{\text{scl}}}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{u}^+_{k-1_{\text{scl}}} \\ \delta t_{T_{\text{scl}}} \end{bmatrix} \tag{14}$$

The scaled controls are then updated using the minimum norm solution. To begin the next iteration, the controls are unscaled and then used in Eq. (3) to determine the new constraint variations and partial derivatives. As before, these are rescaled using the same scale factors, a new minimum norm solution is found, and the process is repeated until convergence.

Choosing appropriate scale factors is more art than science and is typically problem-specific. For the level I process, one possible approach is to first choose scales for the constraints, say $r_{\text{scl}} = 1$. Next, choose u_{scl} and t_{scl} so that the partial derivatives in Eq. (14) are $O(1)$. Run the level I process. If a large number of iterations are required to achieve convergence, it may be necessary to stop the process after some number of iterations and reexamine the partial derivatives. If they are far from $O(1)$, adjust u_{scl} and t_{scl} to bring the partial derivatives back to $O(1)$ and resume the process using the current values of the controls.

Finally, note that scaling can also be implemented in the level II process. In level II, the controls are $\mathbf{r}_{k-1}$, t_{k-1}, $\mathbf{r}_k$, t_k, $\mathbf{r}_{k+1}$, and t_{k+1}, the constraints are $\Delta\mathbf{v}_k$ and any terminal trajectory conditions, and the partial derivatives are given in Eq. (18). Note that, in level I, $\mathbf{r}_k$ was a constraint. In level II, however, $\mathbf{r}_{k-1}$, $\mathbf{r}_k$, and $\mathbf{r}_{k+1}$ are now controls. Similarly, in the level I process, t_T was a control and represented a burn time. In level II, t_{k-1}, t_k, and t_{k+1} are also controls but now represent patch-point epochs. A similar scaling approach is also taken for the maneuver sum constraint.

B. Level II Process

The finite burn level II process, like the impulsive level II correction, uses the positions and times of the patch points as control variables. In the classical two-level corrector [4], velocity discontinuities between coast segments arise due to the level I process. This is also applicable to the finite burn formulation, except at the point where a finite burn maneuver is initiated. Here, the burn segment is always assumed to start with the same initial velocity as the terminal velocity of the preceding arc. Thus, a velocity discontinuity can occur, during the level I process, at the point where the coast subarc, as defined in Fig. 3, joins with the following trajectory segment. Although this problem, at first, seems identical to the impulsive maneuver targeting because the velocity discontinuity falls between two coast arcs, the partial derivatives for $\delta\mathbf{v}^-_k$ with respect to the control variables $\delta\mathbf{r}_{k-1}$, t_{k-1}, $\delta\mathbf{r}_k$, and t_k differ due to the thrust segment at the beginning of the arc.

Recall from the level I formulation that $\mathbf{v}^-_T = \mathbf{v}^+_T$ at the terminal point of the burn arc and, thus, that $\delta\mathbf{v}^+_T - \mathbf{a}^+_T\delta t_T = \delta\mathbf{v}^-_T - \mathbf{a}^-_T\delta t_T + (\mathbf{a}^-_T - \mathbf{a}^+_T)\delta t_T$. For the level II process, $\delta m^+_{k-1} = \dot{m}^+_{g_{k-1}} = 0$ and $\dot{\mathbf{u}}^+_{k-1} = \mathbf{0}$. It is still assumed that $\dot{m}_g$ is a fixed constant, i.e., $\delta\dot{m}^+_{g_{k-1}} = 0$. Note also that the terminal time of the burn, t_T, is directly related to the time at patch point $k-1$: $t_T = t_{k-1} + \Delta t_{\text{burn}}$, where Δt_{burn} is the duration of the burn and is held constant in the level II process. Using these relationships and assumptions, along with Eqs. (4) and (6), an expression is found for $\delta\mathbf{v}^-_k$ in terms of the state at patch point $k-1$ and the state transition matrix:

$$\begin{aligned}\delta\mathbf{v}^-_k = {} & C_{k,T}[A_{T,k-1}(\delta\mathbf{r}_{k-1} - \mathbf{v}^+_{k-1}\delta t_{k-1}) + B_{T,k-1}(\delta\mathbf{v}^+_{k-1} - \mathbf{a}^+_{k-1}\delta t_{k-1}) \\ & + E_{T,k-1}\dot{m}^+_{g_{k-1}}\delta t_{k-1} + G_{T,k-1}\delta\mathbf{u}^+_{k-1}] + D_{k,T}[C_{T,k-1}(\delta\mathbf{r}_{k-1} - \mathbf{v}^+_{k-1}\delta t_{k-1}) \\ & + D_{T,k-1}(\delta\mathbf{v}^+_{k-1} - \mathbf{a}^+_{k-1}\delta t_{k-1}) + H_{T,k-1}\dot{m}^+_{g_{k-1}}\delta t_{k-1} \\ & + J_{T,k-1}\delta\mathbf{u}^+_{k-1} + (\mathbf{a}^-_T - \mathbf{a}^+_T)(\delta t_{k-1} + \Delta t_{\text{burn}})] + \mathbf{a}^-_k\delta t_k\end{aligned} \tag{15}$$

To write $\delta\mathbf{v}^-_k$ only in terms of the level II control variables, the first vector equation from Eq. (4) is used to solve for $\delta\mathbf{v}^+_{k-1}$, $\delta\mathbf{u}^+_{k-1}$, and

Δt_{burn} in terms of those control variables. From the minimum norm solution,

$$\begin{bmatrix} \delta \mathbf{v}_{k-1}^{+} \\ \delta \mathbf{u}_{k-1}^{+} \\ \Delta t_{\mathrm{burn}} \end{bmatrix} = Z^T(ZZ^T)^{-1}[\delta \mathbf{r}_k - \mathbf{v}_k^{-}\delta t_k - (A_{k,T}A_{T,k-1} + B_{k,T}C_{T,k-1})(\delta \mathbf{r}_{k-1} - \mathbf{v}_{k-1}^{+}\delta t_{k-1}) + (A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1})\mathbf{a}_{k-1}^{+}\delta t_{k-1} - (A_{k,T}E_{T,k-1} + B_{k,T}H_{T,k-1})\dot{m}_{g_{k-1}}^{+}\delta t_{k-1} - B_{k,T}(\mathbf{a}_T^{-} - \mathbf{a}_T^{+})\delta t_{k-1}] \tag{16}$$

where $Z = [(A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1})\ (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1})\ B_{k,T}(\mathbf{a}_T^{-} - \mathbf{a}_T^{+})]$. With this expression, the partial derivatives of $\Delta \mathbf{v}_k$ with respect to each control variable can be found using the same method as in the impulsive formulation. Let

$$\tilde{Z} = [(C_{k,T}B_{T,k-1} + D_{k,T}D_{T,k-1})\ (C_{k,T}G_{T,k-1} + D_{k,T}J_{T,k-1})\ D_{k,T}(\mathbf{a}_T^{-} - \mathbf{a}_T^{+})]Z^T(ZZ^T)^{-1} \tag{17}$$

Then, because it is assumed that the arc from patch point k to $k+1$ is a coast arc, the partial derivatives of $\Delta \mathbf{v}_k$ are

$$\begin{aligned}
\frac{\partial \Delta \mathbf{v}_k}{\partial \mathbf{r}_{k-1}} &= -[(C_{k,T}A_{T,k-1} + D_{k,T}C_{T,k-1}) - \tilde{Z}(A_{k,T}A_{T,k-1} + B_{k,T}C_{T,k-1})] \\
\frac{\partial \Delta \mathbf{v}_k}{\partial t_{k-1}} &= -\{[(C_{k,T}E_{T,k-1} + D_{k,T}H_{T,k-1}) - \tilde{Z}(A_{k,T}E_{T,k-1} + B_{k,T}H_{T,k-1})]\dot{m}_{g_{k-1}}^{+} - [(C_{k,T}B_{T,k-1} + D_{k,T}D_{T,k-1}) \\
&\quad - \tilde{Z}(A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1})]\mathbf{a}_{k-1}^{+} - [(C_{k,T}A_{T,k-1} + D_{k,T}C_{T,k-1}) \\
&\quad - \tilde{Z}(A_{k,T}A_{T,k-1} + B_{k,T}C_{T,k-1})]\mathbf{v}_{k-1}^{+} + (D_{k,T} - \tilde{Z}B_{k,T})(\mathbf{a}_T^{-} - \mathbf{a}_T^{+})\} \\
\frac{\partial \Delta \mathbf{v}_k}{\partial \mathbf{r}_k} &= -B_{k+1,k}^{-1}A_{k+1,k} - \tilde{Z} \\
\frac{\partial \Delta \mathbf{v}_k}{\partial t_k} &= B_{k+1,k}^{-1}A_{k+1,k}\mathbf{v}_k^{+} + \mathbf{a}_k^{+} - (\mathbf{a}_k^{-} - \tilde{Z}\mathbf{v}_k^{-}) \\
\frac{\partial \Delta \mathbf{v}_k}{\partial \mathbf{r}_{k+1}} &= B_{k+1,k}^{-1}, \qquad \frac{\partial \Delta \mathbf{v}_k}{\partial t_{k+1}} = -B_{k+1,k}^{-1}\mathbf{v}_{k+1}^{-}
\end{aligned} \tag{18}$$

These partials are employed in the standard level II process [4].

C. Maneuver Sum Constraint

In addition to the velocity continuity constraint, endpoint and interior path constraints may be imposed during the level II process [16]. One such constraint is on the total $\Delta \mathbf{v}$ sum of the maneuvers. The finite burn formulation of this constraint is based on the impulsive maneuver sum constraint [17]. Only the composition of the associated partial derivatives and the error calculation changes.

To derive the burn maneuver constraint, it is necessary to determine the partial derivatives of the magnitude of $\Delta \mathbf{v}_k$, i.e., the maneuver that results from the burn at patch point k, with respect to the level II control variables. From the rocket equation, $\Delta \mathbf{v}_k$ is given by

$$\Delta \mathbf{v}_k = -I_{\mathrm{sp}} g_0 \ln\left(1 - \frac{\dot{m}_{g_k}\Delta t_{\mathrm{burn}}}{m_k}\right) \tag{19}$$

Because I_{sp}, g_0, and $\dot{m}_{g_k}$ are fixed, the cost of a maneuver at patch point k depends on the duration of the burn at k and the initial mass. The burn duration Δt_{burn} and mass m_k are, in turn, functions of the control variables, and the chain rule is used to determine the partials of $\Delta \mathbf{v}_k$ with respect to the controls. The partial derivative of $\Delta \mathbf{v}_k$ with respect to Δt_{burn} at patch point k is given by

$$\frac{\partial \Delta \mathbf{v}_k}{\partial \Delta t_{\mathrm{burn}}} = I_{\mathrm{sp}} g_0 \left(\frac{m_k}{m_k - \dot{m}_{g_k}\Delta t_{\mathrm{burn}}}\right)\left(\frac{\dot{m}_{g_k}}{m_k}\right) \tag{20}$$

With respect to m_k, the partial derivative is

$$\frac{\partial \Delta \mathbf{v}_k}{\partial m_k} = \frac{I_{\mathrm{sp}} g_0}{m_k}\left(\frac{\dot{m}_{g_k}\Delta t_{\mathrm{burn}}}{m_k - \dot{m}_{g_k}\Delta t_{\mathrm{burn}}}\right) \tag{21}$$

The partial derivatives of Δt_{burn} with respect to the control variables are determined using the variational equations from points $k-1$ to k, k to T (the termination of the burn segment), and $k+1$ to T. Recalling that $\Delta t_{\mathrm{burn}} = t_T - t_k$, the partials are found to be

$$\begin{aligned}
\frac{\partial \Delta t_{\mathrm{burn}}}{\partial \mathbf{r}_{k-1}} &= -\frac{\hat{\mathbf{u}}_k^T}{\|\Delta \mathbf{a}_T\|}(D_{T,k}B_{k-1,k}^{-1} - \tilde{S}B_{T,k}B_{k-1,k}^{-1}), \\
\frac{\partial \Delta t_{\mathrm{burn}}}{\partial t_{k-1}} &= \frac{\hat{\mathbf{u}}_k^T}{\|\Delta \mathbf{a}_T\|}(D_{T,k}B_{k-1,k}^{-1} - \tilde{S}B_{T,k}B_{k-1,k}^{-1})\mathbf{v}_{k-1}^{+}, \\
\frac{\partial \Delta t_{\mathrm{burn}}}{\partial \mathbf{r}_k} &= -\frac{\hat{\mathbf{u}}_k^T}{\|\Delta \mathbf{a}_T\|}[(C_{T,k} + D_{T,k}D_{k,k-1}B_{k,k-1}^{-1}) - \tilde{S}(A_{T,k} + B_{T,k}D_{k,k-1}B_{k,k-1}^{-1})], \\
\frac{\partial \Delta t_{\mathrm{burn}}}{\partial t_k} &= \frac{\hat{\mathbf{u}}_k^T}{\|\Delta \mathbf{a}_T\|}\{[(C_{T,k} + D_{T,k}D_{k,k-1}B_{k,k-1}^{-1}) - \tilde{S}(A_{T,k} + B_{T,k}D_{k,k-1}B_{k,k-1}^{-1})]\mathbf{v}_k^{-} - (D_{T,k} - \tilde{S}B_{T,k})(\mathbf{a}_k^{-} - \mathbf{a}_k^{+}) \\
&\quad - (H_{T,k} - \tilde{S}E_{T,k})\dot{m}_g\}, \qquad \frac{\partial \Delta t_{\mathrm{burn}}}{\partial \mathbf{r}_{k+1}} = \frac{\hat{\mathbf{u}}_k^T}{\|\Delta \mathbf{a}_T\|}(C_{T,k+1} - \tilde{S}A_{T,k+1}), \\
\frac{\partial \Delta t_{\mathrm{burn}}}{\partial t_{k+1}} &= -\frac{\hat{\mathbf{u}}_k^T}{\|\Delta \mathbf{a}_T\|}[(C_{T,k+1} - \tilde{S}A_{T,k+1})\mathbf{v}_{k+1}^{-} + (D_{T,k+1} - \tilde{S}B_{T,k+1})\mathbf{a}_{k+1}^{-}]
\end{aligned} \tag{22}$$

where $S = [-G_{T,k}\quad B_{T,k+1}]$ and $\tilde{S} = [-J_{T,k}\quad D_{T,k+1}]S^T(SS^T)^{-1}$. It is more complicated to determine the partials of m_k with respect to the control variables. Because $\dot{m}_g$ is a fixed, constant value, m_k depends only on the previous burn durations. Thus, the initial mass at the beginning of a maneuver will have a dependence on the positions and times associated with any previous maneuvers that have occurred. Using the chain rule, the final form of the partial derivative of the constraint α (the sum of all the burn $\Delta \mathbf{v}$s) with respect to any control variable β_k in the set of control variables associated with patch point k is

$$\frac{\partial \alpha}{\partial \beta_k} = \sum_{n=1}^{N_{\Delta \mathbf{v}}} \frac{\partial \Delta \mathbf{v}_n}{\partial \beta_k} \tag{23}$$

where $N_{\Delta \mathbf{v}}$ is the total number of maneuvers implemented along the trajectory and

$$\frac{\partial \Delta \mathbf{v}_n}{\partial \beta_k} = \frac{\partial \Delta \mathbf{v}_n}{\partial \Delta t_{\mathrm{burn}_n}}\frac{\partial \Delta t_{\mathrm{burn}_n}}{\partial \beta_k} + \frac{\partial \Delta \mathbf{v}_n}{\partial m_n}\frac{\partial m_n}{\partial \beta_k} \tag{24}$$

Because the mass at the time of a burn, m_n, depends on the propellant mass consumed during the previous burns,

$$\frac{\partial m_n}{\partial \beta_k} = \frac{\partial m_{n-1}}{\partial \beta_k} - \dot{m}_{g_{n-1}}\frac{\partial \Delta t_{\mathrm{burn}_{n-1}}}{\partial \beta_k} \tag{25}$$

A similar relationship exists for the remaining mass partials (m_{n-2} to m_1) with respect to β_1. These partials are then employed during the level II process [4].

III. Sample Applications

Over the course of the following sections, the finite burn targeting algorithm is tested and validated through two examples. The first focuses on a precision entry application, which involves a lunar return trajectory and the associated trans-Earth injection (TEI) sequence. The second example considers a Mars encounter. It is important to note that the finite burn targeting algorithm implementation used in both cases is identical; the only things that vary are the user-specified trajectory and mission constraints. The methodology already described is otherwise unchanged.

A. Example 1: Orion Trans-Earth Injection Simulation and Results

The finite burn targeting algorithm is applied, in this section, to the three-burn TEI phase of a lunar return trajectory. The goal is to identify the three-maneuver sequence that meets the specified set of lunar departure and Earth-entry interface constraints without violating the available fuel budget. The performance of the finite burn algorithm is contrasted against that of the impulsive algorithm. As a final metric, an optimal trajectory is also generated for performance comparisons.

For each case considered, the parameters that define the initial low-lunar-orbit departure are the same. These conditions are listed in Table 1; engine parameters for the finite burn maneuvers are given in Table 2. Components of the initial state in Table 1 are in the J2000 moon-centered inertial (MCI) frame. Similarly, the Earth-entry interface conditions targeted are listed in Table 3. These terminal constraints represent entry parameters that would allow for a safe crewed reentry into the Earth's atmosphere [18–21]. The epoch of the entry interface conditions is unconstrained. Convergence is achieved when the terminal and path constraints are met within the tolerances listed in Table 4. Finally, for cases in which variable scaling is implemented, the scale values used are listed in Table 5.

1. *Finite Burn Example with Main Engine*

The first case is representative of a nominal Earth return during which the maneuvers are performed by the vehicle's main engine. The initial guess file consists of 12 patch points (position, velocity, and time) taken from an optimized finite burn trajectory. It should be noted that the initial guess does not contain burn duration or direction information for the maneuvers; the algorithm must target these values on its own. The first patch point corresponds to a state on the initial lunar parking orbit. The interior patch points correspond to the states and epochs at each of the maneuver locations (TEI-1, 2, 3 and TCM 1, 2, 3) and some additional waypoints along the trajectory. The final patch point in the initial guess is the state and epoch at the desired entry interface.

For this case, both the impulsive targeter and the finite burn targeter are executed to find a feasible trajectory that satisfies the specified terminal constraints, while keeping the $\Delta\mathbf{v}$ sum of the individual maneuvers within the available fuel budget. The finite burn algorithm is run both with and without variable scaling implemented. Table 6 compares the individual maneuvers, final $\Delta\mathbf{v}$ sum, and number of iterations required for convergence for the impulsive solution and the scaled and unscaled finite burn (FB) solutions. The burn parameters for each finite burn maneuver are given in Table 7 for the unscaled algorithm and in Table 8 for the scaled algorithm.

The individual maneuvers and total $\Delta\mathbf{v}$ sum for the finite burn targeter are fairly similar to the impulsive targeting results, which is to be expected given that the burn durations with the main engine are short enough for an impulsive assumption to be used. It should be noted, though, that, for this particular case, the impulsive algorithm requires several more iterations to converge than the finite burn algorithm does. This suggests that the impulsive assumption, while still valid, may be reaching its limit.

2. *Finite Burn Example with Auxiliary Engines*

For this example, a main engine failure is assumed to occur after TEI-1, and the auxiliary engines are used to perform the final two maneuvers. The trajectory is retargeted from the first patch point after TEI-1, using the TEI-1 burn data and postburn state from the previous example. All terminal and path constraints are the same. Burn data

Table 1 Initial conditions

	Initial value
Epoch	4 Apr. 2024 15:30:00 TDT
Mass, kg	20339.9 (total fuel = 8063.65 kg)
x, km	−1236.7970783385588
y, km	1268.1142350088496
z, km	468.38317094160635
$\dot{x}$, km/s	0.0329108058365355
$\dot{y}$, km/s	0.589269803607714
$\dot{z}$, km/s	−1.528058717568413

Table 2 Engine parameters

Engine	Thrust, N	I_{sp}, s
Main	33,361.6621	326
Auxiliary	4,448.0	309

Table 3 Terminal constraints

Constraint	Value
Geodetic altitude, km	121.92
Longitude, deg	175.6365
Geocentric azimuth, deg	49.3291
Geocentric flight path angle, deg	−5.86

Table 4 Constraint tolerance values

Constraint	Tolerance
Geodetic altitude, km	1e − 4
Longitude, deg	1e − 4
Geocentric azimuth, deg	1e − 4
Geocentric flight path angle, deg	1e − 4

Table 5 Scale values

	Value
Position, km	1
Time, s	3600
$\Delta\mathbf{v}$, km/s	0.001
Altitude, km	1000
Longitude, rad	π
Azimuth, rad	π
Flight path angle, rad	1

Table 6 Maneuver and convergence data

	Impulsive $\Delta\mathbf{v}$, km/s	Unscaled FB $\Delta\mathbf{v}$, km/s	Scaled FB $\Delta\mathbf{v}$, km/s
TEI-1	0.6619	0.6900	0.6827
TEI-2	0.3257	0.2598	0.2576
TEI-3	0.4115	0.4133	0.4185
Total	1.3991	1.3630	1.3589
Iterations	20	7	6

Table 7 Burn data: unscaled algorithm

Maneuver	Duration, s	Prop. mass consumed, kg
TEI-1	381.0984	3975.542
TEI-2	123.4850	1288.171
TEI-3	176.8428	1844.789
Total	681.4262	7108.502

Table 8 Burn data: scaled algorithm

Maneuver	Duration, s	Prop. mass consumed, kg
TEI-1	377.5064	3914.0619
TEI-2	122.7568	1272.7671
TEI-3	179.4875	1860.9622
Total	679.7507	7047.7912

Table 9 Burn data using auxiliary engines: unscaled algorithm

Maneuver	Duration, s	Prop. mass consumed, kg	$\Delta\mathbf{v}$, km/s
TEI-1	381.0984	3975.542	0.6900
TEI-2	917.1483	1347.401	0.2602
TEI-3	1408.5339	2069.305	0.4489
Total	2706.7806	7392.248	1.3991

Table 10 Burn data using auxiliary engines: scaled algorithm

Maneuver	Duration, s	Prop. mass consumed, kg	$\Delta\mathbf{v}$, km/s
TEI-1	377.5064	3914.062	0.6827
TEI-2	939.5373	1380.293	0.2657
TEI-3	1344.6461	1975.446	0.4262
Total	2645.3467	7269.801	1.3746

Table 11 Constraint error data with auxiliary engines

	Unscaled	Scaled
Iterations	5	5

Table 12 Optimal burn data using auxiliary engines

Maneuver	Duration, s	Prop. mass consumed, kg	$\Delta\mathbf{v}$, km/s
TEI-1	3327.008	3501.37	0.6040
TEI-2	873.040	1504.85	0.2993
TEI-3	1312.285	1828.22	0.4059
Total	5512.333	6834.44	1.3092

for each maneuver are listed in Tables 9 and 10, and the convergence data are given in Table 11.

The finite burn algorithm is able to meet the entry and cost constraints in the same number of iterations as in the preceding example, where all three maneuvers are performed by the main engine. Interestingly, the total $\Delta\mathbf{v}$ for the scaled algorithm, in this case, is still lower than the impulsive solution. This will not always be the case; numerous feasible solutions can exist for any given set of patch points. Consider the results from the finite burn algorithm with and without scaling implemented: although the same algorithm is used, simply scaling the variables causes the targeter to converge on a different solution with a different total cost. In this particular case, the finite burn level II process identified a lower cost solution than that determined with the impulsive targeter. In both cases, the total cost constraint is always enforced to ensure that the total cost is within the available fuel budget.

3. *Optimized Finite Burn Trajectory*

As a final step in this analysis, the trajectory generated from the finite burn targeter with auxiliary thrusters was optimized using the Copernicus trajectory optimization system [22]. The results of this optimized run are available in Table 12. The total $\Delta\mathbf{v}$ for the optimal run is $1.3092\ \mathrm{km/s}$. This is an improvement of approximately $0.055\ \mathrm{km/s}$ of $\Delta\mathbf{v}$ over the finite burn targeting solution. However, the cost constraint imposed during the level II process specified the total cost should not exceed $1.40\ \mathrm{km/s}$. Specifying a lower boundary on this constraint may have identified a similar solution. It is always important to bear in mind that a targeter does not seek optimal solutions, only feasible solutions. If a feasible solution exists in the vicinity of the initial guess, either the impulsive or the finite burn targeting algorithm can typically identify it.

4. *Finite Burn Example over the Lunar Cycle*

To further demonstrate the capabilities of the finite burn algorithm, return trajectories were generated over several days spanning the lunar cycle from 1–28 February 2024, 0:00:00 Terrestrial Dynamical Time, again using the auxiliary engines for the second and third TEI maneuvers. In this case, however, it is assumed that the use of the auxiliary engines is intentional and not the result of an unexpected main-engine failure. Thus, all three burn maneuvers are targeted simultaneously by the algorithm. Running different cases over the entire lunar cycle allows for testing of the algorithm under varying Earth-moon configurations. For these runs, only two entry constraints are targeted: geodetic altitude and geocentric flight path angle. The targeted values for these constraints are the same as those given in Table 3.

In this simulation, the input patch points to the finite burn algorithm come from a converged impulsive trajectory with the same initial point and entry targets. The total $\Delta\mathbf{v}$ of the impulsive trajectory for each case is $1.50\ \mathrm{km/s}$. Tables 13 and 14 list results both without and with scaling, respectively, for days 1, 3, 6, 10, 13, 16, 19, 22, 25, and 28 of the lunar cycle.

In the preceding two examples, the impact of variable scaling is negligible. Here, however, several cases over the lunar cycle show a marked disparity in convergence when scaling is applied. Most notably, the number of iterations required to converge for day 25 jumps from five to 20 when scaling is introduced. Conversely, applying scaling to the case with the worst convergence unscaled, day 16, reduces the number of iterations from nine to five and decreases the total $\Delta\mathbf{v}$ noticeably. As was stated already, there are multiple feasible solutions for any given set of patch points; implementing variable scaling allows the algorithm to explore a different part of the solution space than it would otherwise, resulting in a different converged solution. Furthermore, some sets of scale values will produce better results for a given problem than others. The values used in these examples are chosen arbitrarily, and so, performance of the scaled algorithm could potentially be improved if the scale values are determined in a more optimal manner. Several different scaling methods can be found in the literature [23]; however, such investigation is beyond the scope of the present study. Although scaling clearly does not always provide a better solution

Table 13 Unscaled burn data over the lunar cycle

Day	TEI-1		TEI-2		TEI-3		Total cost	
	$\Delta\mathbf{v}$, km/s	Duration, s	$\Delta\mathbf{v}$, km/s	Duration, s	$\Delta\mathbf{v}$, km/s	Duration, s	Total $\Delta\mathbf{v}$, km/s	Iterations
1	0.6238	348.0345	0.4743	1651.2834	0.4019	1210.4498	1.5000	3
3	0.6992	385.6885	0.4402	1505.0542	0.3606	1079.8977	1.5000	5
6	0.6041	338.0300	0.5248	1823.7242	0.3709	1111.0038	1.4998	7
10	0.6045	338.2269	0.5781	1991.5292	0.3175	942.3371	1.5000	5
13	0.6168	344.5208	0.3691	1310.0636	0.5141	1577.7328	1.5000	4
16	0.6580	365.1850	0.3286	1158.8930	0.5135	1576.9396	1.5000	9
19	0.6014	336.6757	0.5866	2020.2097	0.3116	924.1053	1.4997	6
22	0.6978	384.9900	0.3844	1326.8775	0.4072	1233.2389	1.4894	7
25	0.6041	338.0202	0.6310	2155.7213	0.2650	779.7340	1.5000	5
28	0.6174	344.7791	0.4917	1710.5830	0.3910	1175.1849	1.5000	4

Table 14 Scaled burn data over the lunar cycle

Day	TEI-1		TEI-2		TEI-3		Total cost	
	$\Delta\mathbf{v}$, km/s	Duration, s	$\Delta\mathbf{v}$, km/s	Duration, s	$\Delta\mathbf{v}$, km/s	Duration, s	Total $\Delta\mathbf{v}$, km/s	Iterations
1	0.6320	352.1669	0.4740	1646.1692	0.3939	1185.0331	1.4999	6
3	0.7159	393.8848	0.4303	1465.9122	0.3538	1058.6550	1.5000	5
6	0.6122	342.1556	0.5329	1844.8053	0.3548	1059.8574	1.4999	11
10	0.6045	338.2353	0.5765	1986.4955	0.3190	947.2553	1.5000	11
13	0.6154	343.7799	0.3698	1312.9152	0.5148	1580.1157	1.5000	6
16	0.7064	389.2130	0.3342	1159.8967	0.3965	1219.9719	1.4371	5
19	0.6017	336.8295	0.5992	2059.0593	0.2991	885.0120	1.5000	12
22	0.7172	394.5247	0.3867	1326.0707	0.3962	1194.1109	1.5001	9
25	0.6119	342.0196	0.6402	2178.5320	0.2480	727.6002	1.5001	20
28	0.6207	346.4521	0.4884	1698.1462	0.3910	1175.4618	1.5001	7

than the one found by the unscaled algorithm, it adds a measure of flexibility to the finite burn algorithm. This is particularly useful for cases that will not converge when the unscaled targeter is applied, as shown in the next section.

5. *Delayed Patch Point Simulations*

Another test of the finite burn algorithm is whether or not it can converge on a feasible solution given a set of patch points that are not current. A set of patch points corresponding to a current or future departure time may not always be available, especially when ground communications are lost. The algorithm must, therefore, be able to converge even when the departure time listed in the input file has already passed. For this example, the input patch point file from the 1 February run in the preceding section is used. However, the initial epoch is shifted so that it no longer matches that of the startup arc. The patch-point times are updated to reflect this time shift, but the corresponding positions and velocities are not. This results in an initial guess trajectory with significant errors in the terminal constraint conditions; the algorithm must retarget the direction, duration, and ignition times of all three burns to correct these errors. As before, the auxiliary engines perform the TEI-2 and TEI-3 maneuvers, and all three maneuvers are targeted simultaneously. To ensure that the characteristics of the initial lunar orbit remain the same, the patch points are converted to the MCI frame before the time delay is introduced. The initial time of the simulation is perturbed for 3, 6, 9, and then 12 h. The initial entry constraint errors due to the time delay are given in Table 15, and the convergence data for the scaled and unscaled algorithm are listed in Table 16.

The benefits of variable scaling are most apparent in this example. Without scaling, the finite burn algorithm is only able to find a feasible solution for a 3 h delay and no more. After the scaling is implemented, however, the targeter is able to find a feasible solution that satisfies the entry constraints and fuel budget for even a 12 h delay. As mentioned in the preceding example, it is possible that the scaled algorithm performance could be even further improved with a more optimal set of scale values. These results underscore the adaptability of the two-level targeting structure; instead of trying to match a previously determined nominal trajectory, the algorithm explores the nearby solution space and is able to converge on a trajectory despite the poor quality of the initial input.

B. Example 2: Mars Encounter Simulation and Results

In this section, the two-level targeting algorithm is applied to a simulated Mars encounter. The simulation begins in a low-Earth parking orbit and is followed by the trans-Mars injection (TMI) burn. The vehicle then coasts for the remainder of the simulation until Mars arrival, approximately 258 days later. Patch points are placed along the trajectory at the following locations: initial conditions, TMI, interior patch points (one point every two months), and Mars encounter. The constraints in this simulation are: 1) velocity continuity at the interior patch points, 2) terminal altitude, and 3) terminal flight path angle.

The initial conditions in the low-Earth parking orbit are given in Table 17. The state components are expressed in the J2000 Earth-centered inertial frame. The engine parameters are given in Table 18. The constraints and their tolerances are given in Table 19. Scale values are given in Table 20. Finally, TMI impulsive and finite burn data and the number of iterations required for convergence are given in Table 21. Note that, for the Mars simulation, only scaled solutions were computed. Although similar in nature to the preceding precision entry example, this simulation demonstrates the applicability of the finite burn two-level targeter to a broader class of missions that are different in scope and outside of the Earth-moon system. In general, the methodology is applicable to any path-

Table 15 Initial entry constraint errors

Delay, hr	Altitude error, km	Flight-path-angle error, deg
3	11040.163	1.6526
6	22415.148	4.3289
9	33857.773	7.3130
12	45332.406	10.3842

Table 16 Convergence with delays

Delay, hr	Unscaled	Scaled
3	10	17
6	Did not converge	23
9	Did not converge	29
12	Did not converge	35

Table 17 Mars initial conditions

Epoch	23 Nov. 1994 07:28:06.184 TDT
Mass, kg	136642.391
x, km	−329.168
y, km	5758.377
z, km	3131.700
$\dot{x}$, km/s	−7.7836
$\dot{y}$, km/s	−0.4398
$\dot{z}$, km/s	−0.0046

Table 18 Engine parameters

Engine	Main
Thrust, N	1,309,779
I_{sp}, s	465

Table 19 Mars constraints and tolerances

Constraint	Value	Tolerance
$\Delta\mathbf{v}^{+} - \Delta\mathbf{v}^{-}$, km/s	0	0.001
Terminal altitude, km	300	0.001
Terminal flight path angle, deg	0	1

Table 20 Mars scale values

Position, km	1
Time, s	3600
$\Delta\mathbf{v}$, km/s	0.001
Altitude, km	1000
Flight path angle, deg	3

planning or guidance problem that seeks feasible arcs in the presence of interior path, terminal, and/or cost constraints.

Given the very high main engine thrust in Table 18, the TMI finite burn duration in Table 21 is relatively short (~4.5 min); thus, the scaled finite burn solution is not vastly different from the impulsive solution. In other words, the example as is does not stress the convergence of the finite burn algorithm.

To stress the algorithm, the main engine thrust and flow rate are now reduced (in proportion) until the impulsive solution no longer provides a valid initial guess for the finite burn algorithm. As the thrust and flow rate are reduced, the finite burn duration increases. Because the thrust direction is inertially held throughout the burn, increasing the finite burn duration also increases the radial component of the finite burn $\Delta\mathbf{v}$ vector. At some point, the finite burn does not provide enough energy to escape from Earth and leaves the vehicle in a highly eccentric Earth orbit. At this thrust and flow rate, the impulsive solution no longer provides a valid initial guess, and an alternate approach must be used to seed the finite burn algorithm. Investigating this alternate approach is beyond the scope of this paper. Table 22 compares the original TMI burn data in Table 21 with the burn data for the minimum thrust and flow rate case. Note that thrust and flow rate were successfully reduced by 57% from the original case to the minimum case before failure. Also note, however, the simultaneous increase in all other burn quantities ($\Delta\mathbf{v}$, propellant mass consumed, burn duration, etc.).

Finally, an attempt is made to reduce the thrust and flow rate even further by increasing the density of the interior patch points from one point every two months to one point per month. By providing a shorter distance from TMI to the first interior patch point, the algorithm should be able to tolerate a greater divergence of the finite burn solution from the impulsive solution (i.e., the valid region of the impulsive solution as an initial guess for the finite burn solution is larger). Table 23 compares the TMI burn data for the original and minimum thrust/flow rate cases for a one-month intermediate patch point density. For the one-month density, thrust and flow rate were successfully reduced by 61% from the original case to the minimum case before failure. Note, again, however, the simultaneous increase in all other burn quantities. As expected, the increases in the one-month density case are greater than in the two-month density case given the larger reduction in thrust and flow rate. Also, as expected, the burn data for the original thrust/flow rate case with the two-month density are close but not identical to the data for the original case with one-month density. This reinforces the message that solutions to a given problem are dependent on the number and placement of patch points in the simulation.

Table 21 TMI burn data

Impulsive $\Delta\mathbf{v}$, km/s	4.0347
Scaled FB $\Delta\mathbf{v}$, km/s	4.2048
Prop. mass consumed, kg	82282.144
Burn duration, s	286.580
Iterations	7

Table 22 TMI stress case with two-month intermediate patch point density

Case	Original T, $\dot{m}$	Minimum T, $\dot{m}$	Change
Thrust, N	1,309,779	561588	57% ↓
Flow rate, kg/s	287.116	123.105	57% ↓
Scaled FB $\Delta\mathbf{v}$, km/s	4.2048	4.9231	17% ↑
Prop. mass consumed, kg	82282.144	90202.333	10% ↑
Burn duration, s	286.580	732.721	156% ↑
FB local vertical/local horizontal (LVLH) frame$\Delta\mathbf{v}_z$, km/s	−0.5530	−1.1760	113% ↑
Imp. $\Delta\mathbf{v}$ to FB $\Delta\mathbf{v}$ angle, deg	7.079	13.347	89% ↑

Table 23 TMI stress case with one-month intermediate patch point density

Case	Original T, $\dot{m}$	Minimum T, $\dot{m}$	Change
Thrust, N	1,309,779	511545	61% ↓
Flow rate, kg/s	287.116	112.128	61% ↓
Scaled FB $\Delta\mathbf{v}$, km/s	4.2038	5.1011	21% ↑
Prop. mass consumed, kg	82270.442	91976.784	12% ↑
Burn duration, s	286.539	820.277	186% ↑
FB LVLH $\Delta\mathbf{v}_z$, km/s	−0.5501	−1.2577	129% ↑
Imp. $\Delta\mathbf{v}$ to FB $\Delta\mathbf{v}$ angle, deg	7.078	13.812	95% ↑

IV. Conclusions

This paper presents the theoretical framework and numerical validation of a finite burn linear targeting process suitable for autonomous path planning and guidance applications. The numerical solution process is as simple as that of an impulsive two-level targeting algorithm, though incorporating finite burn capabilities does increase the complexity of the analytical gradients employed. To accommodate the goals of autonomous flight, the total propulsive cost is constrained according to the fuel budget available at the time the numerical process is initiated. The development of that constraint is presented and demonstrated as well. An initial exploration into performance enhancements is presented through the incorporation of variable scaling. Each of these developments is tested on two examples, a precision entry problem and a rendezvous problem. The increased complexity of analytical gradients naturally translates to an increase in computational overhead, in contrast to the impulsive targeting process. However, unlike the impulsive algorithm, the finite burn targeting process is capable of addressing lower-thrust scenarios, which may arise during main-engine-failure contingencies. This investigation successfully demonstrates that linear targeting algorithms for path planning and guidance need not be limited by impulsive assumptions. Furthermore, the algorithm presented serves as an alternative to nonlinear optimization methods, which may not be feasible for autonomous flight applications.

Acknowledgments

This research was carried out at The University of Texas at Austin and funded under NASA award no. NNX07AR46G. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NASA.

References

[1] Hyde, C. T., Foggat, C. E., and Weber, B. D., "Apollo Experience Report—Abort Planning," NASA, TN-D-6847, Houston, TX, June 1972.

[2] Yencharis, J. D., Wiley, R. F., Davis, R. S., Holmes, Q. A., and Zeiler, K. T., "Apollo Experience Report—Development of Guidance and Targeting Techniques for the Command Module and Launch Vehicle," NASA TN-D-6848, Houston, TX, June 1972.

[3] Ocampo, C. A., and Saudemont, R. R., "Initial Trajectory Model for a Multi-Maneuver Moon to Earth Abort Sequence," *AAS/AIAA Space Flight Mechanics Meeting*, AIAA Paper 2009-195, Savannah, GA, Feb. 2009.

[4] Marchand, B. G., Howell, K. C., and Wilson, R. S., "Improved Corrections Process for Constrained Trajectory Design in the *n*-Body Problem," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007, pp. 884–897.

doi:10.2514/1.27205

[5] Jezewski, D. J., "Primer Vector Theory and Applications," NASA TR-454, NASA Johnson Spaceflight Center, 1975.

[6] Enright, P. J., and Conway, B. A., "Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 981–985.
doi:10.2514/3.20739

[7] Ocampo, C., "Finite Burn Maneuver Modeling for a Generalized Spacecraft Trajectory Design and Optimization System," *Annals of the New York Academy of Sciences*, Vol. 1017, 2004, pp. 210–233.
doi:10.1196/annals.1311.013

[8] Ranieri, C. L., and Ocampo, C. A., "Optimization of Roundtrip, Time-Constrained, Finite Burn Trajectories via an Indirect Method," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 2, 2005, pp. 306–314.
doi:10.2514/1.5540

[9] Wilson, R. S., and Howell, K. C., "Trajectory Design in the Sun-Earth-Moon System Using Lunar Gravity Assists," *Journal of Spacecraft and Rockets*, Vol. 35, No. 2, 1998, pp. 191–198.
doi:10.2514/2.3309

[10] Zimmer, S., and Ocampo, C., "Use of Analytical Gradients to Calculate Optimal Gravity-Assist Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 2, 2005, pp. 324–332.
doi:10.2514/1.4825

[11] Teofilatto, P., and Pasquale, E. D., "A Fast Guidance Algorithm for an Autonomous Navigation System," *Planetary and Space Science*, Vol. 46, No. 11/12, 1998, pp. 1627–1632.
doi:10.1016/S0032-0633(97)00233-X

[12] Miele, A., Ciarcia, M., and Weeks, M. W., "Guidance Trajectories for Spacecraft Rendezvous," *Journal of Optimization Theory and Applications*, Vol. 132, 2007, pp. 377–400.
doi:10.1007/s10957-007-9165-5

[13] Howell, K. C., and Pernicka, H. J., "Numerical Determination of Lissajous Trajectories in the Restricted Three-Body Problem," *Celestial Mechanics*, Vol. 41, 1987, pp. 107–124.
doi:10.1007/BF01238756

[14] Howell, K. C., Barden, B. T., Wilson, R. S., and Lo, M. W., "Trajectory Design Using a Dynamical Systems Approach with Application to GENESIS," *Proceedings of the AAS/AIAA Astrodynamics Conference*, AIAA, Reston, VA, 1997, pp. 1665–1684.

[15] Wilson, R. S., Barden, B. T., Howell, K. C., and Marchand, B. G., "Summer Launch Options for the Genesis Mission," *Advances in the Astronautical Sciences*, Vol. 109, 2002, pp. 77–94.

[16] Weeks, M. W., Marchand, B. G., Smith, C. W., and Scarritt, S. K., "Design of the Onboard Autonomous Targeting Algorithm for the Trans-Earth Phase of Orion," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2008-7262, Honolulu, HI, Aug. 2008.

[17] Smith, C. W., "An Onboard Targeting Algorithm with Earth-Return Applications," Master's Thesis, Univ. of Texas at Austin, Austin, TX, Aug. 2008.

[18] Vinh, N. X., Busemann, A., and Culp, R. D., *Hypersonic and Planetary Flight Mechanics*, Univ. of Michigan Press, Ann Arbor, MI, 1980.

[19] Harpold, J. C., and C. A. Graves, J., "Shuttle Entry Guidance," *Journal of the Astronautical Sciences*, Vol. 27, July–Sept. 1979, pp. 239–268.

[20] Loh, W. H. T., *Re-Entry and Planetary Entry Physics and Technology*, Univ. of Michigan Press, Ann Arbor, MI, 1980.

[21] Penzo, P. A., "An Analysis of Moon-to-Earth Trajectories," NASA CR-132100, Redondo Beach, CA, Oct. 1961.

[22] Ocampo, C. A., "An Architecture for a Generalized Spacecraft Trajectory Design and Optimization System," *Libration Point Orbits and Applications: Proceedings of the Conference*, Institut D'Estuis Espacials de Catalunya, Aiguablava, Spain, June 2002, pp. 529–571.

[23] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, New York/London/Orlando, FL, 1981.