An Autonomous Onboard Targeting Algorithm Using Finite Thrust Maneuvers

Sara K. Scarritt[*], Belinda G. Marchand[†],
Aaron J. Brown[‡], William H. Tracy[§], Michael W. Weeks[¶]

I. Abstract

In earlier investigations, the adaptation and implementation of a modified two-level corrections (or targeting) process as the onboard targeting algorithm for the Trans-Earth Injection phase of Orion is presented. The objective of that targeting algorithm is to generate the times of ignition and magnitudes of the required maneuvers such that the desired state at entry interface is achieved. In an actual onboard flight software implementation, these times of ignition and maneuvers are relayed onto Flight Control for command and execution. Although this process works well when the burn durations or burn arcs are small, this might not be the case during a contingency situation when lower thrust engines are employed to perform the maneuvers. Therefore, a new model for the two-level corrections process is formulated here to accommodate finite burn arcs. This paper presents the development and formulation of the finite burn two-level corrector, used as an onboard targeting algorithm for the Trans-Earth Injection phase of Orion. A performance comparison between the impulsive and finite burn models is also presented. The present formulation ensures all entry constraints are met, without violating the available fuel budget, while allowing for low-thrust scenarios with long burn durations.

II. Introduction

UNDER nominal operational conditions, the Crew Module's (CM) 33,361 N main engine performs the 3-burn Trans-Earth Injection (TEI) sequence. However, the spacecraft must

[*]Graduate Student, Aerospace Engineering, The University of Texas at Austin, 210 E. 24th St., Austin, TX 78712.
[†]Assistant Professor, Aerospace Engineering, The University of Texas at Austin, 210 E. 24th St., Austin, TX 78712.
[‡]Engineer, DM34/Rendezvous Guidance and Procedures, NASA JSC, 2101 NASA Pkwy. Houston, Texas 77058.
[§]Engineer, DM34/Rendezvous Guidance and Procedures, NASA JSC, 2101 NASA Pkwy. Houston, Texas 77058.
[¶]Engineer, EG-6/Aeroscience and Flight Mechanics Division, NASA JSC, 2101 NASA Pkwy. Houston, Texas 77058.

also have the ability to autonomously target and execute the maneuvers to return the crew safely to Earth using the backup auxiliary engines. Since the total thrust of these eight (8) auxiliary engines is 4,448 N, the burn durations are naturally much longer. For instance, the baseline TEI-1 maneuver, implemented by the main engine, lasts roughly 5.5 minutes. In contrast, if TEI-1 is executed using solely the auxiliary engines, the duration increases to 55 minutes. Due to these extended burn durations, it is no longer accurate to approximate each maneuver as impulsive. Thus, it is necessary to develop an autonomous targeting algorithm that allows maneuvers of finite duration during a main engine failure scenario.

Autonomy, for the Orion trans-Earth phase, refers to the ability to (a) automatically identify a suitable startup arc[1–3] and (b) 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 discrepancices in the timing and state. More recent methods[3] consider the use of infeasible solutions (i.e. 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, such as the Orion TEI sequence, require an increased number of samples, roughly one every 12 hours. 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 (i.e.fuel available). The solutions supplied are also based on the availability of the main engine. As such, in a main engine failure scenario, the quality of the initial guess supplied degrades significantly. The present study is strictly focused on the second stage of the autonomous targeting process, re-targeting 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 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 algorithm presented here is a modified two-level corrector,[4,9,13–16] employed during the design of the Genesis trajectory,[9,14,15] that 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. 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 correction process also allows for straightforward addition of path constraints, both those at specific points (i.e. 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. For all the comparisons, the same startup arc (based on impulsive maneuvers) and the same entry targets are used.

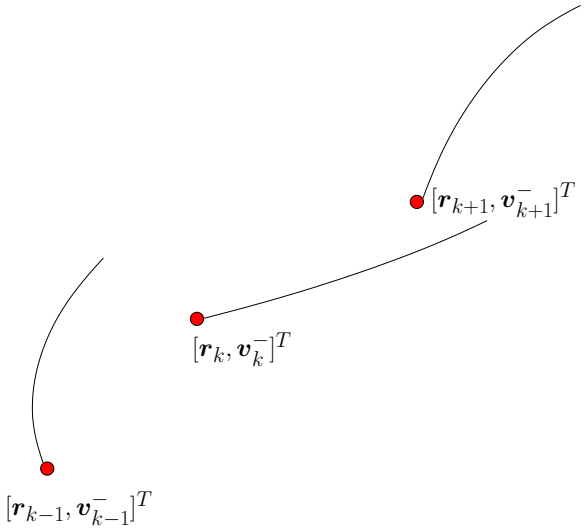
III. Finite Burn Targeting Algorithm

The basic structure of the Level II process in a two-level targeter that incorporates finite burns is the same as that of the impulsive two-level targeter.[4] First, both algorithms treat the position and times of each patch point as control variables. Furthermore, both algorithms employ a minimum norm solution in computing the updates to these 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. Due to the interdependency between these state variables, the partial derivatives are also more complex in nature than those of the impulsive targeter.

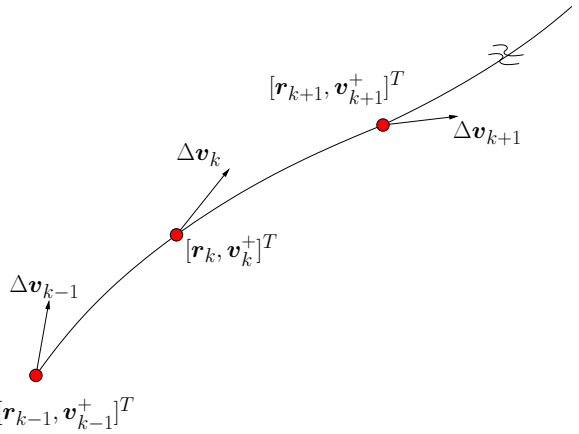
Traditionally, an impulsive two-level targeter requires a startup arc represented by a series of "patch states." These states, also termed "patch points," are selected by the user as representative waypoints along the trajectory. The user supplies the time and state at each patch point, t_k and $\mathbf{x}_k^+ = [\ \mathbf{r}_k \quad \mathbf{v}_k^+\ ]$ for $k = 1, \cdots, N$, respectively. Each state $\mathbf{x}_k^+$ is then numerically integrated forward over an interval $[t_k, t_{k+1}]$, for $k = 1, \cdots, N-1$. The integrated state, at time t_{k+1}, is recorded as $\mathbf{x}_{k+1}^-$. This is to allow for the possibility that the user supplied velocity at that point, $\mathbf{v}_{k+1}^+$, may not coincide with that identified during the propagation, $\mathbf{v}_{k+1}^-$. 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 Figure 1(a)-1(b). 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.

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 meet any additional user specified constraints. This is graphically illustrated in Figure 2(a)-2(c). Figure 2(a) is representative of the scenario in Figure 1(b). Figure 2(b) illustrates how the patch state positions, and potentially the associated times, have been adjusted by the Level II process. Subsequently, since 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 sequentially applied once more 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 con-

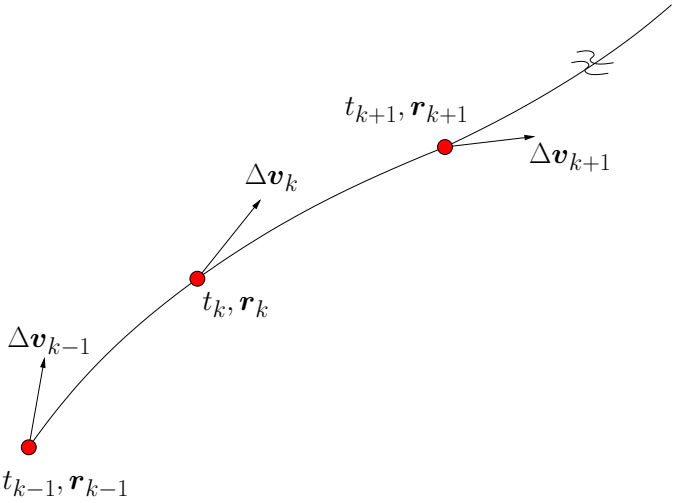


(a) Impulsive Two-Level Targeter: Before Level I Process is Applied

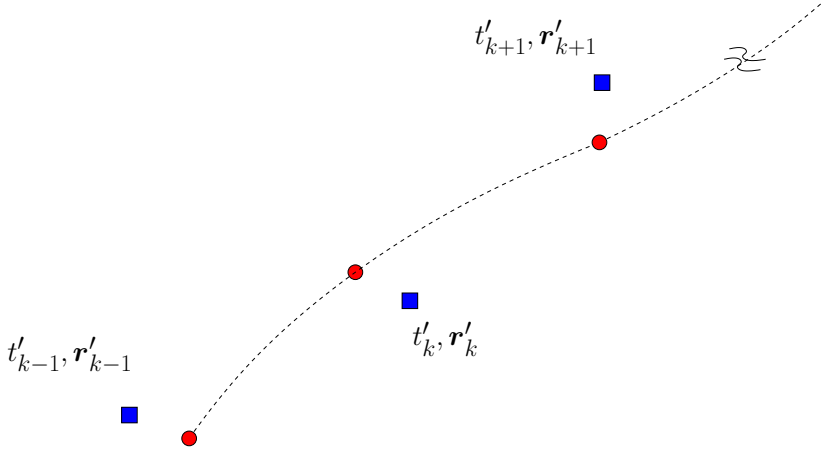


(b) Impulsive Two-Level Targeter: After Level I Process is Applied

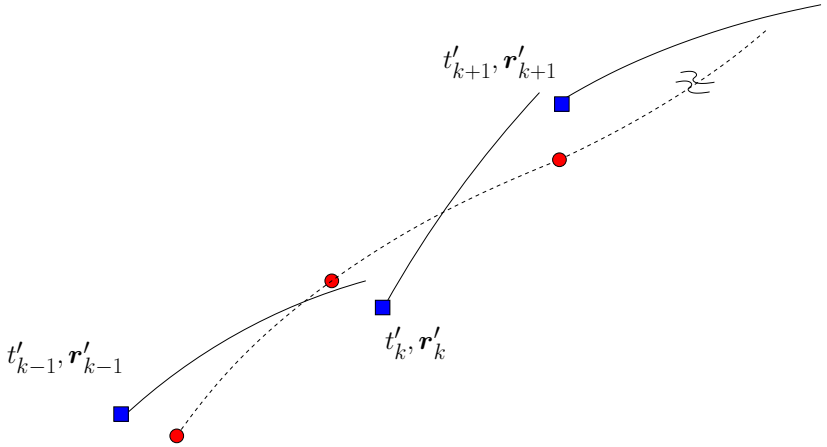
Figure 1. Level I Process



(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

Figure 2. Level II Process

verge. However, since 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 non-convergence. 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 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 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 employed Δv's at the start of each segment to achieve position continuity. These Δv'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 equivalent of a two-body Lambert targeter, except the time of flight is not necessarily fixed or pre-specified. This ultimately reduces to some form of linear differential correction where Δv'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 Figure 3. In a Level I process that employs finite burns rather than impulsive maneuvers, the burn arc is considered to be a subsegment of the arc between patch points $k-1$ and k. The end of

the burn occurs at point T. In identifying finite burn arcs, it is necessary to consider an

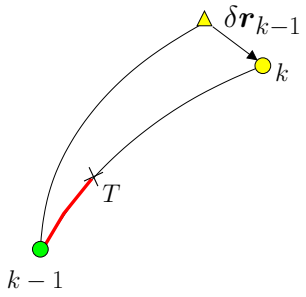


Figure 3. Level 1 Process

augmented state vector $\begin{bmatrix} \mathbf{r}_k & \mathbf{v}_k & m_k & \dot{m}_{g_k} & \mathbf{u}_k \end{bmatrix}^T$, where m_k, $\dot{m}_{g_k}$, and $\mathbf{u}_k$ represent the spacecraft mass, the propellant flow rate, and the thrust direction associated with patch point k, respectively. 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 thrust direction ($\mathbf{u}_{k-1} = \mathbf{u}_k$) and the time at the end of the burn (t_T). The variational equation for the burn subsegment is,

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

for the burn subsegment, 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 sub-matrices corresponding to each state:

$$\Phi(T, k-1) = \begin{bmatrix} A_{T,k-1} & B_{T,k-1} & E_{T,k-1} & F_{T,k-1} & G_{T,k-1} \\ C_{T,k-1} & D_{T,k-1} & H_{T,k-1} & I_{T,k-1} & J_{T,k-1} \\ K_{T,k-1} & L_{T,k-1} & M_{T,k-1} & N_{T,k-1} & O_{T,k-1} \\ P_{T,k-1} & Q_{T,k-1} & R_{T,k-1} & S_{T,k-1} & T_{T,k-1} \\ U_{T,k-1} & V_{T,k-1} & W_{T,k-1} & X_{T,k-1} & Y_{T,k-1} \end{bmatrix} \tag{2}$$

For the subsequent coasting subsegment, the variational equation, with partitioned state

transition matrix, takes the same form as in the impulsive formulation,[4]

$$\begin{bmatrix} \delta\mathbf{r}_k - \mathbf{v}_k^- \delta t_k \\ \delta\mathbf{v}_k^- - \mathbf{a}_k^- \delta t_k \end{bmatrix} = \begin{bmatrix} A_{k,T} & B_{k,T} \\ C_{k,T} & D_{k,T} \end{bmatrix} \begin{bmatrix} \delta\mathbf{r}_T - \mathbf{v}_T^+ \delta t_T \\ \delta\mathbf{v}_T^+ - \mathbf{a}_T^+ \delta t_T \end{bmatrix} \tag{3}$$

For this formulation, both the initial and final times of the arc (t_{k-1} and t_k) 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. The mass flow rate, $\dot{m}_{g_{k-1}}$, is also fixed. 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 Equation (1) and Equation (3) can be combined to give an expression for $\delta\mathbf{r}_k$:

$$\delta\mathbf{r}_k = \begin{bmatrix} (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+) \end{bmatrix} \begin{bmatrix} \delta\mathbf{u}_{k-1}^+ \\ \delta t_T \end{bmatrix} \tag{4}$$

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

$$\begin{bmatrix} \delta\mathbf{u}_{k-1}^+ \\ \delta t_T \end{bmatrix} = \tilde{M}^T(\tilde{M}\tilde{M}^T)^{-1}\delta\mathbf{r}_k, \tag{5}$$

where

$$\tilde{M} = \begin{bmatrix} (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & B_{k,T}(\mathbf{a}_T^- - \mathbf{a}_T^+) \end{bmatrix}. \tag{6}$$

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.

To determine an initial guess for the finite burn (i.e. thrust direction, thrust magnitude, and burn time), first the impulsive Level I process is used to compute an impulsive correction. The impulsive $\Delta\mathbf{v}$ direction is used as an initial guess for the thrust direction. The desired thrust value (i.e. the thrust of the engine) is used as an initial guess for the thrust magnitude. Finally, the burn time is deduced by starting with the rocket equation,

$$\Delta \mathrm{v}_k = -I_{sp}g_0 ln(1 - \frac{\dot{m}_{g_k}\Delta t_{burn}}{m_k}), \tag{7}$$

substituting in $\dot{m}_{g_k} = u_k / I_{sp} g_0$, where u_k is the thrust magnitude, and rearranging to obtain

$$\Delta t_{burn} = \frac{m_k}{\dot{m}_{g_k}} \left(1 - e^{\frac{-\Delta v_k \dot{m}_{g_k}}{u_k}} \right) \tag{8}$$

It is important to note that because this initial guess is based on an impulsive maneuver, 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

The $\delta \mathbf{u}_{k-1}^{+}$ in Equation 5 implies changes in both the direction *and* magnitude of the thrust vector, $\mathbf{u}_k$. For a constant thrust engine, however, a change in the thrust magnitude is clearly not desirable. Fortunately, there are several workarounds for controlling the thrust magnitude to the desired value.

The first approach is to simply ignore the change in thrust magnitude suggested by $\delta \mathbf{u}_{k-1}^{+}$ and only use the change in thrust direction. The new thrust direction along with desired thrust magnitude are used to create a new thrust vector in the equations of motion. The Level I process converges using this technique, but convergence is slower since the updates to thrust vector in the equations of motion do not match the updates suggested by the minimum norm solution.

The second approach for controlling the thrust magnitude is to use a **thrust biasing** technique. This technique is similar to final position biasing commonly used in perturbed Lambert targeting.insert FDO Console Handbook citation. The technique uses the same Level I process, but implements the full thrust vector update (direction *and* magnitude) suggested by the minimum norm solution in the equations of motion.[a] At convergence, $\delta \mathbf{r}_k = \mathbf{0}$, but the converged thrust magnitude will different from the desired value. Let $\Delta u_k = u_{desired} - u_{converged}$. 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 $u_{initial\,(new)} = u_{initial\,(old)} + \Delta u_k$. The initial guess for the burn time is also updated using Equation 8 to reflect this change in thrust magnitude. At convergence, the thrust magnitude will again be different from, but now much closer to the desired value. This process of biasing and re-converging is repeated (typically 3 to 5 iterations) until $\Delta u_k = 0$.

Figure 4 provides example output from a Level I process demonstrating the thrust bi-

[a] Since $\dot{m}_{g_k}$ is assumed constant, in effect this creates a fictitious variable I_{sp} engine since $\dot{m}_{g_k} = u_k / I_{sp} g_0$ but u_k is changing after each iteration. This is only temporary, however, since the thrust biasing technique ultimately brings the thrust magnitude back to the desired value, and hence I_{sp} ultimately returns to its assumed value as well.

asing technique. This particular example is targeting the Trans-Earth Injection 1 (TEI-1) burn as part of the Earth entry targeting process. In Figure 4, the first four lines show the convergence of the impulsive corrector to provide the initial guess for the thrust direction. The next six lines demonstrate the first pass through the Level I process. Note how the initial guess of the thrust magnitude (UMAG) is the desired value of 33361 N. At convergence, however, the thrust magnitude is 32802 N - 559 N lower than the desired value. For the second pass, the initial guess of the thrust magnitude is biased upward to 33920 N, and at convergence results in a value of 33355 N - now only 6 N lower than the desired value. The third and final pass through the Level I process converges back to the desired value of 33361 N.

```
iter =   0 : error =         2793469 : V1X =      5.08212 : V1Y =    925.60483 : V1Z =  -2034.54174 : TOF =        32339
iter =   1 : error =          472316 : V1X =   -121.22931 : V1Y =   1005.30872 : V1Z =  -2000.78751 : TOF =        32339 :
iter =   2 : error =            6340 : V1X =   -119.58171 : V1Y =   1017.33932 : V1Z =  -1991.16948 : TOF =        32339 :
iter =   3 : error =              83 : V1X =   -119.05137 : V1Y =   1018.84897 : V1Z =  -1990.42900 : TOF =        32339 :

pos_iter =   0 : error =    750377.234 : u[0] =  -.19656 : u[1] =   .55101 : u[2] =  -.81102 : UMAG =  33361.665 :  TGO = 223.813
pos_iter =   1 : error =    228809.996 : u[0] =  -.14281 : u[1] =   .50839 : u[2] =  -.84921 : UMAG =  32680.589 :  TGO = 223.145
pos_iter =   2 : error =     59932.769 : u[0] =  -.14719 : u[1] =   .51334 : u[2] =  -.84547 : UMAG =  32830.713 :  TGO = 223.262
pos_iter =   3 : error =     14019.855 : u[0] =  -.14683 : u[1] =   .51272 : u[2] =  -.84591 : UMAG =  32796.740 :  TGO = 223.237
pos_iter =   4 : error =      3098.659 : u[0] =  -.14688 : u[1] =   .51279 : u[2] =  -.84585 : UMAG =  32804.226 :  TGO = 223.242
pos_iter =   5 : error =       666.222 : u[0] =  -.14687 : u[1] =   .51278 : u[2] =  -.84586 : UMAG =  32802.619 :  TGO = 223.241

thrust_iter = 0 : thrust_error = 559.046

pos_iter =   0 : error =    741538.873 : u[0] =  -.19656 : u[1] =   .55101 : u[2] =  -.81102 : UMAG =  33920.711 :  TGO = 220.463
pos_iter =   1 : error =    225354.296 : u[0] =  -.14381 : u[1] =   .50907 : u[2] =  -.84863 : UMAG =  33232.782 :  TGO = 219.815
pos_iter =   2 : error =     58340.643 : u[0] =  -.14805 : u[1] =   .51384 : u[2] =  -.84502 : UMAG =  33382.727 :  TGO = 219.927
pos_iter =   3 : error =     13494.151 : u[0] =  -.14771 : u[1] =   .51326 : u[2] =  -.84543 : UMAG =  33349.211 :  TGO = 219.903
pos_iter =   4 : error =      2969.219 : u[0] =  -.14775 : u[1] =   .51332 : u[2] =  -.84538 : UMAG =  33356.518 :  TGO = 219.908
pos_iter =   5 : error =       640.852 : u[0] =  -.14775 : u[1] =   .51331 : u[2] =  -.84539 : UMAG =  33354.956 :  TGO = 219.907

thrust_iter = 1 : thrust_error = 6.709

pos_iter =   0 : error =    741435.196 : u[0] =  -.19656 : u[1] =   .55101 : u[2] =  -.81102 : UMAG =  33927.420 :  TGO = 220.423
pos_iter =   1 : error =    225303.659 : u[0] =  -.14383 : u[1] =   .50907 : u[2] =  -.84862 : UMAG =  33239.409 :  TGO = 219.775
pos_iter =   2 : error =     58310.984 : u[0] =  -.14806 : u[1] =   .51384 : u[2] =  -.84501 : UMAG =  33389.347 :  TGO = 219.887
pos_iter =   3 : error =     13477.757 : u[0] =  -.14772 : u[1] =   .51326 : u[2] =  -.84542 : UMAG =  33355.842 :  TGO = 219.864
pos_iter =   4 : error =      2953.596 : u[0] =  -.14776 : u[1] =   .51333 : u[2] =  -.84538 : UMAG =  33363.142 :  TGO = 219.869
pos_iter =   5 : error =       639.534 : u[0] =  -.14776 : u[1] =   .51332 : u[2] =  -.84538 : UMAG =  33361.587 :  TGO = 219.868

thrust_iter = 2 : thrust_error = .078
```

Figure 4. Thrust Biasing in the Level I Process

A third approach is to add the thrust magnitude as a constraint in Equation 4. Let

$$\alpha_k = u_k = \sqrt{\mathbf{u}_k{}^T \mathbf{u}_k}. \tag{9}$$

Then

$$\delta\alpha_k = u_k - u_k{}^* \tag{10}$$

and

$$\frac{\partial \alpha_k}{\partial \mathbf{u}_{k-1}^+} = \hat{\mathbf{u}}_k^T, \quad \frac{\partial \alpha_k}{\partial t_T} = 0. \tag{11}$$

Equation 4 then becomes

$$\begin{bmatrix} \delta \mathbf{r}_k \\ \delta \alpha_k \end{bmatrix} = \begin{bmatrix} (A_{k,T} G_{T,k-1} + B_{k,T} J_{T,k-1}) & B_{k,T} \left(\mathbf{a}_T^- - \mathbf{a}_T^+ \right) \\ \hat{\mathbf{u}}_k^T & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{u}_{k-1}^+ \\ \delta t_T \end{bmatrix}. \tag{12}$$

During implementation, however, this method lead to divergence in $\delta \mathbf{r}_k$. The minimum norm solution in this case corrected the thrust magnitude (no matter how large the error) fully in one iteration to the desired value and did not deviate in subsequent iterations. This behavior was not surprising given that the partial of the constraint with respect to the thrust vector is the thrust unit vector itself. Once the thrust magnitude became "fixed," it may not have been possible for the targeter to correct final position discontinuities with changes in the burn time and/or direction alone.

A fourth and final approach for controlling the thrust magnitude is a hybrid of the second and third methods. From the second method, the Level I process is run once to convergence using the full thrust vector update from the minimum norm solution. A thrust bias is determined and the initial guess for the thrust magnitude is updated. At this point however, the thrust magnitude constraint from the third method is imposed prior to repeating the Level I process. Running Level I once *prior* to imposing the constraint improves the initial guess and should aid convergence. This approach has not yet been implemented and is left for future work. If successful, it would require only 2 iterations of the Level I process rather than 3 (which is typically the best case scenario) for the thrust biasing technique.

2. Variable Scaling

A well-known tool for aiding (and sometimes enabling) the convergence of iterative processes such as non-linear targeting is the scaling of both the control variables and constraints to O(1). This enables the targeter to evenly adjust control variables and meet constraints of varying orders of magnitude. In a physical sense, this is analogous to a person artificially adjusting masses and sizes to make juggling a bowling ball, a peanut, and an inflated ballon feel more like juggling a tennis ball, a baseball, and a racquet ball.

To implement scaling in the Level I Process, first note that Equation 4 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

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

Prior to forming the minimum norm solution, the controls, constraints, and partial derivatives are scaled by suitable factors so that

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

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

and therefore,

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

The scaled controls are then updated using the minimum norm solution. To begin the next iteration, the controls are un-scaled and then used in the equations of motion to determine the new constraint variations and partial derivatives. As before, these are re-scaled, a new minimum norm solution is found, and the process is repeated until convergence.

Since the constraints are always being driven to zero in Level I, it is sufficient to set $r_{scl} = 1$. u_{scl} can nominally be set to the desired thrust value. t_{scl} can be set to the initial estimate for Δt_{burn} from Equation 8 if the thrust is sufficiently high. If the thrust is sufficiently low such that the impulsive approximation is not valid (at least for the initial guess), then the burn duration will be much longer and t_{scl} should be increased appropriately.

Finally, note that scaling can and should also be implemented in the Level II Process. In Level II, for the velocity continuity constraint, the controls are $\mathbf{r}_{k-1}$, t_{k-1}, $\mathbf{r}_k$, t_k, $\mathbf{r}_{k+1}$, and t_{k+1}, the constraint is $\Delta\mathbf{v}_k$, and the partial derivatives are given in Equation 18. Note that in Level I, $\mathbf{r}_k$ was a constraint, and therefore it was sufficient to set $r_{scl} = 1$. In Level II, however, $\mathbf{r}_{k-1}$, $\mathbf{r}_k$, and $\mathbf{r}_{k+1}$ are now controls, and must be scaled according to their expected magnitudes at convergence. Similarly, in Level I t_T was a burn time, and therefore t_{scl} was set to the initial estimate for Δt_{burn}. In Level II, however, t_{k-1}, t_k, and t_{k+1} are all patch point epochs, and should be scaled according to their expected values at convergence. This same approach is also taken for the maneuver sum constraint.

3. Level 1 Earth Entry Targeting

The following example shows a comparison of solutions obtained from a Level I process based on the impulsive and finite burn targeters. In this simulation, the Level 1 algorithm is implemented for a trajectory segment from a given TEI-3 position to Earth entry. In keeping with the overall structure of the two-level corrector, only the terminal position vector is targeted for this example. TEI-3 occurs on April 6, 2024 10:07:23.8635 UTC, and the desired entry state is as follows:

- Geodetic Altitude (km) 121.92
- Longitude (deg) 175.6365
- Geocentric Azimuth (deg) 49.3291
- Geocentric Flight Path Angle (deg) -5.86

Because altitude is a function of position only, the error in meeting the desired altitude relates directly to the ability of the Level I process to meet the desired final position. Likewise, with the final time fixed, the final longitude error can also be judged by the final position error. Because the main engines are employed for this scenario, the impulsive assumption is reasonably accurate and the results for the finite burn algorithm should therefore match those of the impulsive fairly closely. Table 1 lists the convergence data for this case.

Table 1. Level I Targeting Results

	Impulsive	Finite Burn
Initial Position Error (km)	3.48696e5	3.91018e6
Final Position Error (km)	7.29519e-6	8.83104e-5
FPA Error (deg)	0.0112	0.0123
Azimuth Error (deg)	0.0661	0.0782

The maneuver Δv and final constraint error values are very similar for the impulsive and finite burn Level I processes. It is also evident, from Table 1, that there appears to be a significant discrepancy in the initial position error provided to each targeter. The discrepancies between the startup arcs for each targeter originate from the estimation of the finite burn maneuver previously described. That is, the direction of the burn, which is assumed to be inertially fixed, is initially aligned with the $\Delta\mathbf{v}$ vector computed by the impulsive targeter. For a burn of finite duration, the direction of the burn becomes more significant as the integration time increases. Thus, an error in the thrust direction is the source of the increased initial position error reported in Table 1. As a result of this larger initial error, the finite burn targeter requires added iterations to converge on a solution in this case. However, the larger error is also a good indication of the accuracy of the impulsive assumption in this example.

In general, both cases considered here required an increased number of iterations relative to later examples. This is attributed to the sensitivities traditionally associated with a Level I process. Specifically, the success of a Level I process is sensitive to the integration time. Since this particular example employs a Level I process to transfer the vehicle from the vicinity of the moon, at TEI-3, to the entry interface at Earth, the time of flight is

too long for the number of control parameters available. Here, a Level II process[4] becomes useful because it allows for an increased number of control parameters and also the ability to incorporate an arbitrary number of constraints. The development of a Level II process that accommodates finite burn maneuvers is discussed next.

B. Level II Process

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. The duration of the burn does not exceed the time associated with the following patch point. Thus, a velocity discontinuity can occur, during the Level I process, at the point where the coast subarc, as defined in Figure 3, joins with the following trajectory segment. Although this problem at first seems identical to the impulsive maneuver targeting, since the velocity discontinuity falls between two coast arcs, the partial derivatives for $\delta\mathbf{v}_k^-$ with respect to $\delta\mathbf{r}_{k-1}$, 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, $\delta m_{k-1}^+ = \ddot{m}_{g_{k-1}}^+ = 0$ and $\dot{\mathbf{u}}_{k-1}^+ = \mathbf{0}$. It is still assumed that $\dot{m}_g$ is a fixed constant, i.e. $\delta\dot{m}_{gk-1}^+ = 0$. Using these relationships and assumptions, along with Equations (1) and (3), an expression is found for $\delta\mathbf{v}_k^-$ in terms of the state at patch point $k-1$ and the state transition matrix,

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

In order to write $\delta\mathbf{v}_k^-$ only in terms of the Level II control variables, the first vector equation from Equation (1) is used to solve for $\delta\mathbf{v}_{k-1}^+$, $\delta\mathbf{u}_{k-1}^+$, and δt_T in terms of those control

variables. From the minimum norm solution,

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

where $Z = \begin{bmatrix} (A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1}) & (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & B_{k,T}(\mathbf{a}^-_T - \mathbf{a}^+_T) \end{bmatrix}$. 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

$$\tilde{Z} = \begin{bmatrix} (C_{k,T}B_{T,k-1} + D_{k,T}D_{T,k-1}) & (C_{k,T}G_{T,k-1} + D_{k,T}J_{T,k-1}) & D_{k,T}(\mathbf{a}^-_T - \mathbf{a}^+_T) \end{bmatrix} Z^T(ZZ^T)^{-1}. \tag{17}$$

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

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

The above 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 Δv 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, $\Delta\mathrm{v}_k$ is given by

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

where $\Delta t_{burn} = t_T - t_k$. The partial derivative of $\Delta\mathrm{v}_k$ with respect to Δt_{burn} at patch point k is given by

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

Next, the partial derivatives of Δt_{burn} with respect to the control variables are necessary. This is 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 $\Delta t_{burn} = t_T - t_k$, the partials are found to be

$$\begin{aligned}
\frac{\partial\Delta t_{burn}}{\partial\mathbf{r}_{k-1}} &= -\frac{\hat{u}_k^T}{||\Delta\mathbf{a}_T||}(D_{T,k}B_{k-1,k}^{-1} - \tilde{S}B_{T,k}B_{k-1,k}^{-1}),\\
\frac{\partial\Delta t_{burn}}{\partial t_{k-1}} &= \frac{\hat{u}_k^T}{||\Delta\mathbf{a}_T||}(D_{T,k}B_{k-1,k}^{-1} - \tilde{S}B_{T,k}B_{k-1,k}^{-1})\mathbf{v}_{k-1}^+,\\
\frac{\partial\Delta t_{burn}}{\partial\mathbf{r}_k} &= -\frac{\hat{u}_k^T}{||\Delta\mathbf{a}_T||}[(C_{T,k} + D_{T,k}D_{k,k-1}B_{k,k-1}^{-1}) - \tilde{S}(A_{T,k} + B_{T,k}D_{k,k-1}B_{k,k-1}^{-1})],\\
\frac{\partial\Delta t_{burn}}{\partial t_k} &= \frac{\hat{u}_k^T}{||\Delta\mathbf{a}_T||}([(C_{T,k} + D_{T,k}D_{k,k-1}B_{k,k-1}^{-1}) - \tilde{S}(A_{T,k} + B_{T,k}D_{k,k-1}B_{k,k-1}^{-1})]\mathbf{v}_k^-\\
&\quad - (D_{T,k} - \tilde{S}B_{T,k})(\mathbf{a}_k^- - \mathbf{a}_k^+) - (H_{T,k} - \tilde{S}E_{T,k})\dot{m}_g),\\
\frac{\partial\Delta t_{burn}}{\partial\mathbf{r}_{k+1}} &= \frac{\hat{u}_k^T}{||\Delta\mathbf{a}_T||}(C_{T,k+1} - \tilde{S}A_{T,k+1}),\\
\frac{\partial\Delta t_{burn}}{\partial t_{k+1}} &= -\frac{\hat{u}_k^T}{||\Delta\mathbf{a}_T||}[(C_{T,k+1} - \tilde{S}A_{T,k+1})\mathbf{v}_{k+1}^- + (D_{T,k+1} - \tilde{S}B_{T,k+1})\mathbf{a}_{k+1}^-],
\end{aligned} \tag{21}$$

where $S = \begin{bmatrix} -G_{T,k} & B_{T,k+1} \end{bmatrix}$ and $\tilde{S} = \begin{bmatrix} -J_{T,k} & D_{T,k+1} \end{bmatrix} S^T(SS^T)^{-1}$. These partials leave one term, containing δm_k, unaccounted for. The cost of a maneuver at patch point k depends on the duration of the burn at k and the initial mass m_k. Since $\dot{m}_g$ is a fixed, constant value, δm_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, as shown above. Using the chain rule, the final form of the partial derivative of the constraint α (the sum of all the burn Δvs) with respect to any

control variable β_k in the set of control variables associated with patch point k is

$$\frac{\partial \alpha}{\partial \beta_k} = \sum_{n=1}^{N_{\Delta \mathrm{v}}} \frac{\partial \Delta \mathrm{v}_n}{\partial \beta_k} \tag{22}$$

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

$$\frac{\partial \Delta \mathrm{v}_n}{\partial \beta_k} = \frac{\partial \Delta \mathrm{v}_n}{\partial \Delta t_{burn_n}} \frac{\partial \Delta t_{burn_n}}{\partial \beta_k} + \frac{\partial \Delta \mathrm{v}_n}{\partial m_n} \frac{\partial m_n}{\partial \beta_k}. \tag{23}$$

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

$$\frac{\partial m_n}{\partial \beta_k} = \frac{\partial m_{n-1}}{\partial \beta_k} - \dot{m}_{g_{n-1}} \frac{\partial \Delta t_{burn_{n-1}}}{\partial \beta_k}. \tag{24}$$

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]

IV. Simulation and Results

The two level targeting algorithm is applied in the following section in both the impulsive and finite burn configurations. This is done in order to demonstrate its performance under both configurations and to gain insight as to applications of each. Then, as a final metric, an optimal trajectory is generated to use for performance comparisons.

For each case, the same initial conditions of the lunar orbit will be used as follows:

- Epoch: 4-Apr-2024 15:30:00 TDT
- Initial mass: 20339.9 kg (total fuel = 8063.65 kg)
- Main Engine Thrust: 33,361.6621 N
- Main Engine Isp: 326 sec
- Auxiliary Engine Thrust: 4,448.0 N
- Auxiliary Engine Isp: 309 sec
- State (J2000 Moon-centered inertial frame):
  - X: -1236.7970783385588 km
  - Y: 1268.1142350088496 km

- Z: 468.38317094160635 km
- Vx: 0.0329108058365355 km/sec
- Vy: 0.589269803607714 km/sec
- Vz -1.528058717568413 km/sec

Likewise, the same terminal target conditions will be used for each case as shown below:

- Geodetic Altitude (km) 121.92
- Longitude (deg) 175.6365
- Geocentric Azimuth (deg) 49.3291
- Geocentric Flight Path Angle (deg) -5.86

A. Finite Burn Example with Main Engines

The first case is representative of a nominal Earth return during which the maneuvers are performed by the CM main engine. The initial guess file consists of 12 "patch points" or states taken from an optimized trajectory. 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 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 (EI).

For this case, both the impulsive targeter and the finite burn targeter are executed in order to find a feasible trajectory that satisfies the specified terminal constraints, while keeping the Δv sum of the individual maneuvers within the available fuel budget. Table 2 compares the individual maneuvers and final Δv sum for the impulsive solution and the finite burn solution. The burn parameters for each finite burn maneuver are given in Table 3, and final constraint errors are listed in Table 4.

Table 2. Maneuver Data

Maneuver	Impulsive Δv (km/s)	Finite Burn Δv (km/s)
TEI-1	0.6619	0.7348
TEI-2	0.3257	0.2561
TEI-3	0.4115	0.4087
Total	1.3991	1.3996

The individual maneuvers and total Δv sum for the finite burn targeter are fairly similar to the impulsive targeting results, which is to be expected given that the burn durations

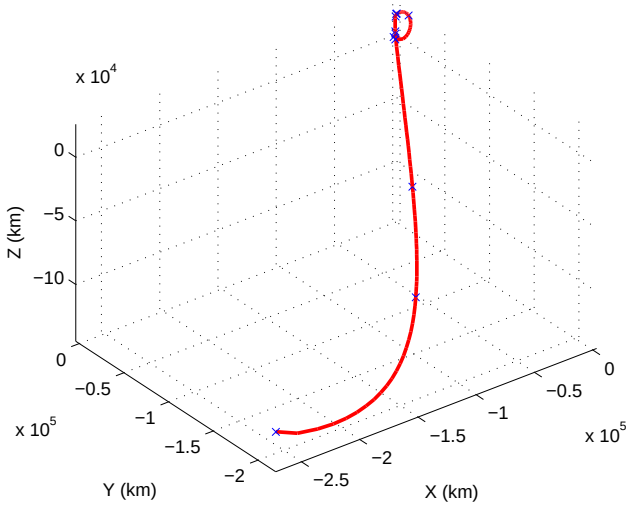


Figure 5. Initial Guess and Impulsive Solution

Table 3. Burn Data

Maneuver	Duration (s)	Prop. Mass Consumed (kg)
TEI-1	400.2308	4175.128
TEI-2	119.2402	1243.890
TEI-3	171.6074	1790.174
Total	691.0784	7209.192

Table 4. Constraint Error Data

	Impulsive Algorithm	Finite Burn Algorithm
Iterations	20	6
Altitude (km)	-1.0e-8	-5.2e-8
Flight Path Angle (deg)	2.8e-10	-3.3e-10
Longitude (deg)	4.0e-8	-5.0e-7
Flight Path Azimuth (deg)	-2.0e-7	1.6e-7

with the main engines 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.

B. 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. Figures 6 through 8 show the closeup views of TEI-1, TEI-2, and TEI-3, and the solid red portions of these closeups indicate the segments of the trajectory in which the engines are thrusting. Burn data for each maneuver is listed in Table 5, and the final constraint errors are given in Table 6.

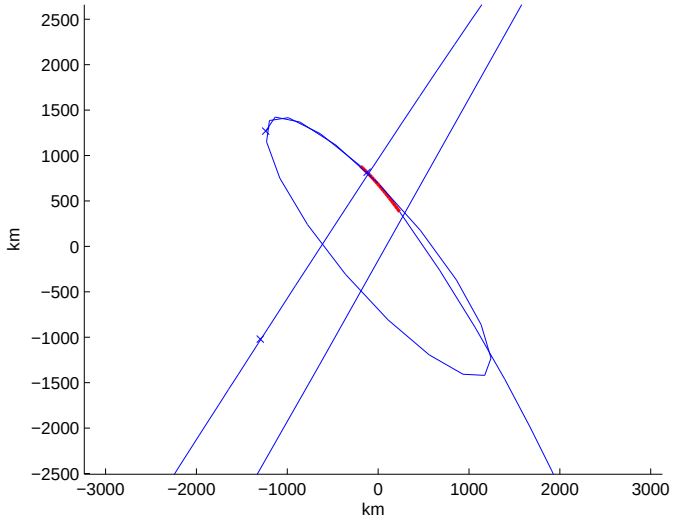


Figure 6. Finite Burn Solution with Auxiliary Engines, TEI-1 Closeup

Table 5. Burn Data Using Auxiliary Engines

Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	363.5548	3792.531	0.6255
TEI-2	949.2614	1392.946	0.2666
TEI-3	1400.5756	2055.205	0.4418
Total	2713.3918	7240.682	1.3339

The finite burn algorithm is able to meet the entry and cost constraints in only a few more iterations than the impulsive targeter required. Interestingly, the total Δv for this case

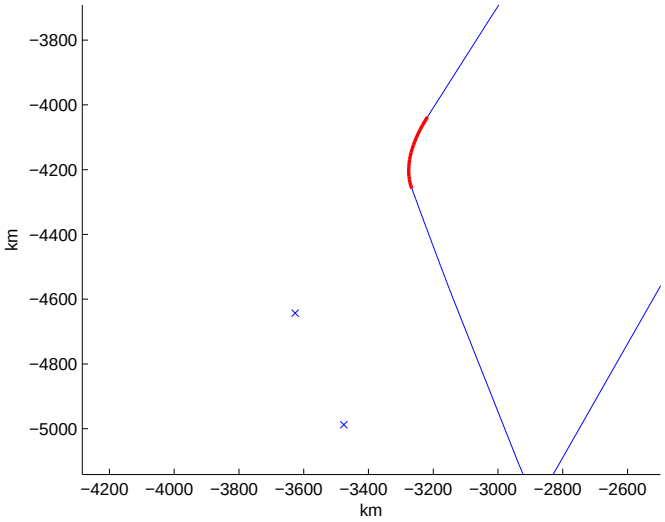


Figure 7. Finite Burn Solution with Auxiliary Engines, TEI-2 Closeup

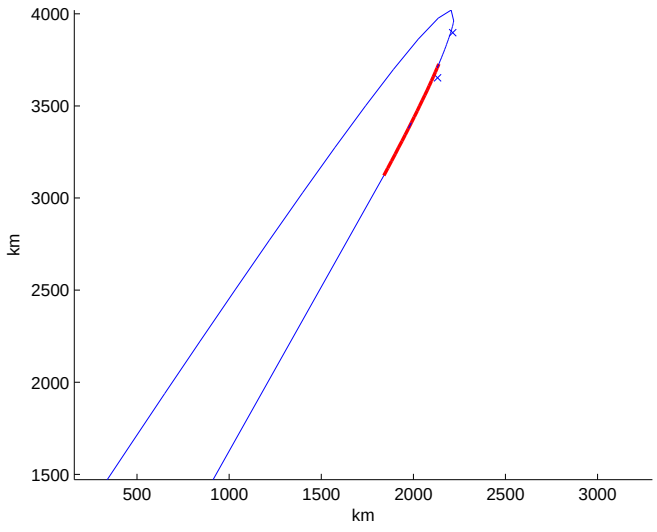


Figure 8. Finite Burn Solution with Auxiliary Engines, TEI-3 Closeup

Table 6. Constraint Error Data with Auxiliary Engines

Iterations	23
Altitude (km)	4.1e-7
Flight Path Angle (deg)	4.5e-10
Longitude (deg)	2.1e-6
Flight Path Azimuth (deg)	-2.6e-7

is about 0.065 km/s *lower* than the impulsive solution. This will not always be the case; numerous feasible solutions can exist for any given set of patch points. For this particular example, 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.

C. 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 Copernicus. The results of this optimized run are available in Table 7. The total Δv for the optimal run is 1.2413 km/s. This is an improvement of approximately 0.09 km/s of Δv 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 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.

Table 7. Optimal Burn Data Using Auxiliary Engines

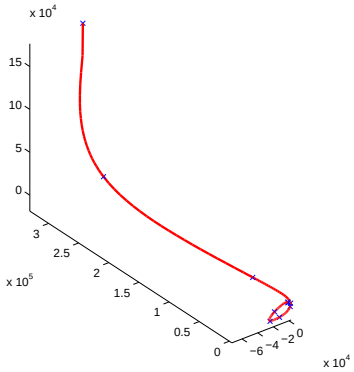
Maneuver	Duration (s)	Prop. Mass Consumed (kg)	Δv (km/s)
TEI-1	3327.008	3501.37	0.6040
TEI-2	873.040	1504.85	0.2993
TEI-3	1312.285	1828.22	0.4059

D. Finite Burn Example Over The Lunar Cycle

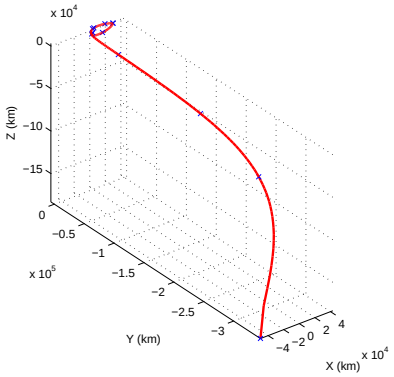
To further test the finite burn algorithm, return trajectories were generated over several days spanning the lunar cycle from February 1-28, 2024, 0:00:00 TDT, again using the auxiliary engines for the second and third TEI maneuvers. For these runs, only two entry constraints are targeted:

- Geodetic Altitude (km) 121.92
- Geocentric Flight Path Angle (deg) -5.86

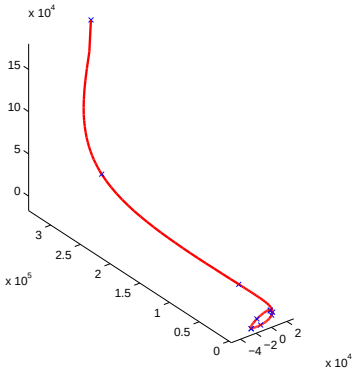
In this simulation, the input patch points to the finite burn algorithm come from a converged impulsive trajectory with the same initial point and entry targets. The total Δv of the impulsive trajectory for each case is 1.50 km/s. Table 8 lists results for days 1, 3, 6, 10, 13, 16, 19, 22, 25, and 28 of the lunar cycle. Figures 9(a)-9(c) show some examples of these trajectories.



(a) Day 1



(b) Day 13



(c) Day 28

Figure 9. Return Trajectories (MCI) at Day 1 (a), Day 13 (b), and Day 28 (c) of the Lunar Cycle

With the exception of the simulations for February 6 and February 16, the targeter converges to valid solution within the entry and cost constraints for each day tested over the lunar cycle. The solutions for February 6 and 16 satisfy the entry constraints, but would not converge when the total cost constraint of 1.50 km/s was imposed. The values listed in Table 8 are the total cost of the converged solution without the maneuver sum constraint active. The precise effect of the quality of initial guess on a given date will be the subject of future investigations.

Table 8. Burn Data over the Lunar Cycle

Day	TEI-1		TEI-2		TEI-3		Total Cost	
	ΔV (km/s)	Duration (s)	ΔV	Duration	ΔV	Duration	Total ΔV	Iterations
1	0.5963	348.1673	0.4741	1648.5129	0.3995	1202.2754	1.4698	4
3	0.6837	393.6876	0.4393	1492.3184	0.3585	1067.4428	1.4814	4
6	0.5901	344.8973	0.5271	1820.9042	0.3929	1165.7458	1.5101	15
10	0.5849	342.1671	0.5808	1992.6327	0.3259	961.9933	1.4916	4
13	0.5869	343.1966	0.3661	1299.5151	0.5079	1561.4187	1.4609	4
16	0.6806	392.0921	0.3205	1111.0622	0.6251	1856.5447	1.6262	39
19	0.5778	338.3921	0.5925	2033.8077	0.3189	940.8576	1.4892	4
22	0.7004	402.2770	0.3915	1332.7907	0.4080	1217.7097	1.4999	11
25	0.5862	342.8320	0.6318	2149.2973	0.2740	801.4648	1.4919	4
28	0.5861	342.7945	0.4912	1709.0768	0.3882	1167.9040	1.4656	4

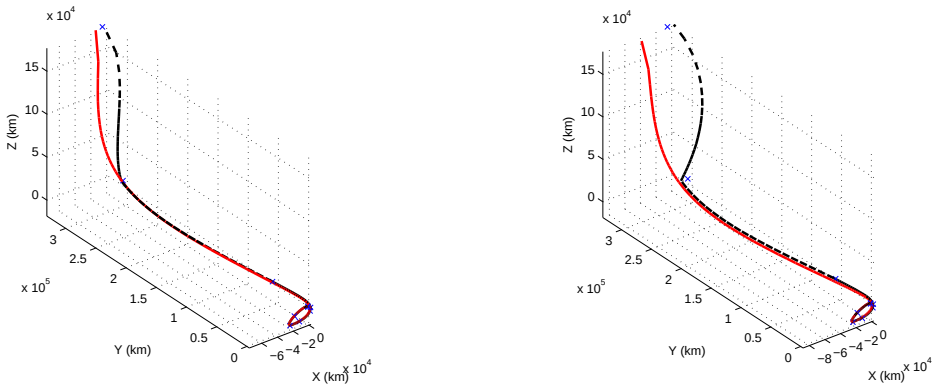
E. Delayed Patch Point Simulations

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 February 1 run in the previous section is used. As before, the auxiliary engines perform the TEI-2 and TEI-3 maneuvers. 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 first for only 3 hours, then for a full 12 hours. Figures 10(a)-10(b) show the difference in the initial and final converged trajectories for time delays of 3 hours (10(a)) and 12 hours (10(b)). The initial trajectory appears as a dashed line in Figures 10(a)-10(b). The initial entry constraint errors due to the time delay are given in Table 9, and the final constraint errors for the converged trajectories are listed in Table 10.

Table 9. Initial Entry Constraint Errors

Delay (hr)	Altitude Error (km)	FPA Error (deg)
3	1.1040e4	1.6464
12	4.5332e4	10.4268

Even with a 12 hour delay, the finite burn algorithm is able to find a feasible solution that satisfies the entry constraints and fuel budget. The 12 hour delay trajectory even shows



(a) 3 Hours (b) 12 Hours

Figure 10. Initial and Final Trajectories (MCI) at 3 Hours (a) and 12 Hours (b) Delay

Table 10. Convergence with Delays

Delay (hr)	Total ΔV (km/s)	Altitude Error (km)	FPA Error (deg)	Iterations
3	1.4655	-2.8218e-5	-9.4476e-8	10
12	1.2961	6.0979e-6	1.2184e-8	12

a surprisingly marked decrease in the total Δv cost for this case. As with the low-cost auxiliary engine solution discussed previously, it is the sensitive dynamics of the system that accounts for this result rather than any special property of the finite burn targeter. These results do, however, underscore the flexibility 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 that is more desirable than the one suggested by the initial input.

V. Conclusions

This paper presents a two-level targeting algorithm for finite burn maneuvers. The algorithm is adapted from a classical impulsive two-level targeting algorithm. The development of the finite burn versions of both the Level I and Level II process are discussed, and a total mission cost constraint, originally developed for the impulsive algorithm, is modified and adapted to the finite burn problem. The algorithm keeps the same structure and much of the simplicity of the impulsive two-level corrector even though the partial derivatives required for the calculations are far more complex. For testing, the algorithm is applied to the trans-Earth injection phase of the Orion mission. Results are compared for three different cases: using the impulsive algorithm, using the finite burn algorithm with main engines for all three maneuvers, and using the finite burn algorithm with only the auxiliary engines

for the final two maneuvers. These results show that the finite burn algorithm is able to converge on a feasible solution even in the case of a main engine failure following TEI-1. Due to the additional complexity of the finite burn model, the algorithm does exhibit increased computational overhead in contrast to the impulsive targeter. However, the present formulation addresses the need for a targeting algorithm that accommodates the main engine failure scenario while meeting all the specified constraints.

VI. Acknowledgements

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration. This research was carried out at The University of Texas at Austin and funded under NASA Award Number NNX07AR46G.

References

[1]Hyde, C. T., Foggat, C. E., and Weber, B. D., "Apollo Experience Report - Abort Planning," Tech. Rep. NASA-TN-D-6847, National Aeronautics and Space Administration, 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," Tech. Rep. NASA-TN-D-6848, National Aeronautics and Space Administration, 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*, Savannah, GA, Feb. 2009, Paper No. AAS 09-195.

[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.

[5]Jezewski, D. J., "Primer Vector Theory and Applications," NASA Technical Report R-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.

[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.

[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.

[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.

[10]Zimmer, S. and Ocampo, C., "Use of Analytical Gradients to Calculate Optimal Gravity-Assist Tra-

jectories," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 2, 2005, pp. 324–332.

[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.

[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.

[13]Howell, K. C. and Pernicka, H. J., "Numerical Determination of Lissajous Trajectories in the Restriced Three-Body Problem," *Celestial Mechanics*, Vol. 41, 1988, pp. 107–124.

[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, Sun Valley, Idaho*, 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*, Honolulu, HI, Aug. 2008, AIAA-2008-7262.

[17]Smith, C. W., *An Onboard Targeting Algorithm with Earth-Return Applications*, Master's thesis, University of Texas at Austin, August 2008.