AIAA Guidance, Navigation and Control Conference and Exhibit
18 - 21 August 2008, Honolulu, Hawaii

AIAA 2008-7262

AIAA Guidance, Navigation and Control Conference and Exhibit, 18- 21 Aug 2008, Honolulu, Hawaii

Design of the Onboard Autonomous Targeting Algorithm for the Trans-Earth Phase of Orion

Michael W. Weeks[*] Belinda G. Marchand[†] Chad W. Smith[‡] Sara Scarritt[‡]

This paper discusses the adaptation and implementation of a modified two-level corrections process as the onboard targeting algorithm for the Trans-Earth Injection phase of the Crew Exploration Vehicle. Unlike earlier Apollo missions to the Moon, Project Orion intends to land near the polar regions, a task that can lead to a substantial plane change maneuver prior to Earth return. To reduce the fuel expenditure associated with this plane change, a three-maneuver sequence is employed during the return phase. First, the initial maneuver raises apolune at departure. The subsequent maneuver executes a plane change. The third and final maneuver places the spacecraft on its final approach to Earth while targeting the entry state for precision landing. An autonomous onboard targeting algorithm is sought in the event of loss of communication with the ground. This last scenario presents a very unique challenge, one never required of any Apollo vehicle. The Apollo missions also benefited from flexible entry requirements in contrast to Orion. Precision targeting in multi-body regimes has only been previously demonstrated in unmanned sample return missions, like Genesis. The two-level corrector formulation presented here ensures the entry constraints are met without violating the available fuel budget.

I. Introduction

The the onboard fight software for targeting the Crew Exploration Vehicle (CEV) Earth return is envisioned as a two-level corrections process that targets the desired entry state while ensuring the available fuel budget is not exceeded. Currently, the Trans-Earth Injection (TEI) sequence incorporates three impulsive maneuvers, as illustrated in Figure 1.

The goal of the first two maneuvers is to minimize the final injection that places the spacecraft on a path to Earth given a set of entry constraints. Specifically, the vehicle must meet constraints on flight path angle, latitude, longitude, altitude, and azimuth, all functions of latitude set by the Earth-Moon antipode.[1] In earlier designs, the targeting of these final constraints is controlled only by the final TEI maneuver.[2] While this approach does address some of the goals, it does not fully exploit the first two maneuvers.

For this mission, the success of the targeting process is greatly impacted by third body effects, since the Earth and the Moon are equally significant over certain segments of the trajectory. Experience in trajectory design in multi-body regimes reveals that the process of enforcing interior and end-point constraints benefits greatly from an end-to-end design that simultaneously includes all maneuvers with the objective to meet the final desired constraints. This was previously demonstrated in the design of the Genesis trajectory,[3–5] and later expanded to various applications by Marchand, Howell, and Wilson.[6] As detailed in this latest work, a constrained two-level corrections process is not limited to spacecraft mission design applications. It is designed as a generalized numerical targeting scheme for the analysis of dynamical systems that are subject to constraints. One of the most appealing aspects of this methodology is its numerical simplicity and computational efficiency in contrast to trajectory optimizers. This makes the proposed approach more feasible for autonomous onboard targeting.

[*] GNC Engineer, Aeroscience and Flight Mechanics Division, NASA JSC, 2101 NASA Pkwy. Houston, Texas 77058 / EG-6, AIAA Member.

[†] Assistant Professor, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, 210 E. 24th St., Austin, TX 78712, AIAA Member.

[‡] Graduate Student, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, 210 E. 24th St., Austin, TX 78712, AIAA Student Member.

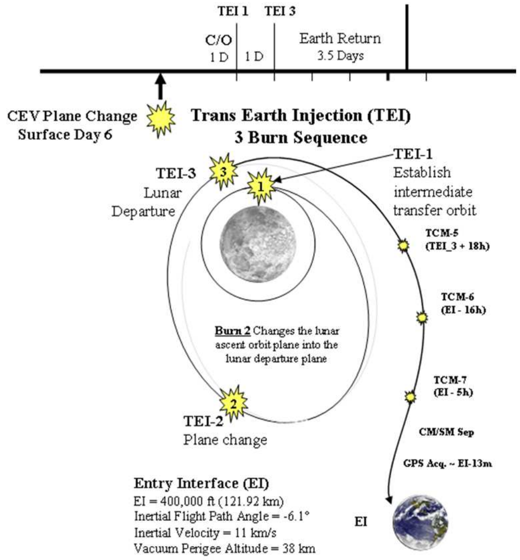


Figure 1. Three dimensional representation of the TEI sequence

The targeting approach described here is formulated completely within the parameters of a generalized perturbed multi-body problem. A constrained two-level corrections process,[4–6] as that used for the preliminary design of the Genesis trajectory, is used as a prototype for the onboard autonomous targeting algorithm. The basics of the algorithm are derived from a time-varying linearized dynamical model, and the solution updates are computed using a minimum norm solution.[7] This type of solution minimizes the requested changes in the control parameters and ensures the search for a solution remains within the vicinity of the startup arc and improves the convergence of the linear process. These updates are incorporated in the nonlinear system through an iterative process until a feasible solution is identified. Of course, adjustments are necessary to account for the autonomous targeting requirement. However, an advantage of this approach is that it does not rely on knowledge of a "nominal" solution. Instead, the entire process depends on the current path of the spacecraft.

It is important to note that earlier targeting algorithms are an excellent stepping stone for the current method,[2] particularly in the identification of suitable startup arcs. Some of the entry interface constraints previously mentioned, like altitude, latitude, and flight path angle, have already been developed and tested in earlier investigations.[4–6] Though a formulation for the true latitude constraint, as defined relative to a non-spherical Earth, is not yet available, one is not yet necessary since the Earth is assumed to be a spherical body in this study. The azimuth and longitude constraints, both measured in time varying coordinate systems, are developed here. The two-level process requires the identification of the functional relation between each of the constraints and the state vector. Then, the state relationship matrix[4–6] is constructed by incorporating the partial derivatives associated with the constraints of interest. A total cost constraint is also developed and successfully implemented. These constraint formulations are employed in assessing the feasibility of the two-level process for onboard targeting. A comparison between targeter and optimal solutions is performed to highlight the significantly reduced computational overhead of the targeting process. This is a key requirement of any targeting algorithm for onboard determination.

II. Constrained Two-level Corrector

The first step in a constrained two-level corrections process[6] requires that the current spacecraft trajectory be divided into a series of arcs. The arcs may be determined from a full ephemeris model analysis, a simplified three-body model, a multi-body numerical solution, or a conic. The final form of the dynamical equations is not important since the algorithm is completely formulated in a generalized form.

An arc can also incorporate some additional force, if appropriate, such as solar radiation pressure. In the next step of the process, the specific constraints are modeled, as well as the associated partials. These partial derivatives are used to construct a state relationship matrix. This matrix is employed in a two-level differential corrections process to enforce constraints along the trajectory.

A. Dynamical Model

The elements of the spacecraft state vector, $\bar{X}(t) = [x(t), y(t), z(t), \dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$, represent components of the spacecraft position and velocity, associated with a generic reference frame. Based on this definition, the nonlinear differential equations that govern the evolution of $\bar{X}(t)$, in any gravitational regime, may be generally represented as,

$$\dot{\bar{X}}(t) = \bar{f}\left(\bar{X}(t)\right). \tag{1}$$

Let $\bar{X}^*(t)$ and $\dot{\bar{X}}^*(t)$ identify a reference state, and the associated time derivative. Then, according to Equation (1), $\dot{\bar{X}}^*(t) = \bar{f}\left(\bar{X}^*(t)\right)$. Suppose this solution exists in the vicinity of the current path, defined by $\bar{X}(t)$. Then,

$$\bar{X}^*(t) = \bar{X}(t) + \delta\bar{X}(t) \tag{2}$$

The vector $\delta\bar{X}(t)$ represents a contemporaneous variation, the difference between the true and reference states measured at the same point in time. The linearized model associated with Equation (1) admits a solution of the form

$$\left(\delta\bar{X}_k' - \dot{\bar{X}}(t_k)\,\delta t_k\right) = \Phi(t_k, t_{k-1})\left(\delta\bar{X}_{k-1}' - \dot{\bar{X}}(t_{k-1})\,\delta t_{k-1}\right), \tag{3}$$

where $\Phi(t_k, t_{k-1})$ denotes the state transition matrix (STM), and $\delta\bar{X}_k'$ and $\delta\bar{X}_{k-1}'$ are non-contemporaneous variations.[6] That is, the difference between the true and reference state vector but measured at different times.

In general, the 6×6 STM, $\Phi(t_k, t_{k-1})$, can be subdivided into four 3×3 submatrices such that

$$\Phi(t_k, t_{k-1}) = \begin{bmatrix} A_{k,k-1} & B_{k,k-1} \\ C_{k,k-1} & D_{k,k-1} \end{bmatrix}. \tag{4}$$

The subscript pair "$k, k-1$" denotes the direction of the propagation. For example, the right subscript, "$k-1$", denotes the start time, t_{k-1}, of the propagation while the subscript to the left, "k", reflects the terminal time, t_k. Consequently, $\Phi(t_k, t_{k-1})^{-1} = \Phi(t_{k-1}, t_k)$ may also be decomposed into sub-matrices,

$$\Phi(t_{k-1}, t_k) = \Phi(t_k, t_{k-1})^{-1} = \begin{bmatrix} A_{k-1,k} & B_{k-1,k} \\ C_{k-1,k} & D_{k-1,k} \end{bmatrix}. \tag{5}$$

Note that, while $\Phi(t_{k-1}, t_k) = \Phi(t_k, t_{k-1})^{-1}$, the same relation does not hold for the submatrices of $\Phi(t_{k-1}, t_k)$ and $\Phi(t_k, t_{k-1})^{-1}$. For instance, $A_{k-1,k} \neq A_{k,k-1}^{-1}$. Expanding Equation (3) in terms of the submatrices in Equation (4) leads to

$$\begin{bmatrix} \delta\bar{R}_k' - \bar{V}_k^-\delta t_k \\ \delta\bar{V}_k^{-\,\prime} - \bar{a}_k^-\delta t_k \end{bmatrix} = \begin{bmatrix} A_{k,k-1} & B_{k,k-1} \\ C_{k,k-1} & D_{k,k-1} \end{bmatrix} \begin{bmatrix} \delta\bar{R}_{k-1}' - \bar{V}_{k-1}^+\delta t_{k-1} \\ \delta\bar{V}_{k-1}^{+\,\prime} - \bar{a}_{k-1}^+\delta t_{k-1} \end{bmatrix} \tag{6}$$

Alternatively, if the flow proceeds backwards from t_k to t_{k-1}, or

$$\begin{bmatrix} \delta\bar{R}_{k-1}' - \bar{V}_{k-1}^+\delta t_{k-1} \\ \delta\bar{V}_{k-1}^{+\,\prime} - \bar{a}_{k-1}^+\delta t_{k-1} \end{bmatrix} = \begin{bmatrix} A_{k-1,k} & B_{k-1,k} \\ C_{k-1,k} & D_{k-1,k} \end{bmatrix} \begin{bmatrix} \delta\bar{R}_k' - \bar{V}_k^-\delta t_k \\ \delta\bar{V}_k^{-\,\prime} - \bar{a}_k^-\delta t_k \end{bmatrix} \tag{7}$$

Note that, when the propagation proceeds forward from t_{k-1} to t_k, $\bar{V}_k^-$ and $\bar{a}_k^-$ represent the arrival velocity and acceleration vectors, along the current solution, at the terminal node. Similarly, $\bar{V}_{k-1}^+$ and $\bar{a}_{k-1}^+$ are the departure velocity and acceleration vectors associated with the initial node on the current trajectory segment.

The vector $\delta\bar{R}_{k-1}^{'}$ represents the difference, or non-contemporaneous variation, in position between the initial node on the reference solution and the initial node on the current solution. Furthermore, the vector $\delta\bar{V}_{k-1}^{'}$ denotes the non-contemporaneous variation in velocity between the reference and current solutions. Similar definitions apply for $\delta\bar{R}_{k}^{'}$ and $\delta\bar{V}_{k}^{'}$.

B. Startup Arcs and Patch States

Targeters and optimizers are not self-starting processes. Both require a reasonably accurate initial guess to proceed. Such a trajectory arc may result from a numerical integration process, perhaps one such that the path does not necessarily satisfy the specified constraints. Other alternatives, even conics, can also serve as a startup arcs in the ephemeris model. In this study, the initial guess is constructed by dividing the trajectory into two components, a Moon centered and an Earth centered segment. The Moon centered segment encompasses everything prior to TEI-3. This consists of an initial burn (TEI-1) to raise apolune and a second burn (TEI-2) to change the orbital inclination. These computations are performed in a two-body regime, but implemented in the ephemeris model. From TEI-2, the solution is propagated, in the ephemeris model, up to the general vicinity of TEI-3. Then, a simple Level I targeting scheme, with fixed terminal time, is used to target the entry altitude, latitude, and longitude with a single maneuver. Since the entry flight path angle and azimuth are not specifically targeted in this initial guess, it is likely that the resulting values are significantly off target. However, the startup arc is otherwise feasible and satisfies at least three of the five entry constraints.

Once the startup solution is available, the trajectory is decomposed into segments and nodes or patch points. The process of selecting patch points is somewhat open ended. It is important, however, that the patch points be selected adequately to capture the overall character and geometry of the solution to enhance convergence. At the same time, an excessive number of patch points unnecessarily constrains the search process and limits the solution space the corrector is able to explore. Traditionally, the selection of the patch states is based on engineering intuition and experience. However, the systematic identification of suitable patch states that facilitate the monotonic convergence of the process is an ongoing subject of study, though outside the scope of the present investigation. It is interesting to note, however, that the location of the patch states, and subsequently the maneuvers, does impact the computational efficiency of the algorithm. This is particularly true when the process is initiated at a point near the Moon and targets a maneuver near the sphere of influence. A level one process can encounter difficulties in this particular scenario. The problem is easily resolved by relocating the patch state either forward or backward in time. Based on this observed behavior, it is possible to identify specific criteria or characteristics that qualify a patch state as suitable for a level one targeting process. This is important since the level one process represents the bulk of the computational effort spent in the combined two-level corrector.

III. Two-Level Targeter

The initial development of the two-level corrector was first presented by Howell and Pernicka[8] as a tool for the identification of natural quasi-periodic arcs in the three-body regime. Later, Howell and Wilson[5] extend the method to incorporate trajectory constraints for mission design applications in the three-body problem. More recently, Marchand, Howell, and Wilson[6] demonstrate the wide range of applications of the two-level targeting scheme as a general end-to-end design tool for analysis of dynamical systems. Some of the appealing features of the two level-targeter, in general, are its versatility, flexibility, and adaptability. Although the method was not initially envisioned[6] as an onboard autonomous targeting tool, the general theory is adaptable for autonomous targeting if all the relevant issues are properly addressed. These issues include the (a) development of a simple and computationally inexpensive initial guess generator that can reasonably approximate the motion in the presence of multiple gravitational perturbations, (b) an automatic patch state selection algorithm that is able to identify the patch state characteristics necessary to offer enhanced convergence properties, (c) automated convergence monitoring and adaptive step techniques, and (d) flight logic for the selection of the necessary constraints based on an entry scenario and available resources. Furthermore, the on-board targeter must (e) incorporate finite burns that employ realistic thruster models and (f) address constraints that are dependent on multiple patch states.

Ongoing investigations have successfully demonstrated the use of a two-level targeter that employs finite burns rather than impulsive maneuver, per item (e) above. Some preliminary results are presented in this document. A constraint dependent on multiple patch states, per item (f), is presented and successfully implemented in this study. This constraint targets the total mission cost within the available fuel budget. A candidate initial guess algorithm is developed and employed in this investigation, per item (a) above, but requires further refinement to enhance the convergence in the most constrained cases. Specifically, cases that are particularly susceptible to timing sensitivities at departure. Of course, the initial guess algorithm must, itself, be an autonomous process. The present study summarizes the results of initial investigations into the adaptability of the two-level targeter for onboard targeting. Specifically, a comparison of the computational performance of a two-level targeter against an optimizer is presented.

A. Level I Process

The two-level differential corrector[6] consists of two fundamental steps. The Level I process is analogous to – though not the same as – a two-body Lambert solver. Traditionally, a Level I process is a differential correction scheme[9] that seeks to identify the transfer arc between two position vectors by adjusting the departure velocity, as illustrated in 2(a). A Level I targeter differs from a Lambert targeter in that the process is not associated with any specific dynamical model. Thus, the user has the flexibility of employing any level of model complexity desired (i.e. perturbations). The only requirements are the availability of an initial guess and the state transition matrix, a necessary component in the Level I calculations. The Level I targeter also differs from a Lambert targeter in that the time of flight along the path is not necessarily fixed. In fact, the time of flight may be specified as an additional control variable to add flexibility in meeting the desired target. In addition, a Level I process need not target a terminal position vector, it can also be configured to target a terminal set of constraints (altitude, flight path angle, etc.) as long as the number of constraints does not exceed the available control variables. The following examples compare a traditional Level I process, one that targets the terminal position vector, to a modified Level I scheme that targets generalized terminal constraints. A level one process that employs finite burns instead of ΔV_{k-1} is also presented. The finite burn level one process is employed in a modified level two scheme that seeks to incorporate realistic thruster models into the corrections algorithm.

1. *Example 1: Targeting the Terminal Position Vector*

In a traditional Level I process,[9] the segment time is fixed, $\delta t_k = \delta t_{k-1} = 0$, and the control variables are the components of $\Delta \bar{V}_{k-1}$, the propulsive impulse applied at t_{k-1}. However, allowing the terminal time to vary adds flexibility into the solution process, $\delta t_k \neq 0$. For example, consider the illustration in 2(a).

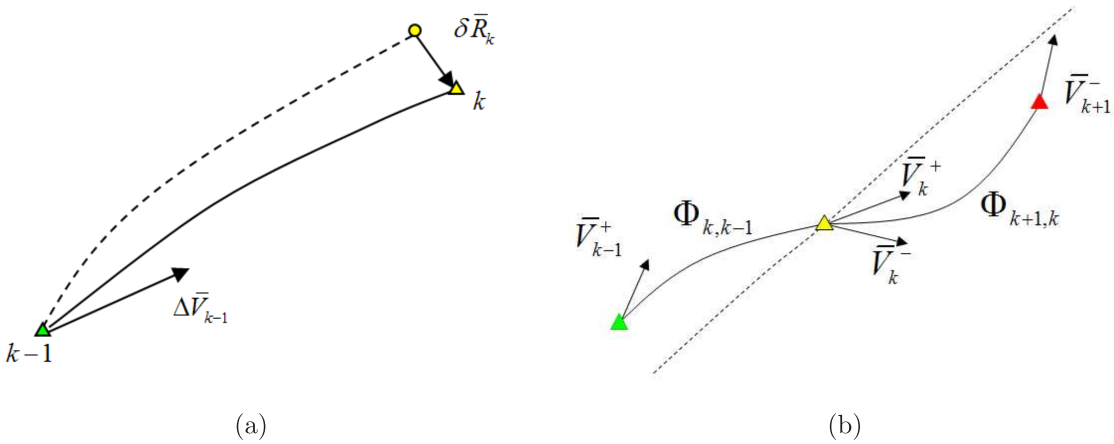


Figure 2. Stylized representation of the structure for the Level II Process

For a fixed time transfer,[6] $\delta t_{k-1} = \delta t_k = 0$, each iterative step of the Level I process updates the outgoing velocity along the segment by

$$\delta\bar{V}^{+}_{k-1} = B^{-1}_{k,k-1}\delta\bar{R}_k, \tag{8}$$

where $\delta\bar{V}^{+}_{k-1} = \delta\bar{V}^{+\,\prime}_{k-1}$ since $\delta t_{k-1} = 0$ and $\delta\bar{R}_k = \delta\bar{R}^{\prime}_k$ because $\delta t_k = 0$. If the terminal segment time is free, a modified update is applicable,

$$\left[\begin{array}{c} \delta\bar{V}^{+}_{k-1} \\ \delta t_k \end{array}\right] = \left[\begin{array}{cc} B_{k,k-1} & \bar{V}^{-}_{k} \end{array}\right]^T \left(\left[\begin{array}{cc} B_{k,k-1} & \bar{V}^{-}_{k} \end{array}\right]\left[\begin{array}{cc} B_{k,k-1} & \bar{V}^{-}_{k} \end{array}\right]^T\right)^{-1}\delta\bar{R}^{\prime}_k, \tag{9}$$

since $\delta t_k \neq 0$. Equation (9) represents the minimum norm solution to the variational position equations when both the initial velocity and terminal time are treated as control variables.

2. *Example 2: Targeting Terminal Constraints*

Though the traditional Level I process targets a terminal position vector, the process is easily tailored to target terminal constraints. That is, a nonlinear vector function of the state at the terminal point that must satisfy some specific value. This is expressed mathematically as follows,

$$\bar{y}(t_k) = h\left(\bar{x}(t_k)\right). \tag{10}$$

The vector $\bar{y}(t_k)$ represents a vector of measurements or conditions at the terminal point. This may include quantities such as altitude, flight path angle, latitude, longitude, etc. Each of these quantities depend explicitly on the state, in this case positions and velocities. If one can identify the functional relationship between the state and the constraint, the linearized approximation of Equation (10) is subsequently determined as,

$$\delta\bar{y}(t_k) = \frac{\partial\bar{h}_k}{\partial\bar{x}_k}\delta\bar{x}(t_k) = C(t_k)\delta\bar{x}(t_k), \tag{11}$$

The matrix $C(t_k)$ represents the partial derivative of the constraint vector relative to the state. The elements of $C(t_k)$ are evaluated at the terminal state. This matrix is traditionally used during a Level II process,[6] but it is employed here in a Level I scheme as a proof of concept. Consider Equation (6) when $\delta t_{k-1} = 0$ and $\delta\bar{R}_{k-1} = \bar{0}$,

$$\begin{aligned} \delta\bar{R}_k &= B_{k,k-1}\delta\bar{V}^{+}_{k-1} + \bar{V}^{-}_{k}\delta t_k, & (12) \\ \delta\bar{V}_k &= D_{k,k-1}\delta\bar{V}^{+}_{k-1} + \bar{a}^{-}_{k}\delta t_k. & (13) \end{aligned}$$

Equations (12)-(13) may also be expressed in a more compact vector form,

$$\delta\bar{x}_k = \underbrace{\left[\begin{array}{cc} B_{k,k-1} & \bar{V}^{-}_{k} \\ D_{k,k-1} & \bar{a}^{-}_{k} \end{array}\right]}_{M_c}\underbrace{\left[\begin{array}{c} \delta\bar{V}^{+}_{k-1} \\ \delta t_k \end{array}\right]}_{\bar{b}_c} \tag{14}$$

Substitution of Equation (14) into Equation (11) leads to,

$$\delta\bar{y}(t_k) = C(t_k)M_c\bar{b}_c. \tag{15}$$

If $\dim\left(\bar{y}(t_k)\right) < \dim\left(\bar{b}_c\right)$, an infinite number of solutions to Equation (15) exist. Subsequently, a minimum norm solution suggests the smallest corrections to the initial velocity and segment duration that meet the desired constraint error value, $\delta\bar{y}(t_k)$, at the terminal point,

$$\bar{b}_c = \left(C(t_k)M_c\right)^T\left[\left(C(t_k)M_c\right)\left(C(t_k)M_c\right)^T\right]^{-1}\delta\bar{y}(t_k). \tag{16}$$

To illustrate the functionality of the above approach, a simple constraint based Level I process is implemented in Table 1 to the transfer from a sample TEI-3 point to Earth entry. The longitude constraint partials are detailed later in this document. The altitude and latitude partials were developed in earlier investigations.[4–6] Since the Earth is assumed to be a spherical body, the latitude is simply determined as the inertial declination relative to the mean equatorial plane.

For the example in Table 1, TEI-3 occurs on December 11, 2014 01:38:19.764 UTC. The entry state of interest corresponds to an altitude of 121.9 km, a flight path angle of -6.03°, a longitude of 134.5456°, and a latitude of -19.2041°. Table 2 summarizes the error in the entry state associated with the initial guess if the state prior to TEI-3 is propagated in the ephemeris model without any corrections. The constraint based Level I process is then employed in Table 1 to determine the TEI-3 maneuver necessary to bridge the gap between the desired entry state and the entry state from the initial guess. Clearly, a significant error exists in all quantities. Thus, this particular initial guess offers an excellent way to demonstrate the type of difficulties a corrections process can encounter when the quality of the initial guess is poor. At the same time, it also demonstrates that the process is also unexpectedly robust under some conditions.

Table 1. Level I Constraint Targeting Results

Entry Constraints	TEI-3 ΔV (km/s)	Iterations	Comp. Time (sec)
Altitude, FPA	0.8796	14	20.0938
Altitude, Latitude	0.8704	7	11.4063
Altitude, Longitude	0.8796	7	10.9375
Altitude, Latitude, Longitude	0.8796	18	24.5000
Altitude, FPA, Latitude	0.8796	16	22.7031

Table 2. Initial Constraint Errors

Constraint	Error
Altitude (km)	5712.2887
FPA (deg)	-34.3866
Latitude (deg)	-2.0963
Longitude (deg)	-75.7495

For the particular initial state selected, the ΔV computed by the Level I process does not appear to change significantly. That is because the location of TEI-3 is purposefully selected to take advantage of the dynamical sensitivities in the four body problem (Sun-Earth-Moon-S/C). Thus, a small change in velocity can bridge the gap in the terminal state. At the same time, however, convergence via a Level I process is not guaranteed over the entire lunar cycle or for all constraint combinations, particularly as the quality of the initial guess degrades. For instance, a case where the altitude, longitude, and flight path angle are targeted via a Level I process, for the particular initial state employed in this example, fails to converge. This is attributed to the fact that the longitude is a time varying position constraint and the flight path angle constrains the attitude of the incoming velocity. Whenever a time dependent quantity related to position is targeted at the same time a velocity orientation quantity is targeted, it is difficult and sometimes impossible to converge on a solution via a linear Level I process. The relation between these quantities is simply too sensitive. This isn't necessarily a flaw of the targeter, it is the nature of the dynamics. A targeter searches the neighboring phase space for solutions that meet the desired criteria. If those solutions do not exist in that neighborhood, a Level I corrections process may experience difficulties in identifying a solution. Difficulties also arise when a solution that meets the entry state exists but at a significantly higher cost than the available fuel budget allows. For these reasons, a Level I process is not suitable for targeting in a highly constrained problem, though it may be suitable in simpler cases.[10] A two-level targeter, however, can easily remedy these deficiencies, as detailed in the following sections. It also significantly reduces the computational overhead in contrast to optimization schemes.

3. *Example 3: Level I Corrections via Finite Burns*

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. In identifying finite burn arcs, it is necessary to consider an augmented dynamical model. Specifically, since mass flow rate and thrust direction are constant over a segment, the dynamical model is augmented with the following additional equations,

$$\begin{aligned} \dot{m}_k &= -\dot{m}_{g_k} && (17) \\ \ddot{m}_{g_k} &= 0 && (18) \\ \dot{\bar{u}}_{k-1} &= \bar{0}, && (19) \end{aligned}$$

where m_k, $\dot{m}_{g_k}$, and $\bar{u}_k$ represent the spacecraft mass, the propellant flow rate, and the thrust direction associated with patch point k, respectively. For the present development, it is assumed that $\dot{m}_{g_k}$ and $\bar{u}_k$ are constant over the burn segment.

The burn and coast segments of the arc are integrated separately, and each segment is associated with an independently determined state transition matrix. The state at the end of the burn arc is denoted by the subscript T. The augmented variational equations for the burn segment are given by,

$$\begin{bmatrix} \delta\bar{R}_T - \bar{V}_T^- \delta t_T \\ \delta\bar{V}_T^- - \bar{a}_T^- \delta t_T \\ \delta m_T^- + \dot{m}_{g_T}^- \delta t_T \\ \delta\dot{m}_{g_T}^- - \ddot{m}_{g_T}^- \delta t_T \\ \delta\bar{u}_T^- - \dot{\bar{u}}_T^- \delta t_T \end{bmatrix} = \tag{20}$$

$$\begin{bmatrix} A_{T,k-1} & B_{T,k-1} & E_{T,k-1} & F_{T,k-1} & G_{T,k-1} \\ C_{T,k-1} & D_{T,k-1} & H_{T,k-1} & I_{T,k-1} & J_{T,k-1} \\ K_{T,k-1} & L_{T,k-1} & M_{T,k-1} & N_{T,k-1} & O_{T,k-1} \\ P_{T,k-1} & Q_{T,k-1} & R_{T,k-1} & S_{T,k-1} & T_{T,k-1} \\ U_{T,k-1} & V_{T,k-1} & W_{T,k-1} & X_{T,k-1} & Y_{T,k-1} \end{bmatrix} \begin{bmatrix} \delta\bar{R}_{k-1} - \bar{V}_{k-1}^+ \delta t_{k-1} \\ \delta\bar{V}_{k-1}^+ - \bar{a}_{k-1}^+ \delta t_{k-1} \\ \delta m_{k-1}^+ + \dot{m}_{g_{k-1}}^+ \delta t_{k-1} \\ \delta\dot{m}_{g_{k-1}}^+ - \ddot{m}_{g_{k-1}}^+ \delta t_{k-1} \\ \delta\bar{u}_{k-1}^+ - \dot{\bar{u}}_{k-1}^+ \delta t_{k-1} \end{bmatrix}$$

Similarly, the equations associated with the subsequent coast segment are determined as,

$$\begin{bmatrix} \delta\bar{R}_k - \bar{V}_k^- \delta t_k \\ \delta\bar{V}_k^- - \bar{a}_k^- \delta t_k \end{bmatrix} = \begin{bmatrix} A_{k,T} & B_{k,T} \\ C_{k,T} & D_{k,T} \end{bmatrix} \begin{bmatrix} \delta\bar{R}_T - \bar{V}_T^+ \delta t_T \\ \delta\bar{V}_T^+ - \bar{a}_T^+ \delta t_T \end{bmatrix} \tag{21}$$

As was true of the impulsive Level I targeter, the goal here is to identify a relation between the control variables and the targets. In this particular example, the target is the terminal position vector at point k. The control variables are the thrust direction ($\bar{u}_{k-1} = \bar{u}_k$), the time where the burn ends (t_T), and the time where the segment terminates (t_k). Thus, the finite burn Level I process seeks to satisfy three constraints with five control variables. A minimum norm solution is once again employed to determine the smallest changes in the control variables, relative to the initial guess, that meet the desired targets. The algebraic details involved in the determination of the finite burn Level I process are omitted for brevity. However, the manipulation of Equation (20) and Equation (21) is similar in structure of the methodology employed in the impulsive targeter.

Aside from the modified variational equations, the fundamental difference between the finite burn scheme and the impulsive targeter is in the assumptions employed. First, in Equation (20), it is assumed that the burn subsegment duration and the thrust direction are the control variables. Of course, the thrust direction is constant during the burn, but the direction can change with each successive iteration of the algorithm. In addition, the mass flow rate over the segment and the initial mass, position, velocity, and time are held fixed. Conversely, for the coast subsegment, it is assumed that $\delta t_T = 0$. That is because the integration of each segments proceeds independently from the other. Thus, once the thrust segment is determined, the terminal time of that subsegment is fixed. Also, it is assumed that no velocity discontinuity exists at the junction between the two subsegments. Subsequently, the initial velocity along the coast segment is equal to the terminal velocity of the burn segment. The key, then, is to determine a relation between the variation in the terminal position vector, $\bar{R}_k$, and the variable parameters associated with the burn arc.

Once these assumptions are implemented into Equation (20) and Equation (21), the variational equation for the finite burn Level I process reduces to

$$\delta\bar{R}_k = \left[\begin{array}{cc} (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & (A_{k,T}\bar{V}_T^+ + B_{k,T}\bar{a}_T^+) \end{array}\right] \left[\begin{array}{c} \delta\bar{u}_{k-1}^+ \\ \delta t_T \end{array}\right] \tag{22}$$

Once again, a minimum norm solution is employed to obtain the smallest changes in the control variable set that meet the specified goal,

$$\left[\begin{array}{c} \delta\bar{u}_{k-1}^+ \\ \delta t_T \end{array}\right] = \tilde{M_{fb}}^T (\tilde{M_{fb}}\tilde{M_{fb}}^T)^{-1}\delta\bar{R}_k \tag{23}$$

where

$$\tilde{M_{fb}} = \left[\begin{array}{cc} (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & (A_{k,T}\bar{V}_T^+ + B_{k,T}\bar{a}_T^+) \end{array}\right] \tag{24}$$

Initially, the direction and duration of the burn are approximated by performing a single impulsive Level I iteration, and using the resulting ΔV, combined with the rocket equation, to estimate the unknown thrust vector parameter. The Level I process described here is the first component of a modified Level II process that incorporates finite burns and realistic engine models. The success of the combined process is demonstrated later in this document.

B. Level II Process

If the startup arc, and subsequently the patch states, are not associated with a natural arc in the dynamical model employed, the Level I process can lead to a series of interior velocity discontinuities. The constraint vector in a Level II process includes these discontinuities as constraints. Aside from any entry constraints specified, the Level II process seeks to drive all constraint errors to zero. The level-two correction, then, suggests changes in the positions and times of all the patch states along the trajectory in order to achieve a solution that is continuous in velocity and satisfies any other path constraints specified. Basically, the two-level corrector explores the neighborhood of the startup arc for solutions that meet the specified constraint values. As is true of any constraint, the two-level process requires a linearized representation of the variation in the constraint as a function of the control parameters. The development of the relevant expressions is the subject of the following sections.

1. Impulsive Level II Targeter

In the traditional approach,[6,8] the control parameters are the positions and times of the patch points. The variational equation for the velocity continuity constraints at the interior patch states is given by,

$$\begin{array}{c} \delta\Delta\bar{V}_k = \Delta\bar{V}_k^* - \Delta\bar{V}_k = \left(-\frac{\partial\bar{V}_k^-}{\partial\bar{R}_{k-1}}\right)\delta\bar{R}_{k-1} + \left(-\frac{\partial\bar{V}_k^-}{\partial t_{k-1}}\right)\delta t_{k-1} + \left(\frac{\partial\bar{V}_k^+}{\partial\bar{R}_k} - \frac{\partial\bar{V}_k^-}{\partial\bar{R}_k}\right)\delta\bar{R}_k \\ + \left(\frac{\partial\bar{V}_k^+}{\partial t_k} - \frac{\partial\bar{V}_k^-}{\partial t_k}\right)\delta t_k + \left(\frac{\partial\bar{V}_k^+}{\partial\bar{R}_{k+1}}\right)\delta\bar{R}_{k+1} + \left(\frac{\partial\bar{V}_k^+}{\partial t_{k+1}}\right)\delta t_{k+1} \end{array} . \tag{25}$$

The determination of each of the partial derivatives in Equation (25) is accomplished through a finite difference approach. The results are presented by Marchand, Howell, and Wilson,[6] but are summarized in Table 3.

$\frac{\partial\bar{V}_k^-}{\partial\bar{R}_{k-1}} = B_{k-1,k}^{-1}$	$\frac{\partial\bar{V}_k^-}{\partial t_{k-1}} = -B_{k-1,k}^{-1}\bar{V}_{k-1}^+$
$\frac{\partial\bar{V}_k^-}{\partial\bar{R}_k} = -B_{k-1,k}^{-1}A_{k-1,k}$	$\frac{\partial\bar{V}_k^-}{\partial t_k} = \left(\bar{a}_k^- - D_{k,k-1}B_{k,k-1}^{-1}\bar{V}_k^-\right)$
$\frac{\partial\bar{V}_k^+}{\partial\bar{R}_k} = -B_{k+1,k}^{-1}A_{k+1,k}$	$\frac{\partial\bar{V}_k^+}{\partial t_k} = \left(\bar{a}_k^+ - D_{k,k+1}B_{k,k+1}^{-1}\bar{V}_k^+\right)$
$\frac{\partial\bar{V}_k^+}{\partial\bar{R}_{k+1}} = B_{k+1,k}^{-1}$	$\frac{\partial\bar{V}_k^+}{\partial t_{k+1}} = -B_{k+1,k}^{-1}\bar{V}_{k+1}^-$

Table 3. Partial derivatives of velocity relative to patch state control variables

For the Level II velocity continuity constraints, the partial derivatives of the velocity discontinuity at the k^{th} point are determined by substituting the above partials into the expressions in Equation (25). The resulting partial derivatives are summarized in Table 4.

$\frac{\partial \Delta \bar{V}_k}{\partial \bar{R}_{k-1}} = -B^{-1}_{k-1,k}$	$\frac{\partial \Delta \bar{V}_k}{\partial \bar{R}_k} = B^{-1}_{k-1,k} A_{k-1,k} - B^{-1}_{k+1,k} A_{k+1,k}$	$\frac{\partial \Delta \bar{V}_k}{\partial \bar{R}_{k+1}} = B^{-1}_{k+1,k}$
$\frac{\partial \Delta \bar{V}_k}{\partial t_{k-1}} = B^{-1}_{k-1,k} \bar{V}^{+}_{k-1}$	$\frac{\partial \Delta \bar{V}_k}{\partial t_k} = \bar{a}^{+}_k - \bar{a}^{-}_k + D_{k,k-1} B^{-1}_{k,k-1} \bar{V}^{-}_k - D_{k,k+1} B^{-1}_{k,k+1} \bar{V}^{+}_k$ or $\frac{\partial \Delta \bar{V}_k}{\partial t_k} = \bar{a}^{+}_k - \bar{a}^{-}_k - B^{-1}_{k-1,k} A_{k-1,k} \bar{V}^{-}_k + B^{-1}_{k+1,k} A_{k+1,k} \bar{V}^{+}_k$	$\frac{\partial \Delta \bar{V}_k}{\partial t_{k+1}} = -B^{-1}_{k+1,k} \bar{V}^{-}_{k+1}$

Table 4. Summary of partial derivatives for Level II:velocity constraints at interior patch points

The traditional statement[6,8] of the Level II step is subsequently summarized as

$$\delta \Delta \bar{V}_k = \Delta \bar{V}^*_k - \Delta \bar{V}_k = \underbrace{\begin{bmatrix} \frac{\partial \Delta \bar{V}_k}{\partial \bar{R}_{k-1}} & \frac{\partial \Delta \bar{V}_k}{\partial t_{k-1}} & \frac{\partial \Delta \bar{V}_k}{\partial \bar{R}_k} & \frac{\partial \Delta \bar{V}_k}{\partial t_k} & \frac{\partial \Delta \bar{V}_k}{\partial \bar{R}_{k+1}} & \frac{\partial \Delta \bar{V}_k}{\partial t_{k+1}} \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} \delta \bar{R}_{k-1} \\ \delta t_{k-1} \\ \delta \bar{R}_k \\ \delta t_k \\ \delta \bar{R}_{k+1} \\ \delta t_{k+1} \end{bmatrix}}_{\bar{b}} . \tag{26}$$

In Equation (26), the matrix M, containing all of the partial derivatives in Table 4, is termed the State Relationship Matrix (SRM) and $\bar{b}$ denotes the vector of variations in position and time. In this case, since velocity discontinuities exist at the interior points, Equation (26) is applied to each interior patch state. For a trajectory with N patch states, this leads to $N-2$ vector equations of the form outlined in Equation (26). These can be assembled into an augmented system of vector equations where the vector $\delta \Delta \bar{V}$ includes all interior $\Delta \bar{V}$ variations, $\bar{b}$ includes the position vectors and times of all N patch states, and M becomes an augmented SRM whose rows consist of the partial derivatives of the constraint with respect to the control variables in $\bar{b}$.

In a well-posed problem, the system is underdetermined; that is, there are more control variables than target quantities. Since the correction is based on a linearized dynamical model, a minimum norm solution is selected to minimize the requested changes in the control variables,

$$\bar{b} = M^T \left(M M^T \right)^{-1} \delta \Delta \bar{V}_k. \tag{27}$$

This assumption is best suited to preserve the accuracy of a linear corrections process. The results from Equation (27) suggest the smallest changes to the position and time of patch states $k-1$, k, and $k+1$ that meet the desired interior velocity continuity constraint at k. These changes are applied in the nonlinear system and the Level I iteration is repeated to achieve position continuity. Of course, the changes suggested by Equation (27) are linearized corrections. Implementation in the nonlinear model is unlikely to yield exactly the desired results. Thus, the combined Level I and Level II process becomes an augmented iterative scheme. The updates are applied until the constraints are met.

Of course, a Level II process can incorporate any constraint, α_{kj}, along a trajectory if that constraint is formulated as a function of the position, velocity, or time of any patch state along the trajectory. It is also possible to formulate constraints that depend on multiple patch states. When the constraint depends only on the time and state associated with one patch state, the dependencies may be of the form,

$$\alpha_{kj} = \alpha_{kj} \left(\bar{R}_k, \bar{V}^{+}_k, \bar{V}^{-}_k, t_k \right), \tag{28}$$

where,

$$\bar{V}^{+}_k = \bar{V}^{+}_k \left(\bar{R}_k, t_k, \bar{R}_{k+1}, t_{k+1} \right), \tag{29}$$

and

$$\bar{V}^{-}_k = \bar{V}^{-}_k \left(\bar{R}_{k-1}, t_{k-1}, \bar{R}_k, t_k \right), \tag{30}$$

represent the outgoing and incoming velocity vectors, respectively. The scalar α_{kj} denotes the j^{th} constraint on the k^{th} patch state. However, it is important to note that this is NOT a requirement in a Level II process. That is, a constraint can depend on ANY combination of patch states and times simultaneously, as demonstrated later in this document. In general, formulating a constraint that depends on the state at multiple patch points requires that the variational equation be adjusted to include all the necessary control variable dependencies.

For the traditional case[4–6] formulated in Equation (28), the variational equation may be summarized as follows,

$$
\begin{aligned}
\delta\alpha_{kj} = {} & \left(\frac{\partial\alpha_{kj}}{\partial\bar{V}_k^-}\frac{\partial\bar{V}_k^-}{\partial\bar{R}_{k-1}}\right)\delta\bar{R}_{k-1} + \left(\frac{\partial\alpha_{kj}}{\partial\bar{V}_k^-}\frac{\partial\bar{V}_k^-}{\partial t_{k-1}}\right)\delta t_{k-1} \\
& + \left(\frac{\partial\alpha_{kj}}{\partial\bar{R}_k} + \frac{\partial\alpha_{kj}}{\partial\bar{V}_k^+}\frac{\partial\bar{V}_k^+}{\delta\bar{R}_k} + \frac{\partial\alpha_{kj}}{\partial\bar{V}_k^-}\frac{\partial\bar{V}_k^-}{\delta\bar{R}_k}\right)\delta\bar{R}_k + \left(\frac{\partial\alpha_{kj}}{\partial t_k} + \frac{\partial\alpha_{kj}}{\partial\bar{V}_k^+}\frac{\partial\bar{V}_k^+}{\partial t_k} + \frac{\partial\alpha_{kj}}{\partial\bar{V}_k^-}\frac{\partial\bar{V}_k^-}{\partial t_k}\right)\delta t_k \\
& + \left(\frac{\partial\alpha_{kj}}{\partial\bar{V}_k^+}\frac{\partial\bar{V}_k^+}{\partial\bar{R}_{k+1}}\right)\delta\bar{R}_{k+1} + \left(\frac{\partial\alpha_{kj}}{\partial\bar{V}_k^+}\frac{\partial\bar{V}_k^+}{\partial t_{k+1}}\right)\delta t_{k+1},
\end{aligned}
\tag{31}
$$

where the partial derivatives are evaluated along the current path. The partial derivatives of α_{kj}, highlighted in red in Equation (31), are ONLY nonzero if the constraint equation depends explicitly on those variables. The remaining partial derivatives are known to be functions of the state transition matrix,[6] and are summarized in Table 3. Since the partials in Table 3 are independent of the constraint, whenever a new constraint is formulated it is only necessary to identify the partial derivatives associated with an explicit dependence on the position, time, or velocity associated with the constrained patch state.

2. Finite Burn Level II Targeter

In the event a contingency scenario is considered, the lower thrust auxiliary engines may be employed to perform the TEI maneuvers. This can lead to longer burn times that render the impulsive assumption of the traditional two-level targeter inadequate. To account for this contingency, the two-level targeter is modified to allow for the incorporation of finite burn arcs. In identifying a Level II process that incorporates realistic engine models, many formulations were considered. The results of these preliminary investigations ultimately lead to a modified Level II formulation that is not conceptually dissimilar from the impulsive burn scheme. In both cases, the only control variables employed are the positions and times of the patch states. Thus, the linearized model in Equation (31) is still applicable, though the partial derivatives of the $\Delta\bar{V}_k$ continuity constraints are different. The thrust parameters are otherwise held constant during the Level II correction. Another similarity between the two processes is that a set of equations that relate the velocity discontinuities between segments to the control parameters is necessary. However, identifying these relations is far more complex in the finite burn case because of the interdependencies between terminal variables at point k and the burn arc that terminates between points $k-1$ and k. This difficulty arises from the fact that the location of the spacecraft at the time the burn is terminated is not considered a patch state. Complexity is also increased due to considerations regarding adjacent segments that involve finite burns or coast arcs and the associated chronological order of events. This mainly affects whether or not a velocity discontinuity exists between segments and how the dependencies are defined. Since the derivation of the finite burn Level II equations is complex and involved, the details of the derivation are omitted from this document. However, the partial derivatives of $\delta\bar{V}_k^+$ are summarized in Equation (32),

$$
\begin{aligned}
\frac{\partial\Delta\bar{V}_k}{\partial\bar{R}_{k-1}} &= -(C_{k,T}A_{T,k-1} + D_{k,T}C_{T,k-1}) - \tilde{Z}(A_{k,T}A_{T,k-1} + B_{k,T}C_{T,k-1}) \\
\frac{\partial\Delta\bar{V}_k}{\partial t_{k-1}} &= -([(C_{k,T}E_{T,k-1} + D_{k,T}H_{T,k-1}) - \tilde{Z}(A_{k,T}E_{T,k-1} + B_{k,T}H_{T,k-1})]\dot{m}_{g_{k-1}}^+ \\
&\quad - [(C_{k,T}B_{T,k-1} + D_{k,T}D_{T,k-1}) - \tilde{Z}(A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1})]\bar{a}_{k-1}^+ \\
&\quad - [(C_{k,T}A_{T,k-1} + D_{k,T}C_{T,k-1}) - \tilde{Z}(A_{k,T}A_{T,k-1} + B_{k,T}C_{T,k-1})]\bar{V}_{k-1}^+) \\
\frac{\partial\Delta\bar{V}_k}{\partial\bar{R}_k} &= -B_{k+1,k}^{-1}A_{k+1,k} - \tilde{Z} \\
\frac{\partial\Delta\bar{V}_k}{\partial t_k} &= B_{k+1,k}^{-1}A_{k+1,k}\bar{V}_k^+ + \bar{a}_k^+ - (\bar{a}_k^- - \tilde{Z}\bar{V}_k^-) \\
\frac{\partial\Delta\bar{V}_k}{\partial\bar{R}_{k+1}} &= B_{k+1,k}^{-1} \\
\frac{\partial\Delta\bar{V}_k}{\partial t_{k+1}} &= -B_{k+1,k}^{-1}\bar{V}_{k+1}^-,
\end{aligned}
\tag{32}
$$

where

$$\tilde{Z} = \left[\begin{array}{ccc} (C_{k,T}B_{T,k-1} + D_{k,T}D_{T,k-1}) & (C_{k,T}G_{T,k-1} + D_{k,T}J_{T,k-1}) & D_{k,T}(\bar{a}_T^- - \bar{a}_T^+) \end{array} \right] Z^T(ZZ^T)^{-1},$$

$$Z = \left[\begin{array}{ccc} (A_{k,T}B_{T,k-1} + B_{k,T}D_{T,k-1}) & (A_{k,T}G_{T,k-1} + B_{k,T}J_{T,k-1}) & B_{k,T}(\bar{a}_T^- - \bar{a}_T^+) \end{array} \right].$$

A minimum norm solution is still employed in solving Equation (26).

IV. Trajectory Constraints

At entry interface, Orion seeks to target up to five quantities: altitude, flight path angle, latitude, longitude, and flight path azimuth. Furthermore, the total ΔV is constrained to satisfy the available fuel budget. Other constraints may designate, for instance, that TEI-2 and TEI-3 take place at an apse, where $\bar{R} \cdot \bar{V} = 0$. The partial derivatives associated with altitude and flight path angle were previously determined during the design of the Genesis mission.[4,5] Furthermore, since the Earth is assumed to be a spherical body, the latitude constraint partials are the same as the declination constraint partials previously formulated.[4,5] While the Genesis trajectory design was, in many ways, more complex and elegant due to the three body interactions, it was also simpler in other respects. For instance, aside from the entry date/time, none of the constraints imposed on the Genesis trajectory were explicitly dependent on time. In contrast, the longitude and flight path azimuth constraints are both measured relative to a rotating coordinate system. This makes the generation of suitable startup arcs difficult for Orion as well, though not nearly as complex as the process required in generating startup solutions for the Genesis mission.

Genesis required accurate initial arcs due to the dynamical sensitivity associated with the region near the libration points. The Orion spacecraft is subject to these sensitivities to a lesser extent, but it also requires an accurate initial guess to minimize the computational time, and address the time dependent constraints without violating the fuel budget. Orion must also be able to return at any time, during abort scenarios, whereas the Genesis return date was planned years in advanced. Thus, autonomy is required in both the generation of a suitable startup arc and the targeting process. The two-level corrections process employed in the design of the Genesis trajectory was never intended for onboard determination. The algorithm was simply designed as an analysts tool. In multi-body regimes, the success of this approach is strongly dependent on the quality of the startup arc. While third body effects are a critical consideration for the Orion return trajectory, their impact is localized to the region where TEI-3 takes place. The motivation for placing TEI-3 in this region is to meet all the desired constraints without violating the fuel budget. Thus, in addition to the longitude and flight path azimuth constraints, a total mission cost constraint is presented and successfully implemented.

Preliminary studies indicate that the most critical feature of a suitable startup arc, for Orion, is a terminal state that meets the entry constraints targeted, even if the startup arc is discontinuous in position elsewhere. It is also important that the startup arc itself does not significantly exceed the fuel budget. These errors do not preclude the targeter from converging, but they do slow the process down. Of course, since longitude and flight path azimuth are time dependent constraints, it is difficult to identify suitable startup arcs. Identifying an initial guess autonomously, in a computationally efficient manner, without over-tasking the onboard computer, is a difficult task on its own and outside the scope of the present investigation. This requires a level of autonomy that is not presently available. An initial methodology for the autonomous generation of startup arcs is employed for the examples presented in this study. However, this methodology can only guarantee the entry altitude, latitude, and longitude. As such, flight path angle and azimuth can take on any values and subsequently overtask the targeter due to an inadequate initial guess. It is important to note that the need for an accurate initial guess is not necessarily because the targeter requires one to converge. With proper norm control and autonomous schemes for monitoring convergence, the two-level targeter is very robust in spite of infeasible initial arcs. However, an infeasible arc does require increased computational overhead, an undesirable characteristic for onboard determination.

During an abort scenario, the onboard targeter must at least achieve the entry altitude and flight path angle, within the available budget. This particular task is relatively effortless for the two-level targeter over the lunar cycle. However, in targeting the remaining constraints, the time dependence can present difficulties if the startup arc does not adequately represent the desired path. This is the subject of future studies.

The constraint formulations employed in this study are summarized in Equations (33)-(37),

$$\begin{aligned}
\alpha_{k1} &= \cos\lambda_k - \cos\lambda_{des}, && (33)\\
\alpha_{k2} &= \sin\delta_k - \sin\delta_{des}, && (34)\\
\alpha_{k3} &= \sin\gamma_k - \sin\gamma_{des}, && (35)\\
\alpha_{k4} &= \sin\chi_k - \sin\chi_{des}, && (36)\\
\alpha_{k5} &= h - h_{des}, && (37)
\end{aligned}$$

where h_k denotes the altitude at entry, γ_k the flight path angle, χ_k the flight path azimuth, λ_k the longitude, and δ_k the latitude. Naturally, these equations represent one of many ways to formulate the same constraint. Some formulations are more numerically well behaved than others. Angular constraints can be particularly cumbersome because of quadrant ambiguities and potential singularities. However, the set selected above is suitable for the current application.

A. Coordinate Frames and Entry Angles

The flight path azimuth, one of the entry constraints of interest, is determined relative to a site fixed up-east-north frame, fixed on the rotating Earth. The illustration in Figure 3 depicts the relation between the Earth rotating frame, defined by unit vectors $\hat{x}_E - \hat{y}_E - \hat{z}_E$, the up-east-north unit vectors, $\hat{x}_L - \hat{y}_L - \hat{z}_L$, and the inertial unit vectors, $\hat{x}_G - \hat{y}_G - \hat{z}_G$. Relative to the inertial frame, the right ascension, θ_k, and declination, δ_k, of the spacecraft are determined as

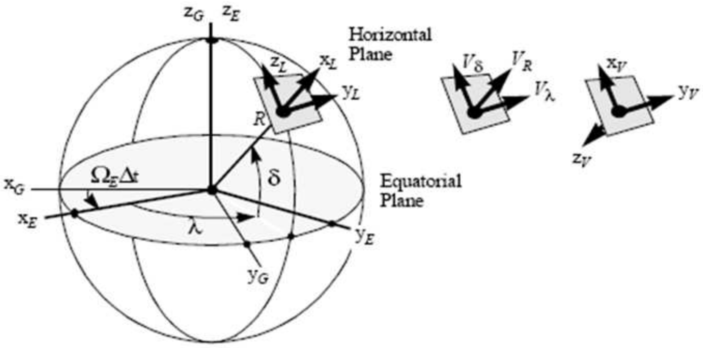


Figure 3. Flight Path Angle and Azimuth

$$\tan\theta_k = \frac{\bar{R}_k^T \hat{y}_G}{\bar{R}_k^T \hat{x}_G}, \tag{38}$$

$$\sin\delta_k = \frac{\bar{R}_k^T \hat{z}_G}{|\bar{R}_k|}. \tag{39}$$

Note that the declination also corresponds to the latitude of the spacecraft for a spherical Earth, as previously mentioned. The spherical Earth assumption also implies that the longitude is related to the right ascension as follows

$$\theta_k = \theta_G + \lambda_k = (\theta_{g0} + \omega_e (t_k - t_0)) + \lambda_k \tag{40}$$

where θ_G represents the location of the prime meridian relative to the Earth-centered inertial x-axis, $\hat{x}_G$. The time t_0 is associated with the 0^{th} hour (midnight) on the day of entry, and θ_{g0} represents the right ascension of the prime meridian at that time. The algorithm employed in approximating θ_{g0} is that listed by Vallado.[11]

The rotational rate of the Earth is denoted by ω_e. Thus, the spacecraft longitude at entry is determined to be

$$\lambda_k = \tan^{-1}\left(\frac{\bar{R}_k^T \hat{y}_G}{\bar{R}_k^T \hat{x}_G}\right) - \theta_{g0} - \omega_e \left(t_k - t_0\right). \tag{41}$$

Another quantity of interest, from Figure 3 and Figure 4, is the entry flight path angle, determined as

$$\sin\gamma_k = \frac{\bar{R}_k^T \bar{V}_k^-}{\left|\bar{R}_k\right| \left|\bar{V}_k^-\right|}. \tag{42}$$

Note that this computation is simple as it does not require knowledge of the coordinate frame explicitly, a simplifying fact that is not applicable to the flight path azimuth.

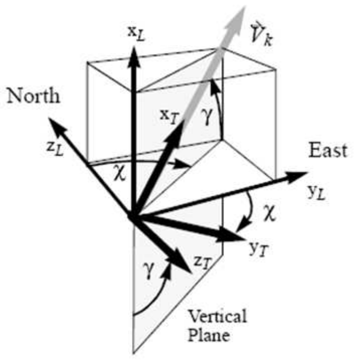


Figure 4. Definitions of Flight Path Azimuth and Flight Path Angle

To identify an expression for the flight path azimuth, consider an alternate representation of the spacecraft velocity in terms of the up-east-north frame. For notational simplicity, let $\hat{x}_L = \hat{r}_k$ represent the radial unit vector (up), $\hat{y}_L = \hat{e}_k$ denote the east direction, and $\hat{z}_L = \hat{n}_k$ represent the unit vector due north along the line of longitude. The mathematical definitions of these unit vectors may be summarized as follows,

$$\hat{r}_k = \hat{r}_k\left(\bar{R}_k\right) = \frac{\bar{R}_k}{R_k}, \tag{43}$$

$$\hat{e}_k = \hat{e}_k\left(\bar{R}_k\right) = \left(\frac{\hat{z}_G \times \hat{r}_k}{\left|\hat{z}_G \times \hat{r}_k\right|}\right) = \left(\frac{\hat{z}_G \times \bar{R}_k}{\left|\hat{z}_G \times \bar{R}_k\right|}\right), \tag{44}$$

$$\hat{n}_k = \hat{n}_k\left(\bar{R}_k\right) = \hat{r}_k \times \left(\frac{\hat{z}_G \times \bar{R}_k}{\left|\hat{z}_G \times \bar{R}_k\right|}\right) = \frac{1}{\left|\bar{R}_k\right| \left|\hat{z}_G \times \bar{R}_k\right|}\left(\bar{R}_k \times \left(\hat{z}_G \times \bar{R}_k\right)\right). \tag{45}$$

where $\hat{z}_G$ represents the inertial axis normal to the Earth's equator. Note, while these unit vectors are clearly time varying, they are not explicit functions of time. They are also independent of the spacecraft velocity. The only explicit dependence apparent is on the position vector, $\bar{R}_k$. Given these definitions, the velocity of the spacecraft – at the k^{th} patch state – may be alternatively represented as follows,

$$\bar{V}_k^- = \left(\bar{V}_k^- \cdot \hat{r}_k\right)\hat{r}_k + \left(\bar{V}_k^- \cdot \hat{e}_k\right)\hat{e}_k + \left(\bar{V}_k^- \cdot \hat{n}_k\right)\hat{n}_k. \tag{46}$$

Using Equation (46) and Figure 4, the flight path azimuth, χ_k, may be expressed as

$$\tan\chi_k = \frac{\bar{V}_k^- \cdot \hat{e}_k}{\bar{V}_k^- \cdot \hat{n}_k} = \frac{\left(\bar{V}_k^-\right)^T \hat{e}_k}{\left(\bar{V}_k^-\right)^T \hat{n}_k}. \tag{47}$$

Note that the tangent of the flight path azimuth angle can be geometrically interpreted from the illustration below.

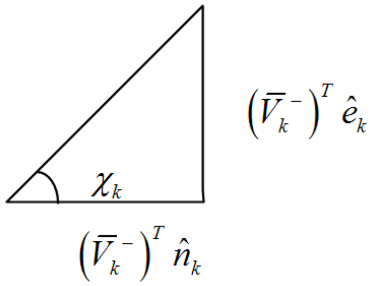

$\left(\bar{V}_k^-\right)^T \hat{e}_k$
χ_k
$\left(\bar{V}_k^-\right)^T \hat{n}_k$

Figure 5. Flight Path Azimuth Functional Dependence

This geometrical arrangement is consistent with the formulation in Equation (47). Subsequently, it is also possible to express the flight path azimuth in terms of a cosine or a sine function,

$$\sin\chi_k = \frac{\left(V_k^-\right)^T \hat{e}_k}{\sqrt{\left(\left(V_k^-\right)^T \hat{n}_k\right)^2 + \left(\left(V_k^-\right)^T \hat{e}_k\right)^2}}. \tag{48}$$

B. Flight Path Azimuth

Consider the flight path azimuth constraint, α_{k4}. Note that the only variable element of Equation (36) is the sine of the true flight path azimuth. From Equation (48), and Equations (43)-(45), it is apparent that the flight path azimuth depends explicitly on the velocity and on the position vector associated with the patch state, but no explicit dependence with respect to time exists. Once Equation (48) is substituted into Equation (36), the partial derivative of the sine constraint with respect to the terminal position vector is given by,

$$\frac{\partial\sin\chi_k}{\partial\bar{R}_k} = \frac{\left\{\begin{array}{l} \left(\left(\bar{V}_k^-\right)^T \frac{\partial\hat{e}_k}{\partial\bar{R}_k}\right)\sqrt{\left(\left(\bar{V}_k^-\right)^T \hat{e}_k\right)^2 + \left(\left(\bar{V}_k^-\right)^T \hat{n}_k\right)^2} \\ -\frac{\left(\left(\left(\bar{V}_k^-\right)^T \hat{e}_k\right)\left(\left(\bar{V}_k^-\right)^T \frac{\partial\hat{e}_k}{\partial\bar{R}_k}\right) + \left(\left(\bar{V}_k^-\right)^T \hat{n}_k\right)\left(\left(\bar{V}_k^-\right)^T \frac{\partial\hat{n}_k}{\partial\bar{R}_k}\right)\right)\left(\left(\bar{V}_k^-\right)^T \hat{e}_k\right)}{\sqrt{\left(\left(\bar{V}_k^-\right)^T \hat{e}_k\right)^2 + \left(\left(\bar{V}_k^-\right)^T \hat{n}_k\right)^2}} \end{array}\right\}}{\left(\left(\bar{V}_k^-\right)^T \hat{e}_k\right)^2 + \left(\left(\bar{V}_k^-\right)^T \hat{n}_k\right)^2}. \tag{49}$$

Similarly, the partial derivative with respect to the velocity vector is determined as

$$\frac{\partial\left(\sin\chi_k\right)}{\partial\bar{V}_k^-} = \frac{\left(\hat{e}_k^T\right)\sqrt{\left(\left(V_k^-\right)^T \hat{n}_k\right)^2 + \left(\left(V_k^-\right)^T \hat{e}_k\right)^2} - \left[\left(V_k^-\right)^T \hat{e}_k\right]\left[\frac{\left(\left(V_k^-\right)^T \hat{n}_k\right)\hat{n}_k^T + \left(\left(V_k^-\right)^T \hat{e}_k\right)\hat{e}_k^T}{\sqrt{\left(\left(V_k^-\right)^T \hat{n}_k\right)^2 + \left(\left(V_k^-\right)^T \hat{e}_k\right)^2}}\right]}{\left(\left(V_k^-\right)^T \hat{n}_k\right)^2 + \left(\left(V_k^-\right)^T \hat{e}_k\right)^2}. \tag{50}$$

These are the only non-zero partial derivatives of those highlighted in red in Equation (44). Note that the partial derivative in Equation (49) is dependent on the availability of the partial derivatives of Equations (44) and (45) with respect to $\bar{R}_k$. In this case, it is helpful to rewrite these expressions in a form convenient for differentiation. For instance, if

$$Z_\times = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{51}$$

then

$$\hat{z}_G \times \bar{R}_k = Z_\times \bar{R}_k. \tag{52}$$

Equation (45) is further simplified through direct application of the vector triple product identity,

$$\bar{R}_k \times \left(\hat{z}_G \times \bar{R}_k\right) = \hat{z}_G\left(\bar{R}_k^T \bar{R}_k\right) - \bar{R}_k\left(\bar{R}_k^T \hat{z}_G\right). \tag{53}$$

Subsequently, and after extensive algebraic manipulations, the partial derivatives of $\hat{e}_k$ and $\hat{n}_k$ may be summarized as,

$$\frac{\partial \hat{e}_k}{\partial \bar{R}_k} = \left(I - \frac{1}{\left| Z_\times \bar{R}_k \right|^2} \left(Z_\times \bar{R}_k \right) \left(Z_\times \bar{R}_k \right)^T \right) \frac{Z_\times}{\left| Z_\times \bar{R} \right|}. \tag{54}$$

and

$$\frac{\partial \hat{n}}{\partial \bar{R}_k} = \frac{1}{\left|\bar{R}_k\right|^2 \left| Z_\times \bar{R}_k \right|^2} \left\{ \begin{array}{l} \left|\bar{R}_k\right| \left| Z_\times \bar{R}_k \right| \left[2\hat{z}_G \bar{R}_k^T - \bar{R}_k \hat{z}_G^T - \left(\bar{R}_k^T \hat{z}_G \right) I \right] \\ - \left(\hat{z}_G \left(\bar{R}_k^T \bar{R}_k \right) - \bar{R}_k \left(\bar{R}_k^T \hat{z}_G \right) \right) \left[\left(\frac{\left| Z_\times \bar{R}_k \right|}{\left|\bar{R}_k\right|} \right) \bar{R}_k^T + \frac{\left|\bar{R}_k\right|}{\left| Z_\times \bar{R}_k \right|} \left(Z_\times \bar{R}_k \right)^T Z_\times \right] \end{array} \right\} \tag{55}$$

Equation (55) may then be substituted into Equation (49). The resulting expression is subsequently employed in the Taylor series expansion listed in Equation (44) during the second step of the two-level process.

C. Flight Path Angle

The flight path angle constraint, given by Equations (35) and (42), reveals an explicit dependence on the position and incoming velocity of the patch state, but no explicit dependence on time. This constraint was previously determined[5] and the associated partials are only summarized here,

$$\frac{\partial \alpha_3}{\partial \bar{R}_k} = \frac{\left(\bar{V}_k^-\right)^T}{\left|\bar{R}_k\right| \left|\bar{V}_k^-\right|} - \sin\gamma_k \frac{\bar{R}_k^T}{\left|\bar{R}_k\right|^2}, \tag{56}$$

$$\frac{\partial \alpha_3}{\partial \bar{V}_k^-} = \frac{\bar{R}_k^T}{\left|\bar{R}_k\right| \left|\bar{V}_k^-\right|} - \sin\gamma_k \frac{\left(\bar{V}_k^-\right)^T}{\left|\bar{V}_k^-\right|^2}. \tag{57}$$

D. Latitude

The latitude constraint, given by Equations (34) and (39), depends explicitly on the position vector associated with the k^{th} patch state. Clearly, no explicit dependence on time or velocity exists in this case. As such, the unidentified partial derivatives of the latitude constraint are equal to the explicit partial derivatives previously identified[5] for the declination constraint,

$$\frac{\partial \alpha_2}{\partial \bar{R}_k} = \frac{1}{\left|\bar{R}_k\right|} \left[\hat{z}_G^T - \frac{\bar{R}_k^T}{\left|\bar{R}_k\right|} \sin\delta_k \right]. \tag{58}$$

E. Longitude

The longitude constraint given by Equations (33) and (41) reveal an explicit dependence on the position vector and time associated with the k^{th} patch state. Thus, the partial derivatives of the longitude with respect to position and time are required before evaluation of Equation (31) is possible. Once again, multiple formulations are available. The longitude, as an angle, can be constrained. Alternatively, the sine or cosine, or some other function of the longitude can be constrained as well. The selected formulation affects the explicit partials. As such, only one formulation is presented here, though multiple formulations have been tested to date. It is important to note that angular constraints can be particularly difficult to converge because one angle accepts an infinite number of representations due to the $2n\pi$ ambiguity. Ultimately, the convergence properties hinge on the formulation employed. One should choose formulations that avoid singularities in the partial derivatives, and quadrant ambiguities in the range of values of interest. The formulation presented here pertains to the cosine of the longitude, though other formulations were developed and tested.

Note that the constraint in Equation (33) may be restated as follows,

$$\cos\lambda_k = \cos\left(\theta_k - \theta_{g0} - \omega_e \Delta t_k\right) = \cos\theta_k \cos\left(\theta_{g0} + \omega_e \Delta t\right) + \sin\theta_k \sin\left(\theta_{g0} + \omega_e \Delta t\right). \tag{59}$$

To simplify the determination of the partial derivatives, consider the illustration below

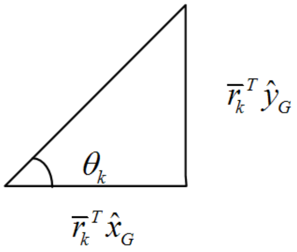


Figure 6. Right Ascension Functional Dependence

where θ_k is as defined in Equation (38). Subsequently,

$$\cos\theta_k = \frac{\bar{R}_k^T \hat{x}_G}{\sqrt{\left(\bar{R}_k^T \hat{x}_G\right)^2 + \left(\bar{R}_k^T \hat{y}_G\right)^2}}, \tag{60}$$

and

$$\sin\theta_k = \frac{\bar{R}_k^T \hat{y}_G}{\sqrt{\left(\bar{R}_k^T \hat{x}_G\right)^2 + \left(\bar{R}_k^T \hat{y}_G\right)^2}}. \tag{61}$$

Using these definitions, and Equation (59), the constraint partials are determined as

$$\frac{\partial \cos\lambda_k}{\partial \bar{R}_k} = \frac{\partial \cos\theta_k}{\partial \bar{R}_k} \cos\left(\theta_{g0} + \omega_e \Delta t\right) + \frac{\partial \sin\theta_k}{\partial \bar{R}_k} \sin\left(\theta_{g0} + \omega_e \Delta t\right), \tag{62}$$

$$\frac{\partial \cos\lambda_k}{\partial t_k} = \omega_e \sin\left(\theta_k - \theta_{g0} - \omega_e \Delta t\right). \tag{63}$$

where

$$\frac{\partial \cos\theta_k}{\partial \bar{R}_k} = \frac{\hat{x}_G^T \sqrt{\left(\bar{R}_k^T \hat{x}_G\right)^2 + \left(\bar{R}_k^T \hat{y}_G\right)^2} - \left(\bar{R}_k^T \hat{x}_G\right) \frac{\left(\left(\bar{R}_k^T \hat{x}_G\right)\hat{x}_G^T + \left(\bar{R}_k^T \hat{y}_G\right)\hat{y}_G^T\right)}{\sqrt{\left(\bar{R}_k^T \hat{x}_G\right)^2 + \left(\bar{R}_k^T \hat{y}_G\right)^2}}}{\left(\bar{R}_k^T \hat{x}_G\right)^2 + \left(\bar{R}_k^T \hat{y}_G\right)^2}, \tag{64}$$

and

$$\frac{\partial \sin\theta_k}{\partial \bar{R}_k} = \frac{\hat{y}_G^T \sqrt{\left(\bar{R}_k^T \hat{x}_G\right)^2 + \left(\bar{R}_k^T \hat{y}_G\right)^2} - \left(\bar{R}_k^T \hat{y}_G\right) \frac{\left(\left(\bar{R}_k^T \hat{x}_G\right)\hat{x}_G^T + \left(\bar{R}_k^T \hat{y}_G\right)\hat{y}_G^T\right)}{\sqrt{\left(\bar{R}_k^T \hat{x}_G\right)^2 + \left(\bar{R}_k^T \hat{y}_G\right)^2}}}{\left(\bar{R}_k^T \hat{x}_G\right)^2 + \left(\bar{R}_k^T \hat{y}_G\right)^2}. \tag{65}$$

The fundamental difference between the right ascension and longitude constraints is in the explicit dependence on time. The partial derivative of α_1 with respect to time is straightforward as it seeks to identify only an explicit dependence on time, and is ultimately determined to be

$$\frac{\partial \cos\lambda_k}{\partial t_k} = -\omega_e \cos\theta_k \sin\left(\theta_{g0} + \omega_e \Delta t\right) + \omega_e \sin\theta_k \cos\left(\theta_{g0} + \omega_e \Delta t\right). \tag{66}$$

F. Total Mission Cost Constraint

A two-level corrector traditionally seeks to drive the interior velocity discontinuities to zero.[4–6,8] However, a series of non-zero maneuvers are naturally required to achieve the entry constraints in a non-natural arc. In a two-level corrector, the analyst can specify constraints on individual maneuvers.[4–6] A maneuver constraint may be formulated as follows,

$$\alpha_{\text{-}k5} = \left|\bar{V}_k^+ - \bar{V}_k^-\right| - \Delta V_{des} = \Delta V_k - \Delta V_{des} = \sqrt{\Delta\bar{V}_k^T \Delta\bar{V}_k} - \Delta V_{des}. \tag{67}$$

Note that this constraint depends explicitly on both the incoming and outgoing velocity vectors associated with the patch state, no dependence on position or time is present. Thus, the partial derivatives of the constraint relative to the velocity before and after the discontinuity are simply given by

$$\frac{\partial \alpha_{k5}}{\partial \bar{V}_k^+} = \frac{1}{2}\left(\Delta\bar{V}_k^T \Delta\bar{V}_k\right)^{-\frac{1}{2}}\left(2\Delta\bar{V}_k^T\right)\frac{\partial \Delta\bar{V}_k}{\partial \bar{V}_k^+} = \frac{\Delta\bar{V}_k^T}{\left|\Delta\bar{V}_k\right|}, \tag{68}$$

$$\frac{\partial \alpha_{k5}}{\partial \bar{V}_k^-} = \frac{1}{2}\left(\Delta\bar{V}_k^T \Delta\bar{V}_k\right)^{-\frac{1}{2}}\left(2\Delta\bar{V}_k^T\right)\frac{\partial \Delta\bar{V}_k}{\partial \bar{V}_k^-} = -\frac{\Delta\bar{V}_k^T}{\left|\Delta\bar{V}_k\right|}. \tag{69}$$

Traditionally, the maneuvers are considered equality constraints and are "active" at all times. If an inequality constraint is desired, however, the user need only remove the appropriate rows of the state relationship matrix before performing the Level II correction. If, at a later point, the constraint becomes active again, then the rows associated with the above partial derivatives can be introduced back into the state relationship matrix, SRM.

In the case of Orion's three TEI maneuver strategy, constraining the magnitude of individual maneuvers is not as advantageous as it is to constrain the sum of the maneuvers. The goal of this particular approach is to ensure that the entry constraints are met while the total propulsive cost does not exceed the available ΔV. This approach gives the two-level process much needed flexibility to place the maneuvers wherever the dynamics dictate it is most advantageous, as long as the sum does not exceed the allowable budget. In this case, the constraint would be of the form,

$$\alpha_5 = \Delta V_1 + \Delta V_2 + \Delta V_3 \tag{70}$$

Note that the variation of the total ΔV may be expressed as the sum of the variations,

$$\delta\left(\Delta V_1 + \Delta V_2 + \Delta V_3\right) = \delta\Delta V_1 + \delta\Delta V_2 + \delta\Delta V_3. \tag{71}$$

Thus, the variation of the sum is the sum of the variations. Since each maneuver is associated with a different vector variational equation, summing these variational equations yields and expression for the sum of the variations in Equation (71). The resulting expression may subsequently be used to constrain the sum of all maneuvers without constraining individual maneuvers. Since each patch state contributes 12 control variables (three positions and three time elements), the total cost constraint for a 3-burn TEI sequence can depend on up to 36 control variables, a far more desirable and successful approach than that in Equation (67). This particular constraint is useful in extending the two-level corrector into an autonomous algorithm because it is less constrained and, subsequently, less susceptible to the quality of the initial guess.

A total cost constraint is also formulated for the finite burn case, and successfully implemented in the last example presented in the results section. To formulate the burn maneuver constraint, it is necessary to determine the partial derivatives of the magnitude of $\Delta\bar{V}_k$ with respect to the Level II control variables. From the rocket equation, ΔV_k is given by

$$\Delta V_k = -I_{sp} g_0 \ln(1 - \frac{\dot{m}_{g_k} \Delta t_{burn}}{m_k}), \tag{72}$$

where $\Delta t_{burn} = t_T - t_k$. The determination of the partial derivatives of the constraint partials is as complex as the equations for the finite burn scheme. Thus, the derivations are omitted from this document. However, it is important to note that the process of identifying the mission cost constraint partials remains unchanged in the finite burn case. That is, while the partial derivatives themselves change, it is still true that the variation of the sum is the sum of the variations.

Thus, once the individual ΔV equations are identified, the results are added together to generate the total mission cost constraint. This implies the total cost depends on the positions and times of at least 12 patch states, in the three-burn scenario. Since the details of the theory behind the finite burn targeter are the subject of a different study, the present document focuses only on contrasting the performance of the impulsive and the finite burn schemes for applications to Orion.

V. Results

The two level targeter is applied here for various startup arcs over the lunar cycle, beginning on December 9, 2014, 00:00:00 UTC. The target entry constraints are: 121.9 km altitude, 134.5456 degrees longitude, and -19.2041 degrees latitude. The startup arc is determined using the methodology previously discussed, where the trajectory from TEI-3 through entry is determined via a Level I corrections process that targets altitude, latitude, and longitude. As previously mentioned, a Level I process alone does not guarantee convergence under these conditions over the lunar cycle. Thus, not all the startup arcs provided are adequate in this case since the initial guess algorithm employs a Level I process from TEI-3 to entry. Table 5 shows the initial error in each constraint for selected days over the lunar cycle. Note the abnormally large error in the second entry of the table. This corresponds to one case where the Level I process was unable to converge on the desired target. Nonetheless, this initial guess is still employed in a full two-level targeter, both with and without a total ΔV constraint.

Table 5. Initial Constraint Errors Over Lunar Cycle

Cycle Day	Init. ΔV (km/s)	Alt. Error (km)	Lat. Error (deg)	Long. Error (deg)
0	1.3323	3.079	-1.300e-6	0.082
7	2.5404	16609.407	24.915	-92.460
14.5	1.3680	3.080	-1.517e-7	-0.0518
21	2.4042	3.080	-2.910e-7	-0.256
27	1.3556	3.080	1.810e-7	-0.009

One way to assess the success of the two-level targeting process for onboard determination is to compare its performance to that of an optimizer. The optimal results presented here target the entry altitude, latitude, and longitude. Later examples consider additional constraints. Both the two-level targeter and the optimizer employ the same initial guess. The two-level targeter is implemented and executed in MATLAB. The optimizer selected is based on a nonlinear programming algorithm and is implemented on a UNIX platform in FORTRAN. The theoretical details of the optimization process are outside the scope of the present study and are, as such, omitted from the discussion. It is important to note, however, that computational efficiency is reduced at least by one order of magnitude for programs executed inside the MATLAB environment in contrast to UNIX based FORTRAN source. MATLAB was specifically selected as the environment of choice in this study due to the availability of autocode capabilities that transition MATLAB source to flight code. Though the homogeneity of the computational platforms is not preserved in this comparison, the following examples demonstrate the computational superiority of the two-level process in spite of the computational deficiencies of the selected platform.

A. Example 1: Targeting Altitude, Latitude, and Longitude Over the Lunar Cycle

Figure 7 represents a comparison between the total ΔV from the initial guess algorithm, the two-level targeter with and without total cost constraints, and an optimizer that seeks to minimize the mission cost while meeting the specified constraints. Note that the solid lines in this plot appear jagged because of the selected sampling rate over the lunar cycle and because the results are determined in the four body problem. However, the general trends are still observable from this figure.

The optimizer results correspond to the solid black line that exhibits the smallest costs, between 800 m/sec and 1.2 km/sec over the lunar cycle. The initial guess algorithm, one that employs a simple Level I correction from TEI-3 to entry, appears as the relatively cyclic solid blue line with the highest costs, ranging from 1.2 to 2.4 km/sec. The patch states corresponding to these startup arcs are employed in both the optimizer and the targeter (with and without cost constraints).

For the targeter solutions, samples were selected at approximately seven day intervals along the lunar cycle. The red circles correspond to the targeter solution without the total cost constraint. The blue crosses are associated with the targeter solutions that are subject to a total cost constraint of 1.2 km/sec. For each of the cases selected, the two-level targeter is able to converge on the desired entry conditions, even for the second case listed in Table 5. The individual TEI maneuvers for both the cost constrained and unconstrained two-level targeter are listed in Table 6 and Table 7, respectively.

Once again, it is not always possible to converge on a solution when the initial guess exhibits significant errors, such as those associated with day 7 along the cycle in Figure 7. Thus, identifying a methodology for consistently generating reasonably accurate initial arcs, one that is both simple and computationally efficient, is a subject of future study. Of course, this is true of any algorithm or scheme that seeks to meet all the demands of the Orion vehicle in the ephemeris model, since no exact solutions are available in this regime. However, the most desirable feature of an onboard targeting algorithm, for applications to Orion, is one that is both numerically robust and minimizes computational overhead. Robustness is achieved by incorporating checks and balances that ensure monotonic convergence of the process. The second step of the process is not iterative. However, each time this step is invoked, a sequential Level I process is executed to achieve position continuity. This is where the majority of the computational effort is spent. Thus, identifying ways of speeding up and enhancing the convergence of the Level I process is the key to enhancing the overall computational efficiency of the two-level targeter. Still, even in contrast to an optimizer, the two-level targeter offers a significant performance improvement. That is because the targeter does not seek to optimize, it merely seeks to identify a solution that meets the desired constraints in the vicinity of the initial guess.

It is important to note that the solutions identified by either a targeter or a non-evolutionary optimization algorithm can change depending on the initial guess provided. All non-evolutionary optimization processes search for optimal solutions in the vicinity of a startup arc. Thus, a gradient based optimization scheme can only identify a locally optimal solution in the vicinity of the startup arc. However, the neighborhood around that arc may foster several locally optimal solutions.There is no systematic way to guarantee that the solution identified is consistent under all conditions. The same is true of a targeter but to a lesser extent. Because the targeting scheme employs a minimum norm solution, the requested changes at any point during the process are the smallest possible that achieve the desired constraints. Thus, in a sense, the two-level targeter is more conservative in its search than an optimizer, a factor that positively influences the computational efficiency of the algorithm. If the main concern is to accomplish the goals of the mission in the least amount of time, and within the available fuel budget, a targeting scheme is a more suitable option than an optimizer.

Table 6. ΔV Values for Constrained TLC Algorithm (km/s)

Cycle Day	TEI-1	TEI-2	TEI-3	Total
0	0.6075	0.0594	0.5330	1.20
7	0.6055	0.1082	0.4863	1.20
14.5	0.6076	0.0730	0.5195	1.20
21	0.6076	0.1075	0.4849	1.20
27	0.6076	0.0525	0.5398	1.20

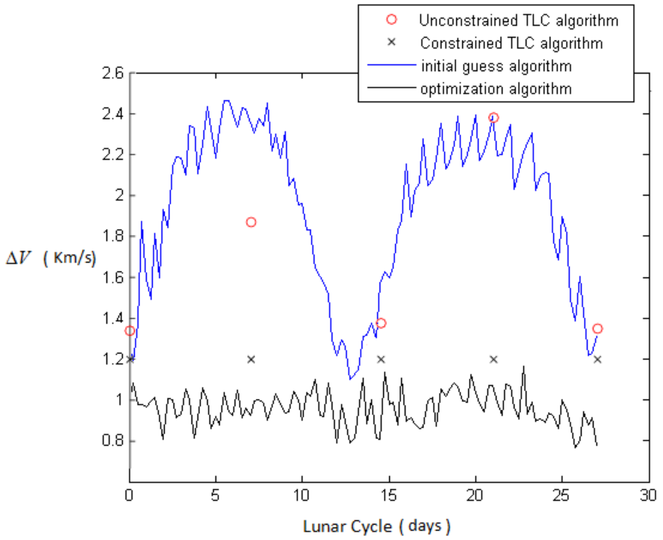


Figure 7. ΔV Values Over Lunar Cycle

Table 7. ΔV Values for Unconstrained TLC Algorithm (km/s)

Cycle Day	TEI-1	TEI-2	TEI-3	Total
0	0.6078	0.0580	0.6716	1.3374
7	0.6098	0.1190	1.1396	1.8684
14.5	0.6081	0.0599	0.7091	1.3771
21	0.6080	0.0244	1.7496	2.3820
27	0.6078	0.0579	0.6887	1.3544

B. Example 2: Targeting Altitude, Latitude, Longitude, and Flight Path Angle

In this example, a three-burn TEI transfer, subject to altitude, latitude, longitude, and flight path angle constraints is presented. The optimal transfer identified, based on the initial guess, requires approximately 860 m/s of impulsive ΔV. The targeting algorithm identifies a solution that meets the same set of constraints at a total cost of 900 m/sec. Figure 8 illustrate a superposition of the resulting optimal and targeter solutions.

For the above example, the optimal solution is achieved in 14.4 minutes. In contrast, the two-level targeter solution requires approximately 7 minutes, a 50 percent improvement in spite of the significant discrepancy in computational platforms. If the computational platforms are evenly matched, this percent improvement would be even more significant.

C. Example 3: Targeting Altitude, Latitude, Longitude, Flight Path Azimuth and Flight Path Angle

In this example, all five entry constraints are targeted: altitude, latitude, longitude, flight path angle, and flight path azimuth. Figure 9 illustrates the corresponding targeter and optimizer solutions, superimposed in the Earth Centered Inertial (ECI). Since all five constraints are targeted, but the initial guess process employed in the previous examples targets only altitude, latitude and longitude, a large discrepancy exists between the flight path angle and azimuth of the startup arc relative to the desired solution. Thus, in order to identify a more suitable initial guess, the level-one process previously employed from TEI-3 to entry interface is replaced with a process. The total time required to complete this computation is 364 seconds, a total of 25 iterations, without a total cost constraint. The bulk of this computation is attributed to the

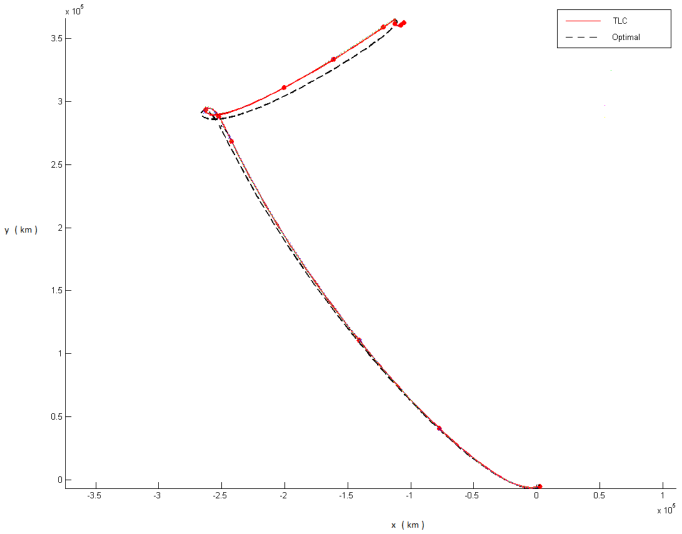


Figure 8. Comparison 1: Targeting vs. Optimal (ECI)

Level I component of the process. As previously mentioned, the time dependent longitude constraint can conflict with constraints associated with the attitude of the entry velocity. Quantities such as the flight path angle and azimuth define the shape of the incoming trajectory. The resulting ΔV associated with the initial guess is 2.3 km/sec.

The initial guess now satisfies all entry constraints, but not the total cost constraint. Thus, the process is repeated from end-to-end, i.e. from Lunar departure to Earth entry, with a total cost constraint of 1.35 km/sec. With this constraint, the end-to-end process requires 370 seconds of computation time. The total budget available for the return phase is 1.5 km/sec. Thus, 1.35 km/sec for the primary maneuvers is still a reasonable cost.

In the interest of identifying similar solutions, the same initial guess is employed in the optimization process. Furthermore, the terminal entry time is constrained to match the entry time previously identified by the two-level targeter. Thus, some disparity is expected between the two trajectories since the problems targeted slightly different scenarios. The optimal total cost is over 350 m/sec lower than the targeter solution in this case. In spite of the slight differences in scenarios, the solutions are still similar. The non-uniform motion seen at the beginning of the optimal transfer is indicative of a significantly delayed maneuver. That is, rather than departing immediately, as the targeter is designed to do, the optimizer determines it is best to wait in orbit before embarking on the final return. This is an artifact of the flight time constraint imposed to match the targeter's original result.

For this example, the computation time required by the optimization process is 18.3 minutes. As evidenced from the optimal solution in Figure 9, generating an initial guess in the vicinity of the optimal is not a simple task. However, this particular example clearly indicates the importance of a reasonable initial guess when dealing with time varying constraints and the coupling between longitude and the attitude of the incoming velocity vector. Given the same startup arc, the optimizer identifies a solution that waits in orbit for a significant amount of time before completing the TEI sequence. Basically, the process seeks a more suitable alignment to meet all the specified constraints. The targeter preserves the geometry of the initial guess and meets all specified constraints within the available budget, though the result is not intended to be fuel optimal. Furthermore, the departure time is purposefully held fixed in the targeting scheme, though that is not a necessary constraint. The optimizer, on the other hand, allows the departure time to vary in its search for the optimal alignment.

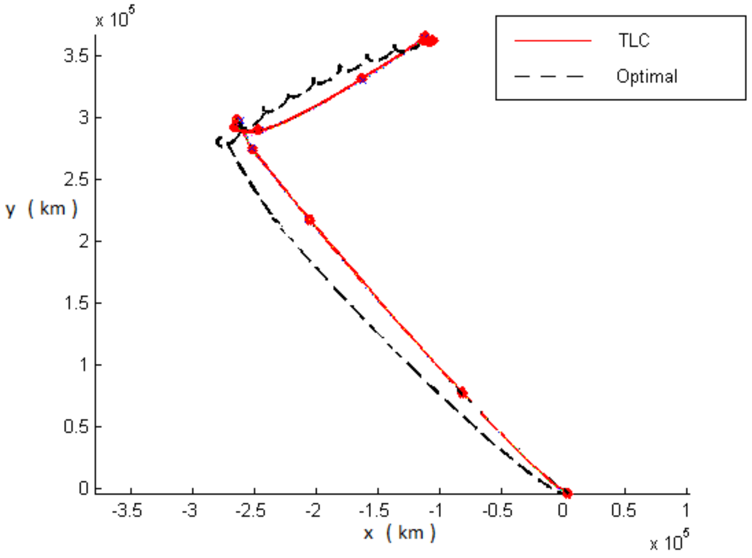


Figure 9. Comparison 2: Targeting vs. Optimal (ECI)

D. Example 4: Impulsive vs. Finite Burn Targeter

The following examples contrast the convergence properties of the impulsive and the finite burn targeting scheme when targeting the entry altitude, flight path angle, latitude, and longitude. A total cost constraint is implemented in both cases. The cost constraint requires that the total mission ΔV does not exceed 1.2 km/sec. The difference between the two examples is in the quality of the initial guess. The first example, in Figure 10, begins with a previously determined impulsive optimal solution. The second example, in Figure 11, employs the initial guess process described earlier in this document. In Figure 10, the combined finite burns lead to a mission ΔV of 1.1108 km/s. The trajectory converged in 4 iterations, using 152.1875 seconds of computation time. The second example, in Figure 11, is a four-burn trajectory instead of a three-burn trajectory. This solution converged in 5 iterations and 271.2344 seconds, with a total burn ΔV of 1.0731 km/s. The duration of each burn is given in Table 8.

Table 8. Individual Burn Durations

Burn	Figure 10	Figure 11
TEI-1	302.3305 s	324.5929 s
TEI-2	74.7421 s	27.2148 s
TEI-3	159.6147 s	118.6056 s
TEI-4	N/A	50.9205 s

VI. Optimal Trans-Earth Injection

The optimal Trans-Earth injection phase has been extensively studied with a variety of optimizers, including Copernicus, DIDO, and SOCS, among others. Though these numerical methods are not suitable for onboard determination, the insight gained from these earlier studies is useful in the development of a robust and numerically efficient autonomous targeting algorithm. Successfully targeting a specific entry state requires careful consideration of the timing and location of the maneuvers, particularly for constraints such as longitude and flight path azimuth.

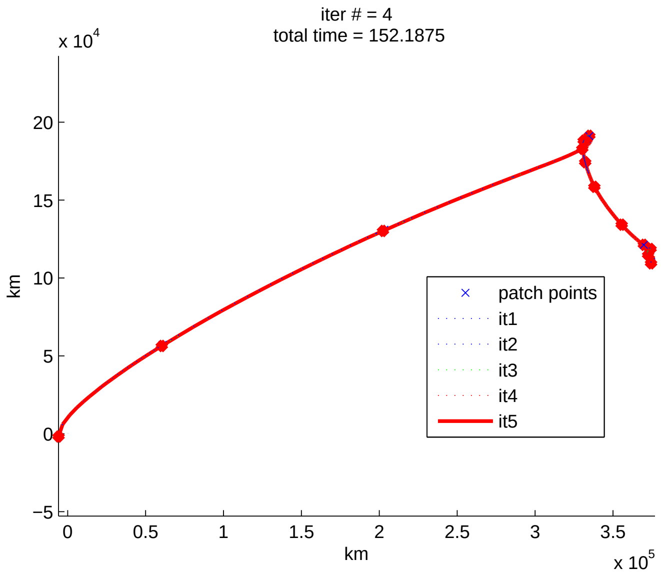


Figure 10. Orion Trajectory (ECI), Mike Weeks Initial Guess

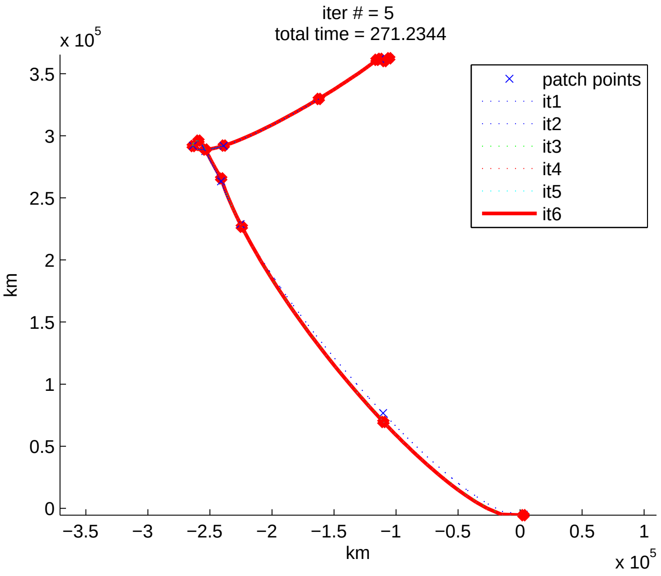
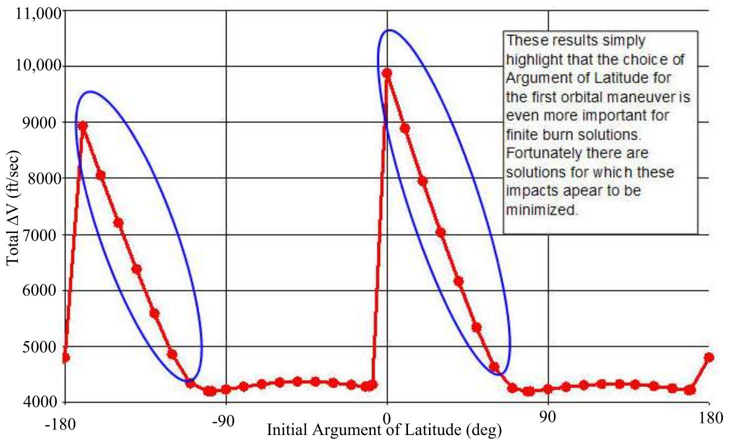


Figure 11. Orion Trajectory (ECI), Chad Smith Initial Guess

As previously mentioned, TEI-1 raises the apolune and TEI-2 executes the plane change. Then, TEI-3 takes place in a more dynamically sensitive region, where multi-body effects can significantly impact the subsequent path of the spacecraft. Executing TEI-3 in a dynamically sensitive region is helpful in targeting the desired entry state. However, TEI-3 alone cannot guarantee that all the desired constraints are met within the available fuel budget, specifically constraints such as longitude and flight path azimuth. To achieve a specific longitude, the startup arc must exhibit the overall character of the desired path. That is, the error in longitude cannot be excessively large. However, the incoming velocity attitude determines the flight path azimuth, and that may be at odds with the specified longitude depending on the geometry of the startup solution. Thus, identifying a suitable initial guess algorithm that addresses this coupling between the timing and the entry geometry is a difficult task on its own, and beyond the scope of the present study. However, it is important to note these difficulties because the success of the final injection maneuver ultimately depends on the timing and location of the first two maneuvers. Thus, an end-to-end targeting algorithm that addresses the coupled impact of these maneuvers is desired.

Preliminary optimization studies suggest that, for a given initial LLO, multiple local minimum TEI solutions exist. These depend on the initial argument of latitude, as seen in Figure 12. Similarly, the right ascension of the ascending node (RAAN) of the initial LLO also leads to multiple optimal TEI solutions, as seen from Figure 13.

Of course, the effect of these parametric sensitivities, and their subsequent effect on the convergence of various optimizers and algorithms, is beyond the scope of this investigation. The results above are significant to the present study because they indicate that the timing and geometry of the Earth return trajectory significantly impact the total mission cost and the availability of solutions that meet the desired constraints. Thus, the success of any iterative process, optimal or otherwise, is strongly dependent on the availability of suitable startup arcs.



These results simply highlight that the choice of Argument of Latitude for the first orbital maneuver is even more important for finite burn solutions. Fortunately there are solutions for which these impacts apear to be minimized.

Figure 12. TEI ΔV (ft/sec) vs. Argument of Latitude (deg)

It is interesting to note, after properly converting the displayed units, that the first peak in Figure 12, leading to a 2.7 km/sec ΔV, is roughly the same cost as the worst case provided by a level one process in Figure 7, about 2.4 km/sec. Even though the methods of acquiring the associated solutions are completely different, these results seem to indicate both methods explore the same neighborhood. The optimal solutions identified in these earlier studies can be of significant use in refining an initial guess algorithm that properly addresses the entry geometry constraints. Figure 13 also indicates that, given the proper alignment, the optimal solutions do not typically exceed a combined mission cost of 1.2 km/sec. This is an important piece of information in providing the targeter a reasonable cost constraint.

VII. Conclusions and Suggested Future Investigations

This investigation presents a series of modifications to the two-level targeting scheme that seek to transition the algorithm into a process suitable for onboard determination. The resulting algorithm is employed here for the Orion trans-Earth injection phase. Aside from the necessary constraint formulations, a total

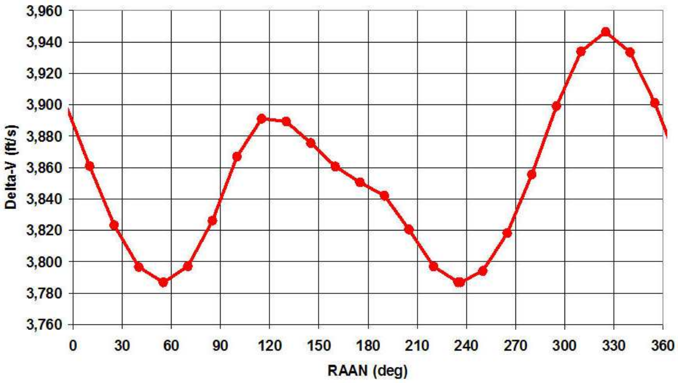


Figure 13. ΔV **(ft/sec) Sensitivity to Initial RAAN (deg)**

mission cost constraint is successfully implemented and tested. Also, a finite burn Level I and process is discussed and contrasted to the impulsive targeter. Additionally, a comparison between optimal solutions and targeter solutions is presented. A Level I process, even one that targets constraints rather than position vectors, is not suitable for point-to-point onboard determination for the Orion trajectory. That is because a Level I process offers no control over the size of the resulting maneuver and the convergence properties are extremely sensitive to the dynamical model and the amount of time elapsed between targets. An optimization scheme is also not suitable for onboard determination due to the increased computational time required and the unpredictability associated with the search for an optimal solution. A modified two-level targeter offers increased capabilities and performance over a Level I process, but at a greatly reduced computational cost when compared to an optimizer. While a two-level targeter does not seek to optimize the total cost, a new total cost constraint presented here ensures that the cost does not exceed the available budget. This constraint is available in both the impulsive and finite burn models presented. This leads to a targeter with unprecedented flexibility, even when the quality of the startup arc seems inadequate. Of course, a two-level targeter is based on a linearized representation of the dynamical model. As such, a suitable startup arc greatly enhances the computational time required to identify a trajectory that meets all the specified interior, initial, and terminal constraints. Thus, with a reasonably accurate initial guess, and suitable modifications and pre-defined logic, a two-level targeting scheme can be transitioned from an analysts tool into an algorithm suitable for autonomous onboard determination. This study represents the first step towards that goal.

References

[1] Crain, T., Condon, J., D'Souza, C., and Weeks, M., "Entry Interface Velocity Variation," CEV Flight Dynamics Presentation, May 16 2007.

[2] Weeks, M. W. and Crain, T., "Description of a TransEarth Midcourse Targeting Algorithm for CEV," Doc. No. FltDyn-CEV-07-012, Feb. 8th 2007.

[3] Howell, K., Barden, B., Wilson, R., and Lo, M., "Trajectory design using a dynamical systems approach with application to GENESIS," *Proceedings of the AAS/AIAA Astrodynamics Conference, Sun Valley, Idaho*, 1997, pp. pp. 1665–1684.

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

[5] Wilson, R. and Howell, K., "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.

[6] Marchand, B., Howell, K., and Wilson, R., "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.

[7] Bretscher, O., *Linear Algebra with Applications*, Prentice Hall, 2001.

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

[9] Bate, R., Mueller, D., and White, J., *Fundamentals of Astrodynamics*, Dover Publications, New York, 1971.

[10] Adamo, D., "Apollo 13 Trajectory Reconstruction Via State Transition Matrices," *30th ANNUAL AAS GUIDANCE AND CONTROL CONFERENCE*, No. AAS 07-330, 2007.

[11] Vallado, D., *Fundamentals of Astrodynamics and Applications*, McGraw-Hill, 1997.