(Preprint) AAS 25-844

ENHANCEMENTS TO SPACE SHUTTLE POWERED EXPLICIT GUIDANCE FOR PLANETARY ASCENT AND DESCENT

Bharat Mahajan* **and Gerald L. Condon†**

A number of enhancements to Space Shuttle Powered Explicit Guidance (PEG) algorithm are developed in this work that increases its validity and accuracy for various planetary ascent and descent trajectory problems. The specific PEG enhancements studied in this work include precise targeting of the line-of-apsides for elliptical orbit insertion, range and burn ignition time control for precision landing, and compensation for atmospheric drag. Additionally, a new general predictor-corrector formulation of PEG is presented that can compute effects due to any arbitrary force model on the guided trajectory by using numerical quadrature. Two different simulations for lunar and Mars descent and ascent vehicles are implemented to validate these PEG modifications.

INTRODUCTION

NASA's Space Shuttle used powered explicit guidance (PEG) algorithm for the onboard guidance of its exo-atmospheric nominal and abort ascent trajectories.[1] Its exo-atmospheric trajectory roughly started at 43 km altitude after jettisoning the two solid rocket boosters and afterwards, PEG was used to guide the Orbiter (while still connected to the external tank) to the target orbit using three Space Shuttle Main Engines (SSME). Following the separation of the external tank at approximately 120 km altitude, two Orbital Maneuvering System (OMS) engines were used to insert Orbiter into a 278 km altitude nominal parking orbit. During ascent, three different abort modes, Return-to-Launch-Site (RTLS), Abort-Once-Around (AOA), and Abort-to-Orbit (ATO), provided abort coverage during the entire ascent trajectory. The latter two abort modes have profiles similar to the nominal ascent while RTLS provided capability to immediately return to the primary launch site during the first stage and part of the second stage trajectory. Using PEG, Space Shuttle's onboard guidance software was designed to have a common set of routines that can satisfy many different nominal and abort profile requirements unlike the case of earlier Apollo missions where a separate program was specifically developed for each case.[2,3] Despite being developed in 1970s, PEG is the foundation of ascent and on-orbit guidance algorithms used by Orion and Space Launch System (SLS) programs, respectively.[4,5]

Shuttle PEG is a fuel-optimal (more precisely, propellant-optimal) closed-loop three degrees-of-freedom (DOF) explicit guidance algorithm that computes the thrust pointing vector to achieve a desired target orbit. In RTLS abort mode, it can also generate SSME throttling commands for returning the Orbiter to the primary launch site. The development of PEG was inspired by Iterative

*Design Engineer/Analyst, Odyssey Space Research LLC, Presently at Flight Mechanics and Trajectory Design Branch, NASA Johnson Space Center, 2101 E NASA Pkwy, Houston, TX 77058, USA.
†Senior Aerospace Engineer (Retired), Flight Mechanics and Trajectory Design Branch, NASA Johnson Space Center, 2101 E NASA Pkwy, Houston, TX 77058, USA.

Guidance Mode (IGM) algorithm that was used as an ascent guidance program for Saturn V launch vehicle in the Apollo program.[6] Similar to IGM, PEG uses simplified dynamics (flat Earth assumption) and is based on the linear tangent optimal guidance law with linearly-varying thrust direction during the flight.[7] Due to these simplifications and assumptions, both IGM and PEG typically have to be run repetitively at an appropriate frequency (Space Shuttle used 1.94 sec delay for PEG calls) to update the steering commands. However, unlike IGM, PEG uses a more accurate method for state prediction at the final time (referred to as time-to-go in PEG literature with the initial time set to zero), has a vector formulation, and uses a predictor-corrector strategy to estimate the costates of the associated optimal control problem. In the later years of PEG development, a more generalized version of PEG was also developed that used bilinear tangent form for the optimal thrust control law that is necessary for fixed downrange at the burnout time suitable for abort trajectory requirements.[8] The readers are referred to Reference 9 for detailed historical account of PEG development.

More recently, a number of enhancements and modifications has been made to the original Shuttle PEG algorithm for applying it to various planetary ascent and descent trajectory problems. The second author and his co-op (at the time) at NASA-Johnson Space Center developed a PEG-based lunar descent simulation to prove its utility for precision landing using thrust modulation and Time-of-Ignition (TIG) control.[10] Around the same time, Thomas Fill from Draper Laboratories also applied Shuttle PEG to the braking burn guidance problem in his lunar descent simulations[11] and later he developed a method for precision lunar landing for NASA's ALHAT program by modifying the PEG corrector algorithm by accommodating in it sensitivity of v_{go} (velocity-to-be-gained) to the downrange distance.[12] This approach for range control was inspired by the original RTLS mode of Shuttle PEG (see Reference 1). Other variations of PEG used by recent NASA programs include Orion's on-orbit guidance software for powered flight (referred to as OrbGuid),[4] SLS multi-phase version of ascent PEG with modifications for stability,[5] and the numerical predictor-corrector based PEG used by Human Landing System (HLS) for lunar braking burn.[13] SLS uses Shuttle PEG for core and upper stage (exo-atmospheric) ascent burns that is wrapped around by an outer loop logic to enable multi-phase (for instance, constant thrust or acceleration) trajectory burn capability and for managing burn times, ascent targets, mass flow rate, and engine parameters across all phases. SLS also uses a number of modifications in PEG to improve its stability for long burn arcs and these include scaling v_{go} in the corrector logic, limiting the tangent of thrust angle in the steering law, protection of sign reversal in thrust turning rate components, limiting the thrust angle rate, elevation limit constraint for inhibiting retrograde steering commands for low thrust-to-weight ratio, etc.[5] Orion also uses PEG predictor-corrector method in OrbGuid for computing burn duration and attitude commands during all of the on-obit burns.[14] In OrbGuid, many burn target options are added in PEG corrector to provide a uniform guidance framework for handling multiple flight phases and maneuver types including orbit insertion, rendezvous, deorbit, orbit raise, lunar transfer, earth-return burns, aborts, and externally-specified ΔV burns. OrbGuid wraps its internal PEG routine in an executive loop that cycles PEG at 1 hz during active guidance to update the steering commands and t_{go} based on the latest navigated vehicle state available.

In the same vein as SLS and Orion, NASA's HLS program has also presented new and unique challenges for onboard trajectory guidance algorithms. HLS lunar landing vehicle uses a restricted three-body orbit called as Near-Rectilinear Halo Orbit (NRHO) as a staging orbit for its lunar sortie mission.[15] After the lunar surface stay, the ascent from the lunar surface requires insertion into a target elliptical orbit with unique in-plane orientation for getting back to the NRHO. The conventional ascent PEG targets (the altitude, velocity magnitude, and flight-path angle) are not suitable

here because the target orbit in-plane orientation is not uniquely determined in this case. As a result, the achieved orbit may have an undesired orbit in-plane orientation. To attain a unique line of apsides, PEG must be modified to use a different set of target parameters.[16] When using PEG for NRHO insertion and departure burns, ignoring the third-body effects (e.g., Earth's gravitation) may introduce significant targeting errors especially for large burn arcs. The conventional PEG uses a linear thrust control law form that is optimal only when the downrange is a free variable. However, for planetary descent, precise range control and time-of-ignition is necessary to achieve precision landing. The Shuttle PEG and its recent variations are used for exo-atmospheric burns only. Compensating for atmospheric drag in PEG predictor algorithm has the potential of improving its ascent performance for Mars-like planets where the atmosphere is 'swollen' and extends further into the space (compared to Earth).

In the present paper, a number of enhancements to conventional PEG algorithm are developed that increases its validity and accuracy for various planetary ascent and descent trajectory guidance applications. These specific PEG enhancements studied in this work include precise targeting of the line of apsides for ascent burns, range and TIG control for precision landing, and compensation for atmospheric drag during ascent burns. Two different simulations for lunar and Mars vehicles are implemented to validate the performance of these enhancements. A general predictor-corrector formulation of PEG algorithm is also developed in this work that can accommodate any arbitrary force model using numerical quadrature for calculating more fuel-efficient guidance commands. It is shown how the conventional analytical PEG equations can be derived from this more general form of equations by employing various simplifying assumptions. There are many papers in the literature that provide derivations of PEG equations, however, most (if not all) are written from a practitioner's point of view with little justification given for the assumptions made in the derivation. Therefore, another objective of this study is to provide a modern derivation of PEG along with details on the assumptions and approximations employed by PEG to solve the optimal trajectory problem analytically using its predictor-corrector scheme.

PEG DERIVATION FROM FIRST PRINCIPLES

This section presents the derivation of PEG equations in their most general form, starting from first principles. The objective here is to provide a modern development of the PEG equations with a consistent notation and more importantly, clearly highlighting the approximations and assumption that are used in the conventional PEG algorithm. A novel generalized predictor-corrector formulation of PEG is also developed here that facilitates inclusion of any arbitrary force model using numerical quadrature for computing more fuel-optimal guidance solution.

Optimal Powered Descent/Ascent Trajectory Computation

The equations of motion (EOM) for a vehicle acted upon by natural forces such as gravity and drag along with thrust from a rocket engine are

$$\dot{\mathbf{r}} = \mathbf{v}, \tag{1}$$

$$\dot{\mathbf{v}} = \mathbf{g} + \frac{T}{m}\hat{\mathbf{u}}, \tag{2}$$

$$\dot{m} = -\frac{T}{c}, \tag{3}$$

where $\mathbf{r}$ and $\mathbf{v}$ denote position and velocity vectors, respectively; m is the wet mass, T is the applied thrust magnitude, $\hat{\mathbf{u}}$ is thrust direction (unit) vector, and c is the exhaust speed of the rocket engine. The vector quantity $\mathbf{g}$ represents the sum of all natural forces acting on the vehicle. It is assumed that thrust magnitude is limited to a maximum value $\mathcal{T}$. All the boldface symbols in this work represent vector quantities and upper case notation is used for matrices. The problem is to compute optimal thrust direction that minimizes the propellant usage and achieves a desired target state. If t_0 and t_f are the initial and final times, respectively, then the cost function in Meyer form is

$$J = -m(t_f). \tag{4}$$

The Hamiltonian of this system is

$$H = \boldsymbol{\lambda}_r^T \boldsymbol{v} + \boldsymbol{\lambda}_v^T \left(\boldsymbol{g} + \frac{T}{m} \boldsymbol{u} \right) - \lambda_m \frac{T}{c}, \tag{5}$$

where $\boldsymbol{\lambda}_r$, $\boldsymbol{\lambda}_v$, and λ_m are adjoint/costate variables. The optimal thrust direction is obtained by applying Pontryagin's principle, which gives the optimal $\hat{\boldsymbol{u}}$ for minimizing H as

$$\hat{\boldsymbol{u}} = -\frac{\boldsymbol{\lambda}_v}{\lambda_v}, \tag{6}$$

where λ_v is the magnitude of $\boldsymbol{\lambda}_v$. Substituting this result back in the equation for H results in

$$H = \boldsymbol{\lambda}_r^T \boldsymbol{v} + \boldsymbol{\lambda}_v^T \boldsymbol{g} + ST, \tag{7}$$

where

$$S = -\frac{\lambda_v}{m} - \frac{\lambda_m}{c}. \tag{8}$$

The optimal thrust magnitude that minimizes H is again obtained from Pontryagin's principle:

$$T = \begin{cases} 0 & \text{if } S > 0, \\ \mathcal{T} & \text{if } S < 0, \\ \tau \text{ where } 0 < \tau < \mathcal{T} & \text{if } S = 0. \end{cases} \tag{9}$$

This shows that the optimal control has bang-bang profile with thrust either off or on depending on the sign of S, which is for this reason is also called as the switching function. The condition $S = 0$ being true for a finite amount of time indicates the presence of singular arcs. For determining the sign of S along the optimal solution, the following costate equations are needed:

$$\dot{\boldsymbol{\lambda}}_r = -\frac{\partial H}{\partial \boldsymbol{r}} = -G_r^T \boldsymbol{\lambda}_v, \tag{10}$$

$$\dot{\boldsymbol{\lambda}}_v = -\frac{\partial H}{\partial \boldsymbol{v}} = -\boldsymbol{\lambda}_r - G_v^T \boldsymbol{\lambda}_v, \tag{11}$$

$$\dot{\lambda}_m = -\frac{T}{m^2} \lambda_v - \boldsymbol{g}_m^T \boldsymbol{\lambda}_v, \tag{12}$$

where

$$G_r = \frac{\partial \boldsymbol{g}_r}{\partial \boldsymbol{r}}, \qquad G_v = \frac{\partial \boldsymbol{g}_r}{\partial \boldsymbol{v}}, \qquad \boldsymbol{g}_m = \frac{\partial \boldsymbol{g}_r}{\partial m}. \tag{13}$$

If gravity is the only natural force considered then the last two terms ($G_{\boldsymbol{v}}$ and $\boldsymbol{g}_m$) are nil and the first term is the gravity gradient matrix. For a spherical central body, its expression is

$$G_r = \frac{\mu}{r^5}\left(3\boldsymbol{r}\boldsymbol{r}^T - r^2 I_3\right), \tag{14}$$

where I_3 is the identity matrix of dimension three and r is the magnitude of $\boldsymbol{r}$. The general boundary conditions at the final time are

$$\left(\phi_t + \boldsymbol{\nu}^T\boldsymbol{\theta}_t + H\right)|_{t_f}\mathrm{d}t_f + \left(\phi_{\boldsymbol{X}} + \boldsymbol{\nu}^T\boldsymbol{\theta}_{\boldsymbol{X}} - \boldsymbol{\lambda}^T\right)|_{t_f}\mathrm{d}\boldsymbol{X}(t_f) = 0, \tag{15}$$

where $\phi = -m(t_f)$, $\boldsymbol{\theta}$ is constraint vector at t_f, $\boldsymbol{X} = [\boldsymbol{r}, \boldsymbol{v}, m]^T$, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_r, \boldsymbol{\lambda}_v, \lambda_m]^T$, and $\boldsymbol{\nu} = [\boldsymbol{\nu}_r, \boldsymbol{\nu}_v]^T$ is a vector of constant Lagrange multipliers. It is immediately clear that $H(t_f) = 0$ and $\lambda_m(t_f) = -1$ because there are no constraints on t_f and $m(t_f)$.

The conventional PEG algorithm assumes constant gravity model (flat planet assumption) for costate (or primer vector) dynamics and as a result, $\boldsymbol{g}$ becomes a constant vector. The position and velocity costate equations can now be integrated analytically that shows $\boldsymbol{\lambda}_r$ is a constant vector and $\boldsymbol{\lambda}_v(t) = \boldsymbol{\lambda}_v(t_0) - \boldsymbol{\lambda}_r(t - t_0)$. The optimal thrust direction then assumes the form:

$$\hat{\boldsymbol{u}}(t) = -\frac{\boldsymbol{\lambda}_v(t_0) - \boldsymbol{\lambda}_r(t - t_0)}{||\boldsymbol{\lambda}_v(t_0) - \boldsymbol{\lambda}_r(t - t_0)||}. \tag{16}$$

For flat-planet assumption, it is easy to see that the singular arcs are not an optimal solutions for this problem. Consider

$$\dot{S} = -\frac{\dot{\lambda}_v}{m} = \frac{\boldsymbol{\lambda}_r^T\boldsymbol{\lambda}_v}{m\lambda_v}. \tag{17}$$

Then the conditions $S = \dot{S} = 0$ imply that $\boldsymbol{\lambda}_v$ remains constant on a singular arc. As a consequence, $\boldsymbol{\lambda}_r = 0$ is satisfied identically on the optimal solution, which contradicts the boundary conditions at t_f because $\boldsymbol{\lambda}_r(t_f)$ must be non-zero if there are any constraints specified on the final position at burnout. Space Shuttle's PEG implementation had two modes for choosing thrust magnitude: constant thrust and constant thrust acceleration. The latter was typically used after the constant thrust phase to limit the maximum thrust acceleration on the crew and vehicle. Henceforth, it will be assumed that the thrust or thrust acceleration is always set to its maximum value during the entire burn. Assuming free $\boldsymbol{r}(t_f)$ and $\boldsymbol{v}(t_f)$, the final costates can be derived from Eq. (15) to obtain

$$\boldsymbol{\lambda}_r^T(t_f) = \boldsymbol{\nu}_r^T\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{r}(t_f)}, \tag{18}$$

$$\boldsymbol{\lambda}_v^T(t_f) = \boldsymbol{\nu}_v^T\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}(t_f)}. \tag{19}$$

For different constraints on the final position and velocity, these final costates assume different values. If a reference frame is defined with its three axes aligned with the downrange, trajectory plane normal, and radial directions in this order then the thrust direction vector may be written as

$$\hat{\boldsymbol{u}} = \begin{bmatrix} \cos\beta\cos\alpha \\ \cos\beta\sin\alpha \\ \sin\beta \end{bmatrix}, \tag{20}$$

where α and β are the out-of-plane and in-plane thrust spherical angles, respectively, with the latter measured from the horizontal direction. From Eq. (16), the in-plane optimal thrust angle is given as

$$\tan\beta(t) = -\frac{\lambda_{v3}(t_0) - \lambda_{r3}(t-t_0)}{\lambda_{v1}(t_0) - \lambda_{r1}(t-t_0)}, \tag{21}$$

where the numeric subscript indicates the element number of the corresponding vector quantity. This expression gives the typical bilinear tangent form of the optimal thrust direction. For powered ascent, the downrange distance is typically free and therefore, $\lambda_{r1} = 0$ from Eq. (15). Setting λ_{r1} to zero in Eq. (21) gives the familiar linear tangent law that says the tangent of the optimal in-plane thrust angle is a linear function of time as shown:

$$\tan\beta(t) = \frac{\lambda_{r3}}{\lambda_{v1}(t_0)}(t-t_0) - \frac{\lambda_{v3}(t_0)}{\lambda_{v1}(t_0)}. \tag{22}$$

If the final constraints on the states or end conditions are simply of the form

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{r}(t_f) - \boldsymbol{r}_d \\ \boldsymbol{v}(t_f) - \boldsymbol{v}_d \end{bmatrix}, \tag{23}$$

where $\boldsymbol{r}_d$ and $\boldsymbol{v}_d$ are the desired position and velocity vectors at burnout time, respectively, then the final costates are found as $\boldsymbol{\lambda}_r = \boldsymbol{\nu}_r$ and $\boldsymbol{\lambda}_v(t_f) = \boldsymbol{\nu}_v$ by using Eq. (15). With these equations, a two-point boundary value problem (TPBVP) with seven coupled nonlinear equations in seven unknowns is obtained that can be solved to compute the optimal powered descent or ascent trajectory solution. This TPBVP can be characterized by specific constraints and unknowns as follows:

PEG TPBVP with $\boldsymbol{r}_d$ and $\boldsymbol{v}_d$ targets

Unknowns	$\left[t_f, \boldsymbol{\lambda}_r, \boldsymbol{\lambda}_v(t_0)\right]^T$
Constraints	$H(t_f) = 0, \boldsymbol{\theta} = 0$

It is possible to eliminate one unknown $\lambda_v(t_0)$ as it only acts as a scaling variable. For this, the thrust direction from Eq. (16) may be written as

$$\hat{\boldsymbol{u}}(t) = -\frac{\hat{\boldsymbol{\lambda}}_v(t_0) - \boldsymbol{\lambda}_r'(t-t_0)}{||\hat{\boldsymbol{\lambda}}_v(t_0) - \boldsymbol{\lambda}_r'(t-t_0)||}, \tag{24}$$

where hat notation represents unit vector and primed quantities represent scaling by $\lambda_v(t_0)$. Thus, the simplified TPBVP with only six unknowns is

Simplified PEG TPBVP with $\boldsymbol{r}_d$ and $\boldsymbol{v}_d$ targets

Unknowns	$\left[t_f, \boldsymbol{\lambda}'_r, \hat{\boldsymbol{\lambda}}_v(t_0)\right]^T$
Constraints	$\boldsymbol{\theta} = 0$

It is noted that the Hamiltonian constraint is not needed in this TPBVP because it only determines the scaling parameter. If the downrange distance is not fixed (typical for ascent guidance) then the number of unknowns and constraint equations both get further reduced by one because in that case, the component of $\boldsymbol{\lambda}'_r$ in the downrange direction is identically zero along the optimal trajectory.

PEG Target Conditions

Different position and velocity target conditions at burnout determine the final values of the costates. These conditions are need for costate propagation and integrating the equations of motion. Jaggers used various orbit targeting conditions to show how the final costate values are related to each other. These relations are also useful to determine simplifying conditions used in Shuttle's PEG implementation that facilitate analytic solution of TPBVP.[7] Here, various target orbit conditions are discussed and the corresponding conditions on the costates are derived.

Orbit Energy Target For targeting specific two-body orbit energy E_d, the boundary condition is

$$\boldsymbol{\theta} = \left[\frac{\boldsymbol{v}(t_f)^T\boldsymbol{v}(t_f)}{2} - \frac{\mu}{r(t_f)} - E_d\right]. \tag{25}$$

Substituting this in Eq. (18-19) gives the final costates:

$$\boldsymbol{\lambda}_r^T(t_f) = -\nu_E\, \boldsymbol{g}^T(t_f), \tag{26}$$
$$\boldsymbol{\lambda}_v^T(t_f) = \nu_E\, \boldsymbol{v}^T(t_f). \tag{27}$$

These equations show that the position and velocity costates are along the position and velocity vectors at the final time, respectively. As a result, the costate vectors will be orthogonal at burnout time when the target conditions specify a particular orbit energy and flight-path angle of 0°.

Circular Orbit Injection For insertion into a circular orbit, the constraint on the final state can be specified using

$$\boldsymbol{\theta} = \left[r(t_f)v(t_f)^2 - \mu = r(t_f)v(t_f)^2 - g\, r(t_f)^2\right]. \tag{28}$$

Substituting it into Eq. (18-19), the final costates are given by

$$\boldsymbol{\lambda}_r^T(t_f) = \nu_r v(t_f)^2 \hat{\boldsymbol{r}}(t_f)^T, \tag{29}$$
$$\boldsymbol{\lambda}_v^T(t_f) = 2\nu_v r(t_f)v(t_f)\hat{\boldsymbol{v}}(t_f)^T. \tag{30}$$

Similar to the target energy case, the final costate vectors will be orthogonal for circular target orbits.

Apse Line Targeting For insertion at the apse line of a target orbit, the end condition may be represented as

$$\boldsymbol{\theta} = \left[\boldsymbol{r}(t_f)^T \boldsymbol{v}(t_f)\right], \tag{31}$$

which gives

$$\boldsymbol{\lambda}_r^T(t_f) = \nu_r \boldsymbol{v}(t_f), \tag{32}$$
$$\boldsymbol{\lambda}_v^T(t_f) = \nu_v \boldsymbol{r}(t_f). \tag{33}$$

The position and velocities are orthogonal at the apse line of any two-body orbit and consequently, the final costates will also be orthogonal at burnout as seen in these equations.

Targeting r, v, γ If the target radius, velocity magnitude, and flight-path angle are constrained then the end condition may bet written as

$$\boldsymbol{\theta} = \begin{bmatrix} r(t_f) - r_d \\ v(t_f) - v_d \\ \gamma(t_f) - \gamma_d \end{bmatrix} = \begin{bmatrix} r(t_f)^2 - r_d^2 \\ v(t_f)^2 - v_d^2 \\ \boldsymbol{r}(t_f)^T \boldsymbol{v}(t_f) - r_d v_d \cos(90^\circ - \gamma_d) \end{bmatrix} = 0, \tag{34}$$

where γ is the flight-path angle. Substituting this in Eq. (15) gives the costates at t_f:

$$\boldsymbol{\lambda}_r^T(t_f) = \nu_r \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{r}(t_f)} + \nu_\gamma \frac{\partial \boldsymbol{\theta}_3}{\partial \boldsymbol{r}(t_f)} = 2\nu_r \boldsymbol{r}(t_f)^T + \nu_\gamma \boldsymbol{v}(t_f)^T, \tag{35}$$
$$\boldsymbol{\lambda}_v^T(t_f) = \nu_v \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{v}(t_f)} + \nu_\gamma \frac{\partial \boldsymbol{\theta}_3}{\partial \boldsymbol{v}(t_f)} = 2\nu_v \boldsymbol{v}(t_f)^T + \nu_\gamma \boldsymbol{r}(t_f)^T. \tag{36}$$

Orbit Plane Targeting The final orbit plane may be specified using the constraint

$$\boldsymbol{\theta} = [\boldsymbol{r}(t_f) \times \boldsymbol{v}(t_f) - \boldsymbol{h}_d] = 0. \tag{37}$$

Substituting this in Eq. (15) gives the costates at t_f:

$$\boldsymbol{\lambda}_r^T(t_f) = \nu_{hx} \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{r}(t_f)} + \nu_{hy} \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{r}(t_f)} + \nu_{hz} \frac{\partial \boldsymbol{\theta}_3}{\partial \boldsymbol{r}(t_f)} = (\boldsymbol{v}(t_f) \times \boldsymbol{\nu}_h)^T, \tag{38}$$
$$\boldsymbol{\lambda}_v^T(t_f) = \nu_{hx} \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{v}(t_f)} + \nu_{hy} \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{v}(t_f)} + \nu_{hz} \frac{\partial \boldsymbol{\theta}_3}{\partial \boldsymbol{v}(t_f)} = -(\boldsymbol{r}(t_f) \times \boldsymbol{\nu}_h)^T. \tag{39}$$

It can be proven that these two costate vectors are orthogonal at the final time. Consider

$$\boldsymbol{\lambda}_r^T(t_f)\boldsymbol{\lambda}_v(t_f) = -(\boldsymbol{v}(t_f) \times \boldsymbol{\nu}_h)^T(\boldsymbol{r}(t_f) \times \boldsymbol{\nu}_h), \tag{40}$$

$$= -\boldsymbol{v}(t_f)^T(\boldsymbol{\nu}_h \times (\boldsymbol{r}(t_f) \times \boldsymbol{\nu}_h)), \tag{41}$$

$$= -\nu_h^2 \boldsymbol{v}(t_f)^T \boldsymbol{r}(t_f) + (\boldsymbol{\nu}_h^T \boldsymbol{r}(t_f))\boldsymbol{v}(t_f)^T \boldsymbol{\nu}_h. \tag{42}$$

This result shows that the boundary conditions from Eqs. (26-27) for energy targeting and Eqs. (38-39) for orbit plane targeting are together satisfied only if $\boldsymbol{\nu}_h$ is in the direction of the angular momentum. As a result, the final position and velocity costates are orthogonal if the targeted final position and velocity vectors are also orthogonal.

Jagger's End Condition Jagger's "Coke Machine" idea is to simplify the equations by assuming that the position and velocity costates are normal to each other through out the burn and not just at the final time.[7,9] He used the following end condition to provide a basis for this assumption:

$$\boldsymbol{\theta} = \left[\boldsymbol{\lambda}_v^T(t_f)(\boldsymbol{r} \times \boldsymbol{r} \times \boldsymbol{v}) - \boldsymbol{\lambda}_r^T(t_f)(\boldsymbol{v} \times \boldsymbol{r} \times \boldsymbol{v})\right]. \tag{43}$$

Generalized Predictor-Corrector Algorithm for PEG

The conventional PEG implementations use a computationally-efficient predictor-corrector approach for solving the TPBVP. This fully analytical approach employs a number of simplifying assumptions including a simplified gravity model for state propagation. In this subsection, a more generalized predictor-corrector algorithm is presented that can accommodate any arbitrary force model affecting the vehicle motion. In the next subsection, the conventional PEG equations will be derived from the generalized equations of this subsection by employing various simplifications. First, the equations of motion from Eqs. (1-2) are integrated to obtain

$$\mathbf{v}(t_f) - \mathbf{v}(t_0) = \mathbf{v}_G + \mathbf{v}_T, \tag{44}$$

$$\mathbf{r}(t_f) - \mathbf{r}(t_0) = \mathbf{v}(t_0)t_{\text{go}} + \mathbf{r}_G + \mathbf{r}_T, \tag{45}$$

where

$$\mathbf{v}_G \equiv \int_{t_0}^{t_f} \mathbf{g}\, \mathrm{d}t, \tag{46}$$

$$\mathbf{v}_T \equiv -\int_{t_0}^{t_f} \frac{T}{m}\frac{\boldsymbol{\lambda}_v}{\lambda_v}\, \mathrm{d}t, \tag{47}$$

$$\mathbf{r}_G \equiv \int_{t_0}^{t_f}\int_{t_0}^{t} \mathbf{g}\, \mathrm{d}s\, \mathrm{d}t, \tag{48}$$

$$\mathbf{r}_T \equiv -\int_{t_0}^{t_f}\int_{t_0}^{t} \frac{T}{m}\frac{\boldsymbol{\lambda}_v}{\lambda_v}\, \mathrm{d}s\, \mathrm{d}t, \tag{49}$$

and $t_{\text{go}} \equiv t_f - t_0$. The quantities with subscripts G and T are similar to those in conventional PEG literature with subscripts 'GRAV' and 'THRUST', respectively. The thrust steering rates are

always assumed constant in this work, which results from the flat-planet assumption in the costate dynamics. If the reference time in Eq. (24) is changed to t_λ then the expressions for $\mathbf{v}_T$ and $\mathbf{r}_T$ may be written as

$$\mathbf{v}_T = -L_0\hat{\boldsymbol{\lambda}}_v(t_\lambda) + (L_1 - t_\lambda L_0)\boldsymbol{\lambda}_r', \tag{50}$$

$$\mathbf{r}_T = -S_0\hat{\boldsymbol{\lambda}}_v(t_\lambda) + (S_1 - t_\lambda S_0)\boldsymbol{\lambda}_r', \tag{51}$$

where

$$L_0 \equiv \int_{t_0}^{t_f} \frac{T/m}{||\hat{\boldsymbol{\lambda}}_v(t_\lambda) - \boldsymbol{\lambda}_r'(t - t_\lambda)||}\,\mathrm{d}t, \tag{52}$$

$$L_1 \equiv \int_{t_0}^{t_f} \frac{(T/m)t}{||\hat{\boldsymbol{\lambda}}_v(t_\lambda) - \boldsymbol{\lambda}_r'(t - t_\lambda)||}\,\mathrm{d}t, \tag{53}$$

$$S_0 \equiv \int_{t_0}^{t_f}\int_{t_0}^{t} \frac{T/m}{||\hat{\boldsymbol{\lambda}}_v(t_\lambda) - \boldsymbol{\lambda}_r'(s - t_\lambda)||}\,\mathrm{d}s\,\mathrm{d}t, \tag{54}$$

$$S_1 \equiv \int_{t_0}^{t_f}\int_{t_0}^{t} \frac{(T/m)s}{||\hat{\boldsymbol{\lambda}}_v(t_\lambda) - \boldsymbol{\lambda}_r'(s - t_\lambda)||}\,\mathrm{d}s\,\mathrm{d}t. \tag{55}$$

These four quadratures are typically referred to as thrust integrals in PEG literature. Substituting Eqs. (50-51) into Eqs. (44-45) gives the final integrated EOM:

$$\mathbf{v}(t_f) = \mathbf{v}(t_0) + \mathbf{v}_G - L_0\hat{\boldsymbol{\lambda}}_v(t_\lambda) + (L_1 - t_\lambda L_0)\boldsymbol{\lambda}_r', \tag{56}$$

$$\mathbf{r}(t_f) = \mathbf{r}(t_0) + \mathbf{v}(t_0)t_{\text{go}} + \mathbf{r}_G - S_0\hat{\boldsymbol{\lambda}}_v(t_\lambda) + (S_1 - t_\lambda S_0)\boldsymbol{\lambda}_r'. \tag{57}$$

The foundation of the predictor-corrector approach rests on expressing the unknown costates in the integrated EOM in terms of the thrust integrals using

$$\hat{\boldsymbol{\lambda}}_v(t_\lambda) = \frac{(S_1 - t_\lambda S_0)\mathbf{v}_T - (L_1 - t_\lambda L_0)\mathbf{r}_T}{S_0(L_1 - t_\lambda L_0) - L_0(S_1 - t_\lambda S_0)}, \tag{58}$$

$$\boldsymbol{\lambda}_r' = \frac{S_0\mathbf{v}_T - L_0\mathbf{r}_T}{S_0(L_1 - t_\lambda L_0) - L_0(S_1 - t_\lambda S_0)}. \tag{59}$$

These equations are not explicit in nature because the thrust integrals themselves depend on the costates. Therefore, an iterative approach may be used to solve these equations starting from a suitable guess for the costates. When the downrange component of position at burnout is free then from Eq. (15), $\boldsymbol{\lambda}_r'$ has no component in that direction, which yields

$$\mathbf{r}_T \cdot \hat{\boldsymbol{d}} = \frac{S_0}{L_0}(\mathbf{v}_T \cdot \hat{\boldsymbol{d}}), \tag{60}$$

where $\hat{\boldsymbol{d}}$ is the unit vector in the downrange direction. The two integrals from Eqs. (46) and (48) also need to be evaluated for completing the predictor-corrector approach. If the force model only includes the two-body gravity term then these integrals are equivalent to the familiar gravity integrals in PEG literature. Here, these integrals are evaluated using numerical quadrature to compensate for a general force model.

The conventional PEG's predictor-corrector algorithm iterates on a single independent variable $\mathbf{v}_{\text{go}}$ to achieve the desired burnout state. For this, it is necessary to express all the unknowns including the thrust integrals as a function of $\mathbf{v}_{\text{go}}$, which is defined as

$$\mathbf{v}_{\text{go}} = \mathbf{v}_d - \mathbf{v}(t_0) - \mathbf{v}_G, \tag{61}$$

where $\mathbf{v}_d$ is the desired burnout velocity vector and is different from $\mathbf{v}(t_f)$ because the latter denotes the predicted velocity state at burnout. By substituting $\mathbf{v}_{\text{go}}$ for $\mathbf{v}_T$ in Eqs. (58-59), a correction to costate values may be obtained. The generalized predictor-corrector algorithm in this work follows the similar approach as used in the conventional PEG where the position and velocity states at burnout are estimated in the predictor step and then $\mathbf{v}_{\text{go}}$ is corrected by using targeting errors. Algorithm 1 gives the pseudocode for generalized preedictor-corrector formulation of PEG.

Algorithm 1 Generalized Predictor-Corrector for PEG

1: Inputs: $\mathbf{r}(t_0), \mathbf{v}(t_0), m(t_0), r(t_f), v(t_f), \gamma(t_f), \hat{\mathbf{h}}(t_f), T, c$
2: Outputs: t_{go}, $\hat{\boldsymbol{\lambda}}_v(t_0)$, $\boldsymbol{\lambda}'_r$
3: Initialize $\mathbf{v}_{\text{go}}$, $\mathbf{r}_d$, $\mathbf{v}_{\text{G}}$, $\mathbf{r}_{\text{G}}$, $\hat{\boldsymbol{\lambda}}_v(t_\lambda)$, $\boldsymbol{\lambda}'_r$
4: Compute t_{go} using the rocket equation
5: $t_\lambda \leftarrow 0, \quad t_0 \leftarrow 0, \quad t_f \leftarrow t_{\text{go}}$
6: Compute thrust integrals in Eqs. (52-55) by numerical quadrature
7: $\mathbf{r}_d \leftarrow r(t_f)\hat{\mathbf{r}}_d$
8: $\mathbf{v}_d \leftarrow v(t_f)\left(\sin(\gamma(t_f))\hat{\mathbf{r}}_d - \cos(\gamma(t_f))(\hat{\mathbf{r}}_d \times \hat{\mathbf{h}}(t_f))\right)$
9: $\mathbf{v}_T \leftarrow \mathbf{v}_d - (\mathbf{v}(t_0) + \mathbf{v}_G)$
10: $\mathbf{r}_T \leftarrow \mathbf{r}_d - (\mathbf{r}(t_0) + \mathbf{v}(t_0)t_{\text{go}} + \mathbf{r}_G)$
11: $\hat{\mathbf{d}}(t_f) \leftarrow \hat{\mathbf{h}}(t_f) \times \hat{\mathbf{r}}_d$
12: $\mathbf{r}_T \leftarrow S_0/L_0(\mathbf{v}_T \cdot \hat{\mathbf{d}}(t_f))\hat{\mathbf{d}}(t_f) + (\mathbf{r}_T \cdot \hat{\mathbf{r}}_d)\hat{\mathbf{r}}_d + (\mathbf{r}_T \cdot \hat{\mathbf{h}}(t_f))\hat{\mathbf{h}}(t_f)$
13: Update $\hat{\boldsymbol{\lambda}}_v(t_0)$ and $\boldsymbol{\lambda}'_r$ by using Eqs. (58-59)
14: Repeat steps 6 to 13 until $L_0, L_1, S_0, S_1, \hat{\boldsymbol{\lambda}}_v(t_0)$, and $\boldsymbol{\lambda}'_r$ converge
15: Predictor: compute the cutoff states $\mathbf{r}_P, \mathbf{v}_P$ at t_{go} using Eqs. (56-57)
16: Predictor: update $\mathbf{v}_{\text{G}}$ and $\mathbf{r}_{\text{G}}$ by numerical quadrature along the predicted trajectory
17: Predictor: $\mathbf{r}_d \leftarrow \mathbf{r}_P - (\mathbf{r}_P \cdot \hat{\mathbf{h}}(t_f))\hat{\mathbf{h}}(t_f)$
18: Corrector: $\mathbf{v}_{\text{miss}} \leftarrow \mathbf{v}_d - \mathbf{v}_P$
19: Corrector: $\mathbf{v}_{\text{go}} \leftarrow \mathbf{v}_{\text{go}} + \mathbf{v}_{\text{miss}}$
20: Repeat steps 4 to 19 until $\mathbf{v}_{\text{go}}$ converges

Analytical Predictor-Corrector Algorithm

The conventional PEG implementation uses a simplified version of Eq. (24) to express the costates as an explicit function of the thrust integrals. Jagger's proposed his "Coke Machine" idea that the position and velocity costates can be assumed to be orthogonal during the entire burn for this

purpose.[7,9] This orthogonality constraint is justified for the optimal finite burns given the thrust direction varies linearly in a plane by a small angle as shown next.[17] Consider the following relation for the turn angle ϕ between $\boldsymbol{u}(t)$ and $\hat{\boldsymbol{\lambda}}_v(t_\lambda)$ obtained from Eq. (24) after replacing t_0 with t_λ:

$$\tan\phi(t) = -\frac{(t-t_\lambda)\lambda_r' \sin\psi}{1-(t-t_\lambda)\cos\psi}, \tag{62}$$

where ψ is the constant angle between $\hat{\boldsymbol{\lambda}}_v(t_\lambda)$ and $\boldsymbol{\lambda}_r'$. If ψ is a right angle then $\tan\phi(t)$ varies linearly with time and the corresponding turning rate of the thrust vector is equal to λ_r'. Then $\lambda_r' t_{\rm go}$ becomes the total change in thrust direction over the burn time $t_{\rm go}$. Additionally,

$$\boldsymbol{u}(t)\cdot\hat{\boldsymbol{\lambda}}_v(t_\lambda) = \cos\phi(t) = -\frac{1}{1+\lambda_r'^2(t-t_\lambda)^2}. \tag{63}$$

For small $\phi(t)$, the term on the right hand side becomes unity. These two assumptions ($\psi = 90^\circ$ and $\phi(t) \approx 0$) are employed by the Space Shuttle's PEG implementation to simplify predictor-corrector formulation. The turn angle $\phi(t)$ is often limited to some maximum value in PEG implementations to avoid breakdown of the small angle assumption.[5] A direct consequence of these assumptions is that the four thrust integrals of Eqs. (52-55) no longer depend on the costates and are fully determined by $t_{\rm go}$ and thrust profile. Additionally, if the reference time t_λ in Eq. (50) is chosen as

$$t_\lambda = \frac{L_1}{L_0} \tag{64}$$

and $\mathbf{v}_{\rm go}$ is used in place of $\mathbf{v}_T$ then Eq. (50) becomes

$$\mathbf{v}_{\rm go} = -L_0\hat{\boldsymbol{\lambda}}_v(t_\lambda). \tag{65}$$

This result shows that the velocity costate at t_λ is approximately in the direction of velocity-to-be-gained. It can be proven that t_λ is equal to the mid point of $t_{\rm go}$ if the thrust integrals are computed by assuming constant thrust acceleration. In constant thrust mode, these integrals simplify to

$$L_0 = ||\mathbf{v}_{\rm go}||, \tag{66}$$
$$S_0 = -L_0(\tau - t_{\rm go}) + c\,t_{\rm go}, \tag{67}$$
$$L_1 = L_0 t_{\rm go} - S_0, \tag{68}$$
$$S_1 = S_0\tau - c\,t_{\rm go}^2/2, \tag{69}$$

where $\tau \equiv -m(t_0)/\dot{m}$ and is always positive. The steering rates are given by Eqs. (58-59), which also simplify to

$$\hat{\boldsymbol{\lambda}}_v(t_\lambda) = -\frac{\mathbf{v}_{\rm go}}{L_0}, \tag{70}$$
$$\boldsymbol{\lambda}_r' = \frac{L_0\mathbf{r}_{\rm go} - S_0\mathbf{v}_{\rm go}}{L_0(S_1 - t_\lambda S_0)}. \tag{71}$$

The PEG implementation in Space Shuttle retains the second-order terms in t_{go} for predicting the cutoff or burnout states.[1] The thrust unit vector from Eq. (24) after employing the orthogonality constraint becomes

$$\hat{\boldsymbol{u}}(t) = -\frac{\boldsymbol{\lambda}'_v(t_\lambda) - \boldsymbol{\lambda}'_r(t - t_\lambda)}{\sqrt{1 + \lambda'^2_r(t - t_\lambda)^2}}. \tag{72}$$

This equation can be approximated as

$$\hat{\boldsymbol{u}}(t) = -\boldsymbol{\lambda}'_v(t_\lambda) + \boldsymbol{\lambda}'_r(t - t_\lambda) + 1/2\lambda'^2_r(t - t_\lambda)^2\boldsymbol{\lambda}'_v(t_\lambda), \tag{73}$$

where the third-order terms are ignored. By substituting this result in Eqs. (1-2) and subsequent integration yield

$$\mathbf{v}(t_f) = \mathbf{v}(t_0) + \mathbf{v}_G - (L_0 - 1/2\lambda'^2_r(L_2 - t_\lambda L_1))\hat{\boldsymbol{\lambda}}_v(t_\lambda), \tag{74}$$

$$\mathbf{r}(t_f) = \mathbf{r}(t_0) + \mathbf{v}(t_0)t_{\text{go}} + \mathbf{r}_G - (S_0 - 1/2\lambda'^2_r(S_2 + t^2_\lambda S_0 - 2t_\lambda S_1))\hat{\boldsymbol{\lambda}}_v(t_\lambda) + (S_1 - t_\lambda S_0)\boldsymbol{\lambda}'_r, \tag{75}$$

where

$$L_2 \equiv \int_{t_0}^{t_f} (T/m)t^2\,\mathrm{d}t, \qquad S_2 \equiv \int_{t_0}^{t_f}\int_{t_0}^{t} (T/m)s^2\,\mathrm{d}s\,\mathrm{d}t. \tag{76}$$

L_2 and S_2 are also thrust integrals. These equations are used in the analytical predictor step for computing cutoff position and velocity states. The two gravity integrals ($\mathbf{v}_G$ and $\mathbf{r}_G$) can be computed analytically by constructing a coasting trajectory that remains close to the powered trajectory[1] or by Jagger's method of approximate analytical integration.[7,17] The corrector step essentially remains the same as given in Algorithm 1. The specific mechanization of this analytical predictor-corrector algorithm as used for Space Shuttle PEG is given by McHenry et al.[1] A flow chart for the analytical PEG algorithm is shown in Fig. 1.

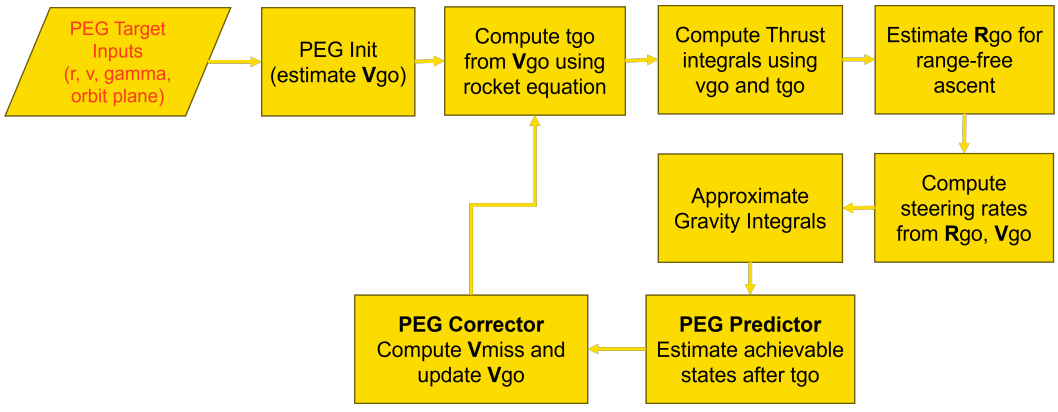

Figure 1. Flowchart for Analytical Predictor-Corrector PEG

ELLIPTICAL ORBIT INSERTION USING PEG

Conventional PEG algorithm uses the target altitude, velocity, flight-path angle, and orbit plane for computing the desired position and velocity states at burnout. For an elliptical target orbit, r, v, and γ fixes the true anomaly at the burnout and as a result, PEG is free to choose the line of apsides orientation for the target orbit because the downrange is not constrained. This may result in an undesirable final orbit whose apse lines is rotated with respect to the desired orientation. This limitation of the conventional PEG target conditions is illustrated in Fig. 2. These targets vary with the insertion point on the desired elliptical orbit and as a result, they cannot be predetermined and must be solved for during PEG iterations. An alternative set of target conditions must use the orbit shape and orientation as fixed quantities with the insertion point kept free. The is achieved by considering the burnout true anomaly as a free variable.

In the present work, new PEG target equations are derived that uses the final orbit apoapsis and periapsis radii in addition to the argument of periapsis (ω_t) to compute the target true anomaly at burnout. Subsequently, the position and velocity magnitudes at the computed true anomaly are determined to form the target position and velocity states at burnout. If θ represents the argument of latitude then the burnout true anomaly (ν_t) is determined using

$$\theta = \cos^{-1}(\hat{n}.\hat{\boldsymbol{R}}_t), \tag{77}$$

$$\nu_t = \theta - \omega_t, \tag{78}$$

where $\hat{n}$ is the unit vector along the line of ascending node, $\hat{R}_t$ is estimated target radius unit vector from the conventional PEG. If r_p and r_a are the target orbit peripsis and apoapsis radii then the target radius magnitude is computed as

$$a_t = \frac{r_p + r_a}{2}, \tag{79}$$

$$p_t = \frac{r_p r_a}{a_t}, \tag{80}$$

$$h_t = \sqrt{\mu p_t}, \tag{81}$$

$$r_t = \sqrt{\frac{h_t}{n_t}}, \tag{82}$$

where n_t is the target orbit mean motion. With r_t determined, the target position and velocity vectors can now be computed using

$$\boldsymbol{R}_t = r_t \hat{R}_t, \tag{83}$$

$$\boldsymbol{V}_t = \frac{\mu e_t}{h_t} \sin(\nu_t) \hat{\boldsymbol{R}}_t + \frac{h_t}{r_t} (\hat{h} \times \hat{\boldsymbol{R}}_t), \tag{84}$$

where e_t is the target orbit eccentricity. These equations replace the ones used in Step 7 and 8 of Algorithm 1. A sample lunar ascent simulation for targeting an elliptical orbit with desired ω_t using the modified PEG is implemented in MATLAB for validating these equations. The resulting lunar ascent trajectory is given in Fig. 3, which shows that the classical orbit elements of the achieved orbit match the desired values.

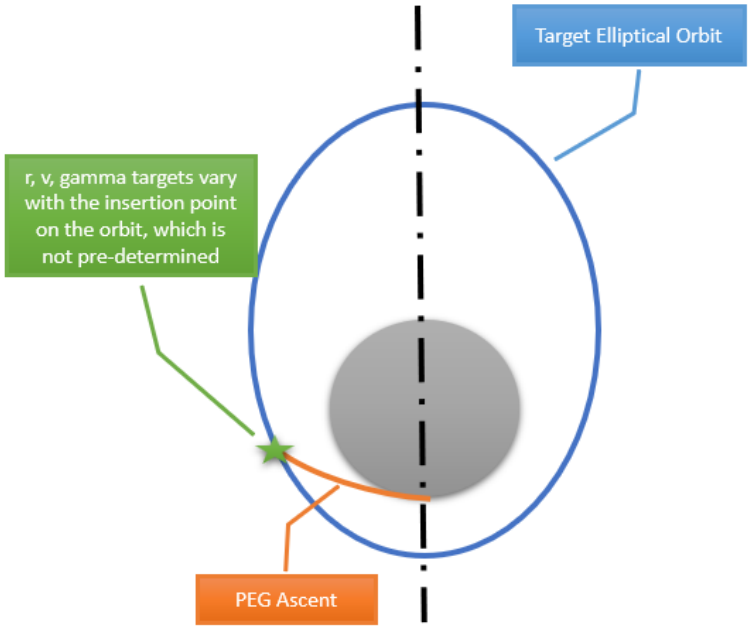

Figure 2. Limitations of conventional PEG targets (r, v, γ) for elliptical orbit insertion

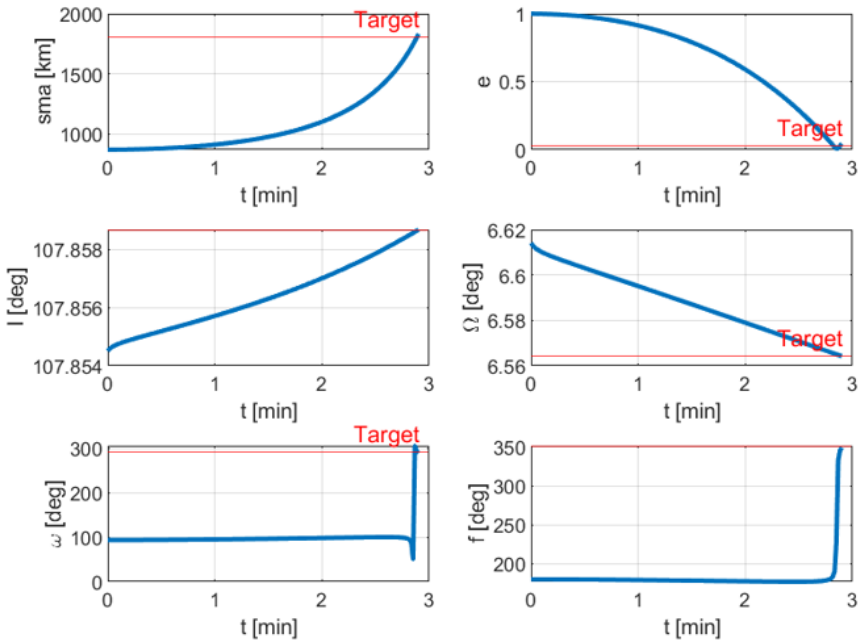

Figure 3. Simulation results for elliptical lunar orbit insertion using the modified PEG equations

DETERMINATION OF PEG TIME-OF-IGNITION

The free-downrange condition used in PEG presents a challenge for guiding powered descent trajectories to the desired landing site. While it is possible to remove the free downrange condition from PEG and formulate a new predictor-corrector algorithm that targets a specified downrange distance at the cutoff time, this work presents a few simple modifications to PEG that enable targeting a specified landing site without reformulating PEG equations for powered descent problem specifically. First, a method to determine the time-of-ignition (TIG) for PEG is presented in this section that may be used as a coarse range control for descent trajectory. For illustration, a lunar descent example is shown in Fig. 4 in which an Apollo-style lander is coasting in an elliptical descent orbit after executing the descent orbit insertion (DOI) burn. The lander coasts until it reaches the point of powered descent initiation (PDI). The TIG determination uses a priori specified (co-planar) downrange arc angle (θ) between the position vector at TIG and landing site position vector ($\mathbf{R}_t$). The PEG downrange arc angle can be computed offline based on a relatively consistent and repeatable powered landing sequence. The TIG for the PEG-guided powered descent burn would be determined by propagating the coast state forward and comparing the resulting downrange arc angle to the predetermined value. Figure 4 shows the total arc angle ϕ from this forward-propagated initial position $\mathbf{R}_o$ to the landing site and is computed using

$$\phi = \cos^{-1}(\mathbf{R}_o \cdot \mathbf{R}_t). \tag{85}$$

The coasting phase error β is simply

$$\beta = \phi - \theta \tag{86}$$

and its current rate of change at $\mathbf{R}_o$ may be computed using

$$\dot{\beta} = \frac{||\mathbf{V}_o|| \cos\gamma_o}{||\mathbf{R}_o||}, \tag{87}$$

where γ_o is the flight-path angle at $\mathbf{R}_o$. The new added (or subtracted) time of coast duration is then simply $\beta/\dot{\beta}$. The TIG is accordingly updated using

$$\text{TIG}_{\text{new}} = \text{TIG}_{\text{old}} + \frac{\beta}{\dot{\beta}} \tag{88}$$

It is noted that the accuracy of this method primarily depends on the availability of an accurate estimate of θ, whose value is a function of PEG inputs including thrust magnitude.

PEG RANGE CONTROL FOR PLANETARY DESCENT

In this section, two different approaches for precise range control during powered descent are presented. Precise range control facilitates the descent burn to adapt to dispersion and achieves a desired range associated with a particular landing site (see Figure 5). Both approaches utilize thrust modulation for range control. The first approach, referred to as Vernier Range Control in this work, uses an outer loop to compute new thrust magnitude to drive the range error to zero. The second method modifies the corrector step of PEG to compute corrections for $\mathbf{v}_{\text{go}}$ and thrust magnitude simultaneously for range control.

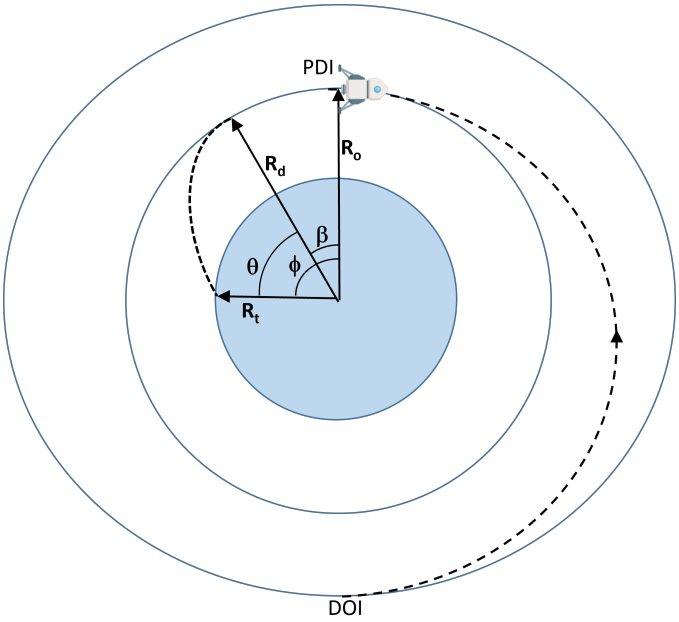

Figure 4. TIG determination for PEG-guided lunar descent trajectory.

Vernier Range Control

The vernier range control method allows the lunar descent vehicle to adapt to dispersions during the landing phase (e.g., thrust dispersions, TIG delays, etc.) and achieve a desired range associated with a particular landing site (see Fig. 5). For a given current position, R, the vernier range control nulls out a range angle error, $\Delta\theta$, so that the predicted final position (from PEG), R_t, coincides with the desired final touchdown position, R_{tf}. Since PEG solves a range-free problem, another control is required to allow the descent vehicle to control range. This additional control is provided through changes in the commanded thrust level. The thrust magnitude changes are dependent on the amount of range angle error to be removed. It is desirable to develop an analytic expression which relates the sensitivity of range angle error to the changes in thrust magnitude (T). This thrust magnitude change (ΔT) is applied to the current thrust level to produce a new thrust level that eliminates the range angle error as shown:

$$T_{NEW} = T_{CURRENT} + \Delta T, \tag{89}$$

where

$$\Delta T = \frac{\mathrm{d}T}{\mathrm{d}\theta}\Delta\theta, \tag{90}$$

$$= \frac{\mathrm{d}T}{\mathrm{d}t_{\mathrm{go}}}\frac{\mathrm{d}t_{\mathrm{go}}}{\mathrm{d}\theta}\Delta\theta, \tag{91}$$

and $\Delta\theta$ is the range angle error. The first derivative term can be obtained by using the rocket equation as shown next:

$$t_{\text{go}} = \frac{mc}{T}(1 - \exp(-v_{\text{go}}/c)), \tag{92}$$

$$\frac{\mathrm{d}T}{\mathrm{d}t_{\text{go}}} = -\frac{T^2}{mc}(1 - \exp(-v_{\text{go}}/c))^{-1}. \tag{93}$$

The second derivative in Eq. (91) may be obtained by observing in Fig. 5 that the change in the downrange distance, ΔD_{dr}, is related to a change in t_{go}. This relation is given as

$$\Delta D_{dr} \approx V_h \Delta t_{\text{go}}, \tag{94}$$

where V_h denotes the horizontal component of the current velocity. On the other hand, the change in the downrange distance is also related to the change in the downrange angle:

$$\Delta D_{dr} \approx R\Delta\theta. \tag{95}$$

Equating the preceding two equation yields the sensitivity of t_{go} to $\Delta\theta$:

$$\frac{\mathrm{d}t_{go}}{\mathrm{d}\theta} = \frac{R}{V_h}. \tag{96}$$

Substituting Eqs. (93) and (96) into Eq. (91) yields the analytic expression for thrust magnitude sensitivity to the range angle error:

$$\Delta T = \frac{-T^2}{mc(1 - \exp(-v_{go}/c))}\frac{R}{V_h}\Delta\theta. \tag{97}$$

The sign of ΔT is determined by the overshoot or undershoot of the desired final position. For example, if the currently predicted final position R_t overshoots the desired position R_{tf} by $\Delta\theta$, then the thrust is increased by $+|\Delta T|$ to reduce t_{go} for powered descent. This has the effect of reducing the overall range, thus canceling out the overshoot. In the opposite scenario where R_t undershoots R_{tf}, the thrust is reduced by $-|\Delta T|$ in order to increase the range, thus canceling out the undershoot.

RTLS-Inspired Range Control

Another way of achieving thrust modulation is to modify the corrector step of PEG to compute corrections for thrust magnitude simultaneously with the corrections for $\mathbf{v}_{\text{go}}$. This approach is similar to that of RTLS abort mode guidance of Space Shuttle that enables it to return to the launch site in case of an emergency. For range control, it will be assumed in this section that the current estimate of the range angle to the desired landing site is available to PEG. Then the corrector step (see Step 19 in Algorithm 1) of PEG may be replaced by following equations:

$$\begin{bmatrix} \Delta\mathbf{v}_{go} \\ \Delta T \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{v}_{\text{miss}}}{\partial \mathbf{v}_{\text{go}}} & \frac{\partial \mathbf{v}_{\text{miss}}}{\partial T} \\ \frac{\partial \theta}{\partial \mathbf{v}_{\text{go}}} & \frac{\partial \theta}{\partial T} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{v}_{\text{miss}} \\ \Delta\theta \end{bmatrix}, \tag{98}$$

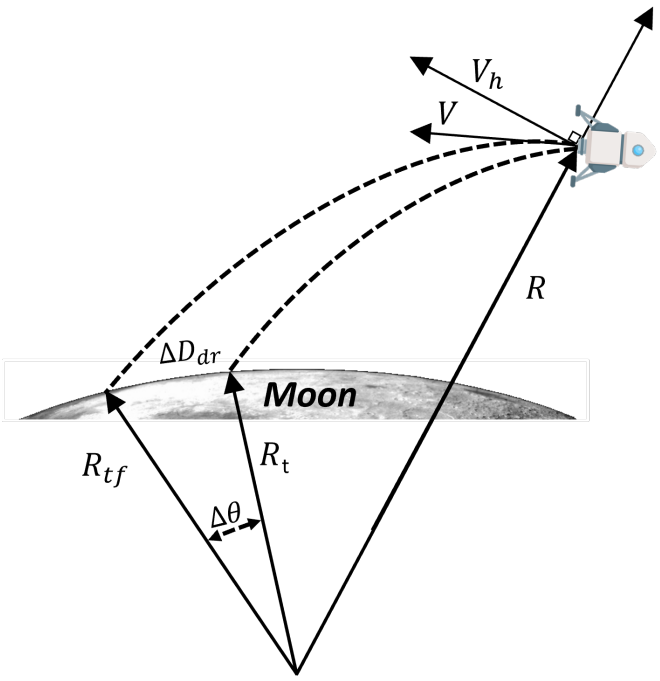

Figure 5. Vernier Range Control for Powered Descent

where

$$\Delta\theta = (\mathbf{r}_d - \mathbf{r}_P) \cdot \hat{\mathbf{r}}(t_0), \tag{99}$$

$$\frac{\partial \mathbf{v}_{\text{miss}}}{\partial \mathbf{v}_{\text{go}}} = \frac{\partial \mathbf{v}_G}{\partial t_{\text{go}}} \frac{\partial t_{\text{go}}}{\partial \mathbf{v}_{\text{go}}} + \frac{\partial \mathbf{v}_T}{\partial \mathbf{v}_{\text{go}}}, \tag{100}$$

$$\frac{\partial \mathbf{v}_{\text{miss}}}{\partial T} = \frac{\partial \mathbf{v}_G}{\partial t_{\text{go}}} \frac{\partial t_{\text{go}}}{\partial T}, \tag{101}$$

$$\frac{\partial \theta}{\partial \mathbf{v}_{\text{go}}} = (\hat{\mathbf{r}}(t_0) \cdot \mathbf{v}_P) \frac{\partial t_{\text{go}}}{\partial \mathbf{v}_{\text{go}}}, \tag{102}$$

$$\frac{\partial \theta}{\partial T} = (\hat{\mathbf{r}}(t_0) \cdot \mathbf{v}_P) \frac{\partial t_{\text{go}}}{\partial T}, \tag{103}$$

$$\frac{\partial \mathbf{v}_G}{\partial t_{\text{go}}} = \frac{(\mathbf{v}_G \cdot \hat{\mathbf{r}}(t_0))}{t_{\text{go}}} \hat{\mathbf{r}}(t_0), \tag{104}$$

$$\frac{\partial t_{\text{go}}}{\partial \mathbf{v}_{\text{go}}} = \frac{m(t_{\text{go}}) \mathbf{v}_{\text{go}}^T}{T||\mathbf{v}_{\text{go}}||}, \tag{105}$$

$$\frac{\partial t_{\text{go}}}{\partial T} = -\frac{t_{\text{go}}}{T}. \tag{106}$$

These equations were implemented in a lunar landing simulation in MATLAB and the resulting powered descent trajectory is shown in Fig. 6. These results show that the modified PEG was able to target the landing site using meter-level accuracy by thrust modulation. Figure 7 shows the thrust commanded by the modified PEG for precise range control in this simulation.

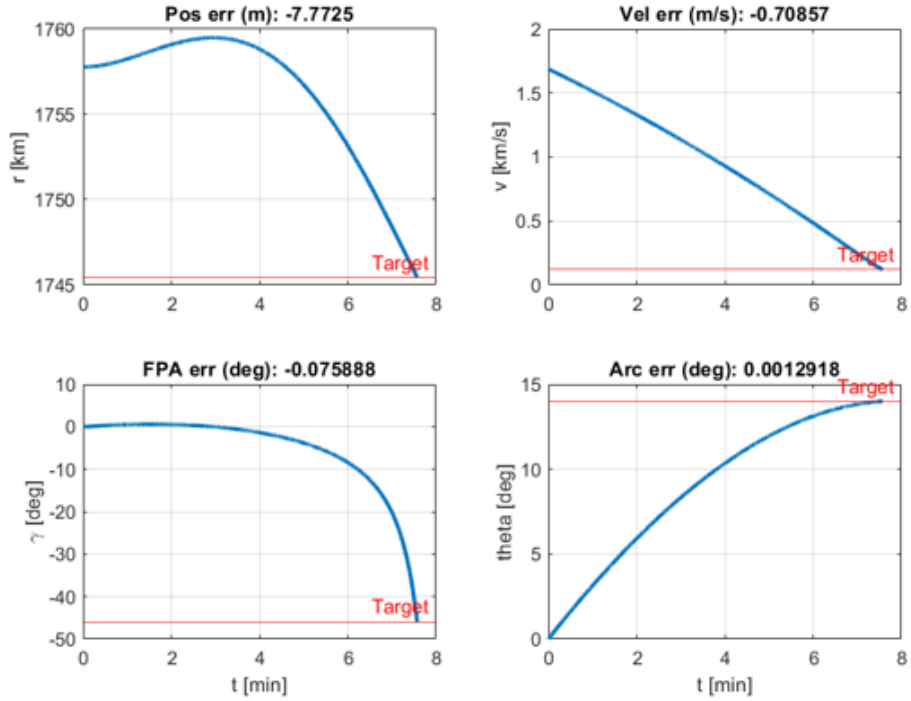

Figure 6. Precise Range Control using Descent PEG for a Lunar Lander

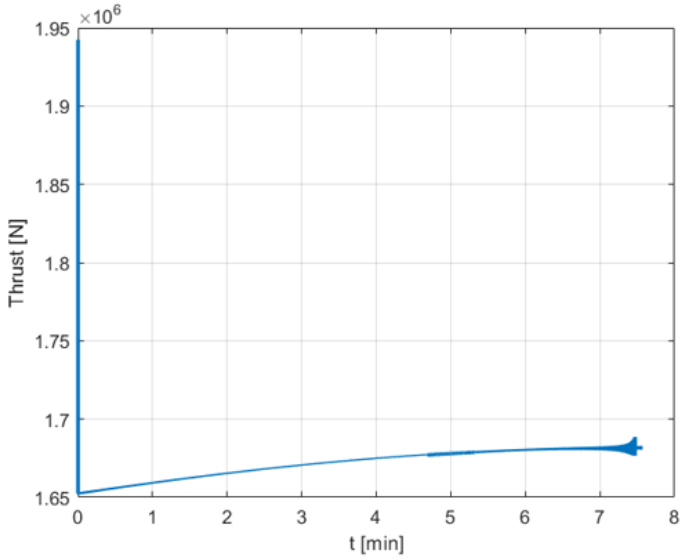

Figure 7. Thrust Modulation by Descent PEG for Precise Range Control

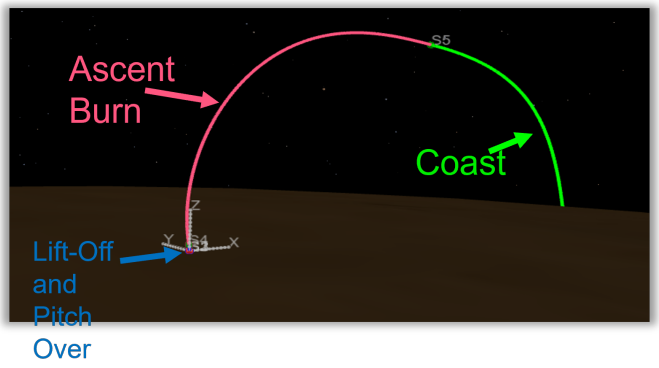

Figure 8. Mars Ascent Trajectory with PEG Ascent Burn

DRAG COMPENSATION IN PEG

The conventional PEG incorporates the effects of an approximate gravity model and thrust accelerations on the predicted trajectory. For those planetary ascent and descent trajectories affected by atmospheric drag significantly, PEG can still be used for guidance because it is used in a closed-loop fashion to frequently correct for the guidance commands based on targeting errors. The consequence of using a low-fidelity force model in PEG predictor step is more fuel consumption. The more general predictor-corrector formulation of PEG given in Algorithm 1 is capable of compensating for atmospheric drag forces to facilitate a more fuel-efficient guidance algorithm.

In this work, acceleration imparted by atmospheric drag to the vehicle are computed and used in the prediction steps 15 and 16 of Algorithm 1. It can be seen from Eqs. (10-12) that adding atmospheric drag in the force model affects the costate dynamics. However, these effects on costate propagation are ignored in the present work and the same predictor-corrector formulation of PEG can still be used. The only modifications needed in PEG are: include atmospheric drag force in the numerical propagation of cutoff states (Step 15 of Algorithm 1) and compute atmospheric drag integrals as part of $\mathbf{v}_G$ and $\mathbf{r}_G$ terms (Step 16 of Algorithm 1). To validate this approach, a Mars ascent simulation was developed by integrating PEG with NASA's Copernicus tool*. The Mars ascent trajectory with ascent burn guided by the modified PEG with drag compensation is shown in Fig. 8. In this simulation, PEG was implemented as an external plugin of Copernicus. The ascent trajectory was optimized using a two-level optimization approach with Copernicus' optimizer acting as an outer loop for determining optimal PEG targets for maximum performance. Figure 9 shows the block diagram of this two-level optimization strategy with PEG computing the ascent guidance commands in the inner loop.

CONCLUSION

PEG is a versatile guidance algorithm that can be adapted to different vehicles and trajectory requirements. In this work, many enhancements to the conventional PEG algorithm are discussed that further improve its versatility by adding capabilities for elliptical orbit insertion, powered descent guidance with thrust modulation for precise range control, compensation for atmospheric drag effects for endo-atmospheric ascent guidance, and PEG time-of-ignition determination. A new

*https://www.nasa.gov/general/copernicus

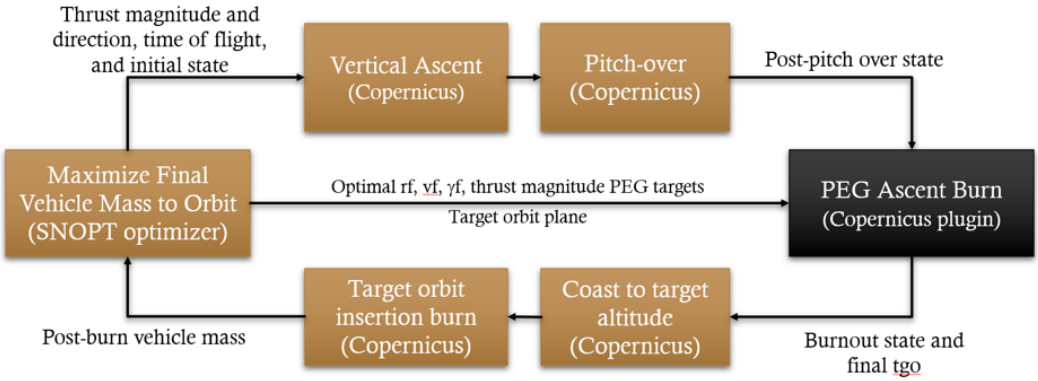

Figure 9. Block Diagram for Mars Ascent Simulation using Copernicus and PEG

general predictor-corrector formulation of PEG algorithm is also presented in this work that can compensate for effects due to any arbitrary force model on the guided trajectory to save propellant.

REFERENCES

[1] R. L. McHenry, T. J. Brand, A. D. Long, B. F. Cockrell, and J. R. Thibodeau III, "Space Shuttle Ascent Guidance, Navigation, and Control," *The Journal of the Astronautical Sciences*, Vol. XXVII, No. 1, 1979, pp. 1–38.

[2] A. R. Klumpp, "Apollo lunar descent guidance," *Automatica*, Vol. 10, Mar. 1974, pp. 133–146, 10.1016/0005-1098(74)90019-3.

[3] J. Goodman, "George Cherry and Apollo Lunar Module Guidance," *AIAA SCITECH 2023 Forum*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, Jan. 2023, 10.2514/6.2023-0313.

[4] T. Fill, J. Goodman, and S. Robinson, "Orion's Powered Flight Guidance Burn Options for Near Term Exploration Missions," Springfield, VA, United States, Feb. 2018.

[5] P. V. d. Porten, N. Ahmad, M. Hawkins, and T. Fill, "Powered Explicit Guidance Modifications and Enhancements for Space Launch System Block-1 and Block-1B Vehicles," Springfield, VA, United States, Feb. 2018. NTRS Document ID: 20180002035.

[6] J. Goodman, "Helmut Horn and the Origin of the Saturn V Iterative Guidance Mode (IGM)," *AIAA Scitech 2021 Forum*, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, Jan. 2021, 10.2514/6.2021-2020.

[7] R. Jaggers, "An explicit solution to the exoatmospheric powered flight guidance and trajectory optimization problem for rocket propelled vehicles," *Guidance and Control Conference*, Hollywood,FL,U.S.A., American Institute of Aeronautics and Astronautics, Aug. 1977, 10.2514/6.1977-1051.

[8] T. Fill, "Introduction to Bi-Linear Tangent Steering For Shuttle Ascent & Aborts," Tech. Rep. EGB-89-108 SHUTTLE-89-022, The Charles Stark Draper Laboratory, Inc., May 1989.

[9] J. Goodman, "Roland Jaggers and the Development of Space Shuttle Powered Explicit Guidance (PEG)," *AIAA Scitech 2021 Forum*, VIRTUAL EVENT, American Institute of Aeronautics and Astronautics, Jan. 2021, 10.2514/6.2021-2021.

[10] C. Schroeder and G. L. Condon, "Application of Powered Explicit Guidance to Lunar Descent," May 1991.

[11] T. Fill, "Guidance Alg. for Simulating Pwrd Descent to the Lunar Surface," Tech. Rep. ESB-92-219, LMI-92-027, The Charles Stark Draper Laboratory, Inc., Apr. 1992.

[12] T. Fill, "LUNAR LANDING AND ASCENT TRAJECTORY GUIDANCE DESIGN FOR THE AUTONOMOUS LANDING AND HAZARD AVOIDANCE TECHNOLOGY (ALHAT) PROGRAM," *Advances in the Astronautical Sciences*, Univelt Inc., 2010.

[13] G. A. Dukeman and W. C. Krolick, "Autonomous Burn Targeting for a Lunar Sortie Staged From a Near-Rectilinear Halo Orbit," Broomfield, CO. NTRS Document ID: 20240009379 NTRS Research Center: Marshall Space Flight Center (MSFC).

[14] S. Thrasher and T. Fill, "Orion's Exoatmospheric Burn Guidance Architecture and Algorithm," *AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon, American Institute of Aeronautics and Astronautics, Aug. 2011, 10.2514/6.2011-6262.

[15] B. Mahajan and G. L. Condon, "HLS NRHO to Lunar Surface and Back Mission Design," Aug. 2022. https://ntrs.nasa.gov/citations/20230002566.

[16] B. Mahajan, "Powered Explicit Guidance (PEG) Modifications for Fixed Line-of-Apsides Targeting," Dec. 2023. HLS Flight Mechanics Internal Presentation, NASA-Johnson Space Center.

[17] R. F. Jaggers, "Shuttle Powered Explicit Guidance (PEG) Algorithm," Tech. Rep. JSC-261 22, National Aeronautics and Space Administration, Houston, TX, Nov. 1992.