(Preprint) AAS 18-084

ORION'S POWERED FLIGHT GUIDANCE BURN OPTIONS FOR NEAR TERM EXPLORATION MISSIONS

Thomas Fill,[*] John Goodman,[†] and Shane Robinson[‡]

NASA's Orion exploration spacecraft will fly more demanding mission profiles than previous NASA human flight spacecraft. Missions currently under development are destined for cislunar space. The EM-1 mission will fly unmanned to a Distant Retrograde Orbit (DRO) around the Moon. EM-2 will fly astronauts on a mission to the lunar vicinity. To fly these missions, Orion requires powered flight guidance that is more sophisticated than the orbital guidance flown on Apollo and the Space Shuttle. Orion's powered flight guidance software contains five burn guidance options. These five options are integrated into an architecture based on a proven shuttle heritage design, with a simple closed-loop guidance strategy. The architecture provides modularity, simplicity, versatility, and adaptability to future, yet-to-be-defined, exploration mission profiles. This paper provides a summary of the executive guidance architecture and details the five burn options to support both the nominal and abort profiles for the EM-1 and EM-2 missions.

INTRODUCTION

NASA's mandate to develop a Multi-Purpose Crew Vehicle (MPVC) to advance human exploration of deep space is well underway. The two centerpieces of this exploration program, the Space Launch System (SLS) and the Orion Crew Module (CM) with its companion Service Module (SM) are approaching their first joint flight sometime in late 2019 in an uncrewed test flight to cislunar space. As follow-on missions take on greater complexity and venture farther from Earth, Orion's mission design will need to cover many different types of burns and trajectory constraints and its systems will be expected to provide much greater levels of autonomy than previous crewed vehicles. Future missions will require further definition and the on-board systems must be flexible enough to handle these new missions without major modification. To satisfy these demands, the Orion powered-flight Orbit Guidance (OrbGuid) leverages many decades of successful flight heritage from the Apollo and Shuttle eras by using algorithms and designs that have proven utility and reliability, while incorporating lessons learned in the areas of advanced precision, autonomy, and architecture design. The result is an algorithm that is flexible and easily extensible to a wide variety of orbit transfer scenarios.

[*] Principal Member Technical Staff, Guidance & Control Group, MS 70, Draper Laboratory, 555 Technology Square, Cambridge, MA.

[†] Specialist Engineer, Odyssey Space Research, LLC, 1120 NASA Parkway, Suite 505, Houston, TX, 77058.

[‡] Aerospace Engineer, Aeroscience and Flight Mechanics Division, NASA Johnson Space Center, 2101 NASA Parkway, Houston, TX, 77058.

THE EM-1 MISSION PROFILE

Exploration Mission 1 (EM-1) has three main objectives to meet before the first crewed flight: to demonstrate spacecraft systems performance, to demonstrate high speed entry and thermal protection system performance, and to achieve a successful landing off the coast of California.

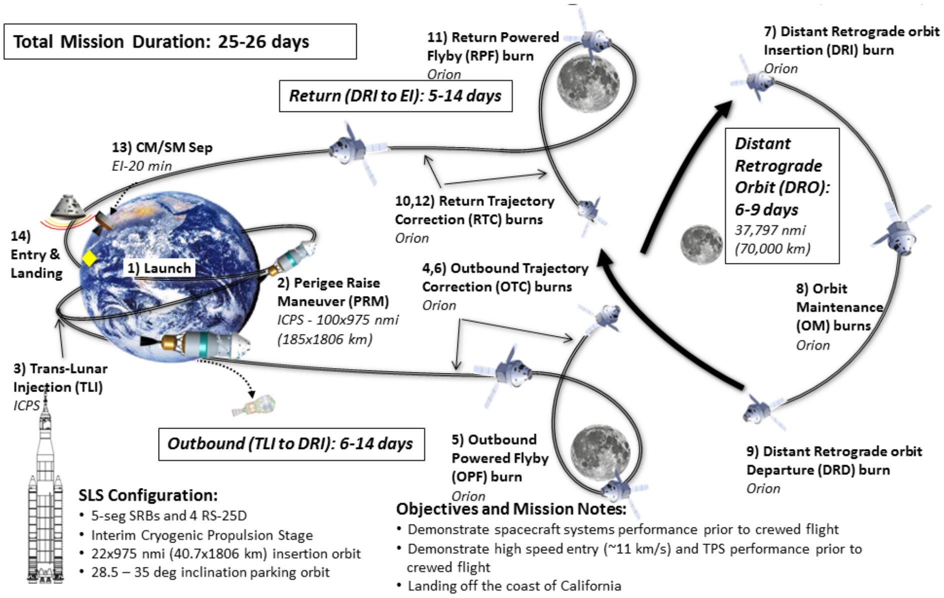

Total Mission Duration: 25-26 days
11) Return Powered Flyby (RPF) burn
Orion
7) Distant Retrograde orbit Insertion (DRI) burn
Orion
Return (DRI to EI): 5-14 days
Distant Retrograde Orbit (DRO): 6-9 days
37,797 nmi (70,000 km)
13) CM/SM Sep
EI-20 min
10,12) Return Trajectory Correction (RTC) burns
Orion
14) Entry & Landing
1) Launch
2) Perigee Raise Maneuver (PRM)
ICPS - 100x975 nmi (185x1806 km)
4,6) Outbound Trajectory Correction (OTC) burns
Orion
8) Orbit Maintenance (OM) burns
Orion
3) Trans-Lunar Injection (TLI)
ICPS
Outbound (TLI to DRI): 6-14 days
5) Outbound Powered Flyby (OPF) burn
Orion
9) Distant Retrograde orbit Departure (DRD) burn
Orion
SLS Configuration:
• 5-seg SRBs and 4 RS-25D
• Interim Cryogenic Propulsion Stage
• 22x975 nmi (40.7x1806 km) insertion orbit
• 28.5 – 35 deg inclination parking orbit
Objectives and Mission Notes:
• Demonstrate spacecraft systems performance prior to crewed flight
• Demonstrate high speed entry (~11 km/s) and TPS performance prior to crewed flight
• Landing off the coast of California

Figure 1. Notional EM-1 Uncrewed Distant Retrograde Orbit (DRO) Mission Profile.

The EM-1 profile will take the Orion capsule and SM to cislunar space on an approximately 25-day mission. The profile is shown in Figure 1.[1] Following liftoff and insertion by the SLS Interim Cryogenic Propulsion Stage (ICPS) into a 975x100 nmi orbit, and within one orbit, the ICPS Trans-lunar Injection (TLI) burn will place Orion on a trajectory towards a flyby of the Moon some 6 days after TLI. As Orion flies past the Moon at 54 nmi (100 km) altitude, it executes its first major burn, the Outbound Powered Flyby (OPF) burn, to a Distant Retrograde Orbit (DRO) target. The OPF is a Lunar Gravity Assisted (LGA) burn. Some 4 days later, Orion executes the Distant Retrograde orbit Insertion (DRI) burn to insert into the DRO. Orion then remains in the DRO for 6 days. After this point, Orion performs a Distant Retrograde orbit Departure (DRD) burn, which targets the Return Powered Flyby (RPF) burn on the way back to Earth. About 4 days after the DRD, Orion executes the RPF burn during a flyby of the Moon, again near 54 nmi (100 km) altitude. Like the OPF, the RPF is an LGA burn. The burn satisfies a specific SM Entry Interface (EI) state of earth longitude, geodetic latitude, altitude, inertial flight path angle, and inertial velocity azimuth at EI to achieve a landing off the coast of California. The flight time from RPF to EI is about 5.5 days. About 20 minutes before EI, the Orion CM separates from the SM, and subsequently executes a small CM-raise burn to achieve the desired EI flight path angle, and provide separation from the SM trajectory before re-entry. The approximate burn ΔVs for the four major EM-1 burns are listed in Table 1.

Table 1. EM-1 Major Burn ΔVs.

Burn	ΔV Magnitude (ft/s)
OPF	515
DRI	432
DRO	446
RPF	828

ORBGUID ARCHITECTURE AND OPERATION

OrbGuid provides burn guidance for all of the burns performed by the Orion CM and SM. OrbGuid's primary function is to provide updates to the vehicle's commanded burn attitude profile and planned engine cutoff time so that the vehicle will meet the desired target conditions at the end of the burn. The OrbGuid algorithm structure derives its heritage from the Space Shuttle insertion/deorbit powered flight guidance.[2, 3] Efforts to unify the various ascent, insertion, on-orbit and deorbit powered guidance phases around a core predictor-corrector algorithm, the Powered Explicit Guidance (PEG) algorithm, led to a flexible framework that could be applied to multiple flight phases and target conditions. OrbGuid takes advantage of PEG's flexible nature to also unify a set of orbital guidance options. Through a menu of various desired velocity options within the predictor/corrector framework, Orion can apply essentially the same guidance algorithm across all of its orbital powered burns such as orbit insertion, rendezvous, deorbit, CM raise burns, externally-specified ΔV burns, lunar transfer, earth-return burns, and aborts.

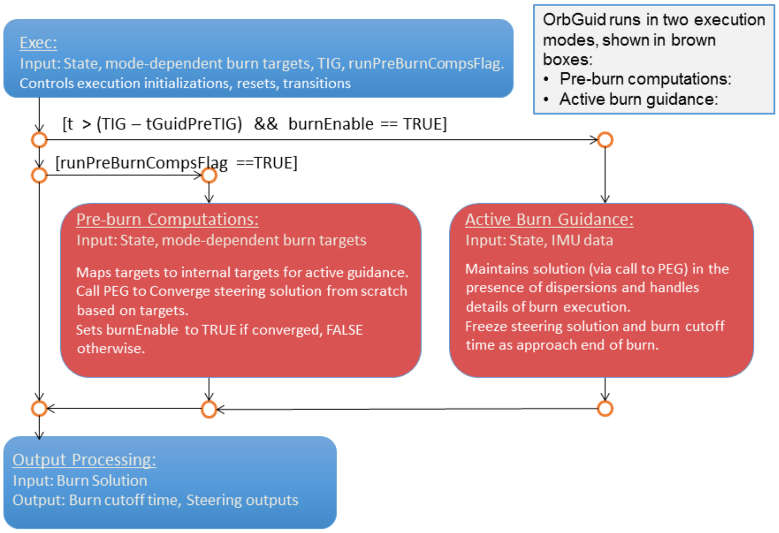

Exec:
Input: State, mode-dependent burn targets, TIG, runPreBurnCompsFlag.
Controls execution initializations, resets, transitions
OrbGuid runs in two execution modes, shown in brown boxes:
• Pre-burn computations:
• Active burn guidance:
[t > (TIG – tGuidPreTIG) && burnEnable == TRUE]
[runPreBurnCompsFlag ==TRUE]
Pre-burn Computations:
Input: State, mode-dependent burn targets
Maps targets to internal targets for active guidance.
Call PEG to Converge steering solution from scratch based on targets.
Sets burnEnable to TRUE if converged, FALSE otherwise.
Active Burn Guidance:
Input: State, IMU data
Maintains solution (via call to PEG) in the presence of dispersions and handles details of burn execution.
Freeze steering solution and burn cutoff time as approach end of burn.
Output Processing:
Input: Burn Solution
Output: Burn cutoff time, Steering outputs

Figure 2. OrbGuid Executive Logic.

OrbGuid consists of two main parts: a top-level internal executive wrapped around PEG and the PEG algorithm itself. Figure 2 shows the flow diagram for the top-level logic. OrbGuid handles initialization and re-initialization by flags passed from an external executive. OrbGuid then executes one of its two main internal modes, pre-burn computation or active guidance, based on the current input time and Time of Ignition (TIG).

Pre-burn computations must be performed long enough before a burn to allow for both validation of the burn solution, and for the vehicle to orient to the desired initial burn attitude. During pre-burn computations, OrbGuid solves for the vector velocity-to-be-gained by thrust ($\vec{v}_{go}$) to achieve the end-of-burn targets. To do that, it first performs a number of variable initializations, including propagation of the current vehicle state to TIG. The PEG algorithm is then called to converge on the burn solution which also yields the steering profile and the desired engine shutdown time.

Active guidance begins at a pre-determined time before TIG, and it begins with the converged solution from the pre-burn computation. During active guidance, PEG takes the vehicle state, either the current vehicle state or the predicted ignition state, whichever is later, and updates the steering and burn solution to maintain convergence to within a specified velocity-miss tolerance.

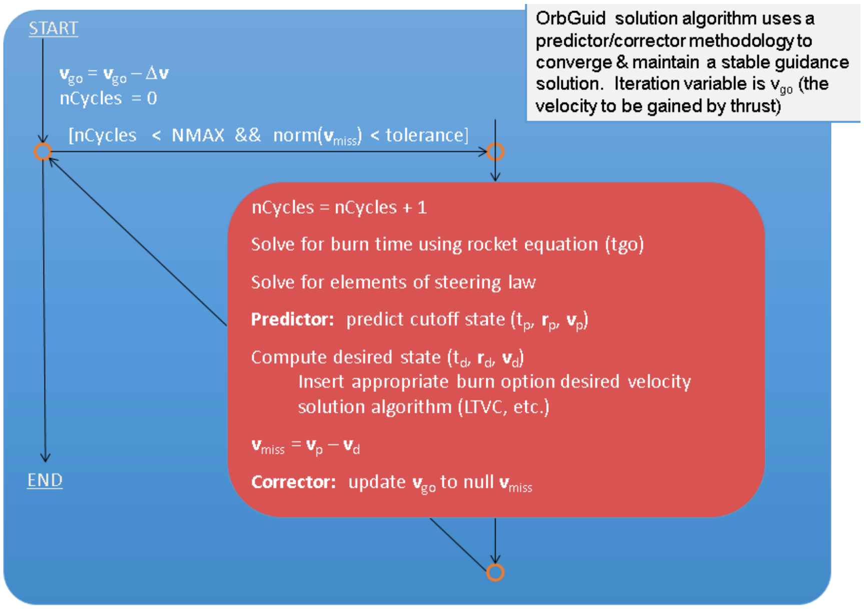
At a pre-determined time interval before desired engine cut-off, OrbGuid holds the steering command direction to avoid large attitude changes as $\vec{v}_{go}$ decreases to a zero magnitude. The control system commands the engine system to shut down at OrbGuid's computed cut-off time. The vehicle holds attitude while the engine system completes its shut-down and the propellant settles. OrbGuid continues to cycle at a 1 Hz rate to provide a continued accurate computation of the residual $\vec{v}_{go}$ required to achieve the burn objectives. Following cleanup of the burn residuals with an optional trim burn using the Reaction Control System (RCS) thrusters, execution of OrbGuid is terminated.

Additional detail on the Orion exo-atmospheric burn guidance architecture and algorithm are available in "Orion's Exo-atmospheric Burn Guidance Architecture and Algorithm."[4] The desired target conditions, including the desired TIG, are provided by either ground uplink or by the onboard Two-Level Targeter (TLT) software.[5] Parameter data on the planned thruster performance and vehicle mass are also part of the target set.

PEG Algorithm Framework

The PEG algorithm performs OrbGuid's primary function, to guide a burn such that the vehicle state at burn cut-off meets the specified target conditions. OrbGuid sets the proper flags and target pre-processing based on the desired burn options and target type, and then executes its core predictor-corrector process to solve for the basis elements of the steering law and burn duration.

PEG is explicit in the sense that it does not require a reference trajectory. It is a semi-analytic algorithm where the form of the steering law allows for certain analytic approximations that enable solutions for elements of that steering law from the current estimate of the velocity-to-be-gained. The prediction of the cut-off state using those steering elements and the estimated burn duration is obtained by a combination of analytic computation of the position and velocity changes due to thrust and a numeric Encke-Beta integration.[6, 7] A correction process is then employed to null the miss between the predicted and desired velocity states at burn cut-off using $\vec{v}_{go}$ as the iteration variable. Over a small number of iterations, the algorithm converges on the $\vec{v}_{go}$, and the burn steering profile which take the vehicle to the desired velocity condition at burn cut-off that satisfies the orbit transfer objectives. A brief synopsis of the algorithm structure for orbit applications follows.


START
$\mathbf{v}_{go} = \mathbf{v}_{go} - \Delta\mathbf{v}$
nCycles = 0
[nCycles < NMAX && norm($\mathbf{v}_{miss}$) < tolerance]
nCycles = nCycles + 1
Solve for burn time using rocket equation (tgo)
Solve for elements of steering law
Predictor: predict cutoff state (t_p, $\mathbf{r}_p$, $\mathbf{v}_p$)
Compute desired state (t_d, $\mathbf{r}_d$, $\mathbf{v}_d$)
Insert appropriate burn option desired velocity solution algorithm (LTVC, etc.)
$\mathbf{v}_{miss} = \mathbf{v}_p - \mathbf{v}_d$
Corrector: update $\mathbf{v}_{go}$ to null $\mathbf{v}_{miss}$
END

OrbGuid solution algorithm uses a predictor/corrector methodology to converge & maintain a stable guidance solution. Iteration variable is v_{go} (the velocity to be gained by thrust)

Figure 3. PEG Top-Level Logic.

Figure 3 illustrates the top level PEG executive. At the heart of the PEG algorithm lies the predictor-corrector loop. This loop exits either when v_{go} converges to a specified tolerance or the number of cycles exceeds a preset limit.

For a given estimate of $\vec{v}_{go}$ the burn time is derived from the rocket equation and the elements of the steering law are computed. With these parameters, PEG's predictor subroutine predicts the vehicle state at burn-cutoff.

The corrector task calls the specified desired velocity routine based on the burn option given the predicted burn cut-off state. For orbital maneuvers, there is no explicit control of the burn cut-off position, and the desired velocity routine determines the velocity that will allow the vehicle to meet the target conditions from that cut-off position.

OrbGuid's versatility comes mostly through the different desired velocity routines in the PEG corrector, each being related to a specific guidance option. The extensible nature of the PEG framework provides the ability to readily add new guidance options as the need arises. The current suite of burn options are detailed in the next section.

DESIRED VELOCITY ROUTINES

OrbGuid's unique guidance elements lie mainly in its desired velocity routines. Orbit guidance typically does not control position at burn cut-off, so the corrector reduces to controlling velocity. For a particular burn, the target burn option parameter controls which routine to execute, and the target set contains the parameters necessary to evaluate a solution. Another parameter, the plane control option, controls how OrbGuid handles the out-of-plane burn component. The separation of in-plane and out-of-plane components is essential near the 180° transfer singularity, but it also enables OrbGuid to protect against propellant and coasting time limits or to target the plane of the landing site in the case of a deorbit burn.

A description of the burn options currently implemented within OrbGuid is presented in the following subsections. All but one of the burn options are solutions to intercept problems, which involves finding the initial velocity ($\vec{v}_d$) which effects a transfer between two given positions ($\vec{r}_p$ and $\vec{r}_T$) subject to some constraint.

1. External Delta Velocity

An external source, such as a separate targeting routine or an uplink from Mission Control, provides a desired velocity change to be achieved by thrust, the External ΔV vector, which is then executed by OrbGuid. This is the simplest burn option within OrbGuid and is the one non-intercept burn option. It will likely be used on the planned trajectory correction burns en-route to and from the Moon, and for orbit maintenance while in the DRO. OrbGuid provides the option to specify the External ΔV vector either in an inertial coordinate frame or a "Local Vertical, Local Horizontal" (LVLH) coordinate frame located at burn ignition. The latter format provides for easier burn visualization and in some situations is less sensitive to variation in burn ignition.

External ΔV guidance begins with $\vec{v}_{go}$ being initialized to the External ΔV vector from the input target set. The subsequent burn prediction yields the predicted cutoff velocity ($\vec{v}_p$). To remain consistent with the PEG predictor-corrector framework, the desired velocity routine reduces to the assignment $\vec{v}_d = \vec{v}_p$. This equality yields convergence on the first iteration. Implementing External ΔV into PEG this way reduces the need for additional guidance software.

2. Linear Terminal Velocity Constraint

In a search for a generalized targeting algorithm, particularly to support deorbits from Low Earth Orbit (LEO), the Space Shuttle Program developed the Linear Terminal Velocity Constraint (LTVC) transfer where a linear constraint was imposed between the radial (v_r) and horizontal (v_h) velocities at the target found in Equation (1).

$$v_r = c_1 + c_2 v_h \tag{1}$$

The historical background of the LTVC transfer is provided by Bond and Allman.[8] The LTVC option was used for Space Shuttle orbit insertion (the OMS-1 and OMS-2 burns) and the deorbit burn.

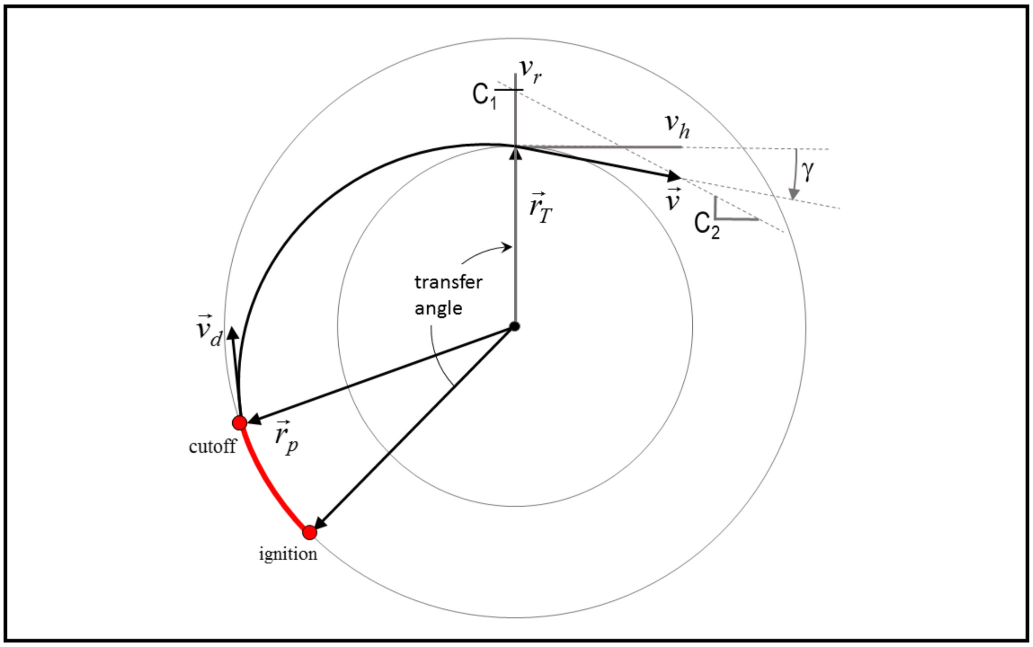


Figure 4. LTVC Geometry.

The LTVC algorithm solves for the initial velocity to intercept a downrange position vector target while constraining the radial and horizontal velocity components, at intercept, by the linear relationship. Figure 4 illustrates the LTVC problem. The target position can be specified as an inertial vector or as a combination of altitude at the target and transfer angle from TIG.

The solution is based on the well-known fact that a conic velocity can be represented as the sum of two vectors of constant length, one perpendicular to the position vector and one perpendicular to the major axis. This yields the following expressions for the radial and horizontal components of velocity.

$$v_r = \frac{\mu e}{h}\sin(f) \qquad (2) \qquad\qquad v_h = \frac{\mu}{h} + \frac{\mu e}{h}\cos(f) \qquad (3)$$

This means that the desired trajectory is completely specified by a knowledge of the radial and horizontal components of velocity at some point along the trajectory. Given Equation (1), one other equation is required to relate these two quantities. That equation is obtained by using Equation (3) at the initial position and relating the time varying quantities to those at the target. The horizontal velocity is readily mapped using the conservation of angular momentum, and true anomaly by the transfer angle ($f_p = f_T - \theta$ where θ is measured from burn cutoff). With $r_d = r_p$ (since burn cutoff position is not explicitly controlled), expansion of the cosine of a difference ($\cos(f_T - \theta)$), and using Equations (2) and (3) to eliminate f_T , yields the needed equation in the form of a quadratic in $v_{T,h}$.

$$\left[\frac{r_T}{r_d}-\cos(\theta)\right](v_{T,h})^2 - v_{T,r}\sin(\theta)v_{T,h} - \frac{\mu}{r_T}(1-\cos(\theta)) = 0 \qquad (4)$$

Simultaneous solution of Equations (1) and (4) is obtained by substituting (1) into (4), yielding the following quadratic in the unknown $v_{T,h}$, where $k = \frac{r_T - r_d}{r_d}$ and $w = \cot\frac{\theta}{2} = \frac{\sin\theta}{1-\cos\theta}$.

$$[k(1+w^2)+2(1-wc_2)](v_{T,h})^2 - 2wc_1 v_{T,h} - \frac{2\mu}{r_T} = 0 \qquad (5)$$

The quadratic solution is then given by the following equations.

$$a = k(1+w^2)+2(1-wc_2) \qquad (6)$$

$$b = wc_1 \qquad (7)$$

$$c = \frac{2\mu}{r_T} \qquad (8)$$

$$d = b^2 + ac \qquad (9)$$

$$v_{T,h} = \frac{c}{\pm\sqrt{d} - b} \qquad (10)$$

$$v_{T,r} = c_1 + c_2 v_{T,h} \qquad (11)$$

Using relationships of velocity components between two positions on a trajectory provides the desired solution at the burn cutoff position.

$$v_{d,h} = (1+k)v_{T,h} \qquad (12)$$

$$v_{d,r} = k\,w\,v_{T,h} - v_{T,r} \qquad (13)$$

It was shown by Shepherd that the plus sign in Equation (10) is the most meaningful from a guidance point of view, and is used in the LTVC solution.[9] The LTVC transfer has been applied to the deorbit problem where the required flight-path angle at EI is dependent on the velocity magnitude. Over a limited region, this relationship can be linearized between the radial and horizontal components of velocity. In the trivial case where the entry flight-path angle is fixed, the constraint line is given by $c_1 = 0$ and $c_2 = \tan(\gamma)$. In practice, the two constraints are chosen to control velocity dispersions at EI. Additionally, LTVC works well for the orbit insertion and orbit transfers located at the apsides where the velocity at the desired intercept (apsis) altitude is horizontal, which corresponds to $c_1 = c_2 = 0$.

For low eccentricity orbits, OrbGuid modifies the above equations analytically to account for the J_2 gravity perturbation. Lineberry derived an analytic solution based on mean element theory for in-plane precision LTVC.[10, 11] McHenry derived a formulation for both in-plane and out-of-plane J_2 perturbations.[12] Both solutions result in a minimal increase in algorithmic complexity over the conic formulation.

3. Free-Range LTVC

Before entry the Orion CM executes a short burn after separation from the SM. This “CM-raise” burn is designed to achieve a slightly shallower flight-path angle at EI to gain separation from the SM before entry. Use of the LTVC option for this burn is problematic for the near-parabolic Earth return trajectory from the Moon due to the extreme sensitivity of the impulse to the in-plane transfer angle. This can be seen by implicitly differentiating Equation (4) with respect to θ to find the partial derivative.

$$\frac{\partial v_{T,h}}{\partial \theta} = \frac{\left[\sin(\theta) - c_2\cos(\theta)\right] v_{T,h}^2 - c_1\cos(\theta)\, v_{T,h} - \frac{\mu}{r_T}\sin(\theta)}{2\left[\cos(\theta) + c_2\sin(\theta) - \frac{r_T}{r_d}\right] v_{T,h} + c_1\sin(\theta)} \tag{14}$$

The sensitivities of the desired radial and horizontal velocity are now found by using the chain rule to differentiate Equations (12) and (13).

$$\frac{\partial v_{d,h}}{\partial \theta} = \frac{r_T}{r_d}\frac{\partial v_{T,h}}{\partial \theta} \tag{15}$$

$$\frac{\partial v_{d,r}}{\partial \theta} = \left[\left(\frac{r_T}{r_d} - 1\right)\frac{\sin(\theta)}{1-\cos(\theta)} - \frac{r_T}{r_d}\right]\frac{\partial v_{T,h}}{\partial \theta} + \left(\frac{r_T}{r_d} - 1\right)\frac{v_{T,h}}{\cos(\theta)-1} \tag{16}$$

Despite the complexity of these equations, a few salient observations can be made: 1) as the ratio of the target radius to the position radius departs unity, the sensitivity of the desired radial velocity to the transfer angle increases; 2) a singularity in the sensitivity of the desired radial velocity exists for transfer angles that are integer multiples of 180 degrees; and 3) both the desired radial and horizontal sensitivities are proportional to the horizontal velocity at the target point. For the EM-1 CM raise burn these effects combine to produce very high sensitivities (approximately 330 m/s per degree root sum squared). Hence, any shift in the effective transfer angle due to finite burn effects (via the burn arc length subtracting from the ignition-relative transfer angle) results in large variations in the velocity-to-be-gained direction and magnitude, potentially injecting instability and inhibiting convergence. Even in the event of successful convergence, the resulting burn may be prohibitively large.

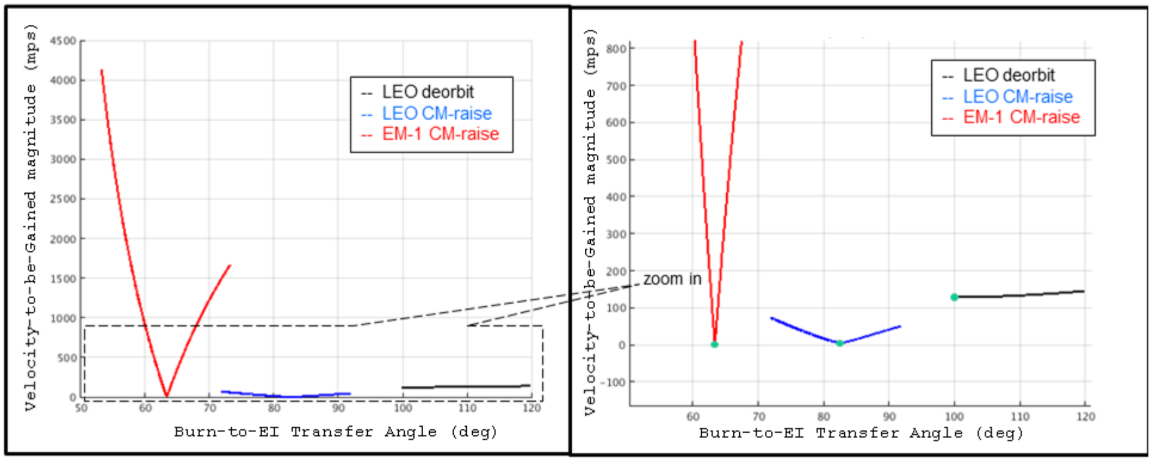


Figure 5. LTVC Impulse Sensitivity to Transfer Angle.

This sensitivity to variation in transfer angle is illustrated in Figure 5 which compares the LTVC impulse in three scenarios: the Earth return from the Moon CM-raise, a LEO deorbit, and a LEO CM-raise burn. An alternate interpretation of these results is that the entry point on the Earth return trajectory is insensitive to the impulse magnitude this late in the approach trajectory. For the EM-1 CM-raise example a burn of nearly 300 m/s oriented to maintain EI conditions will only shift the transfer angle by a single degree. This issue was first raised and a resolution proposed to not specify the transfer angle (i.e., the intercept location) and allow the algorithm to solve for the transfer angle that yields the minimum burn impulse.[13] This variant of the LTVC problem is termed the Free-Range LTVC (FR-LTVC) problem, and has been implemented as a new guidance option to provide explicit closed-loop control of the EI flight-path angle by the CM-raise burn.

The FR-LTVC solution equation is obtained by considering the following vis-viva integral, with *a* defining the orbital semi-major axis.

$$\frac{1}{a}=\frac{2}{r}-\frac{v^2}{\mu} \tag{17}$$

Using Equation (17), since the semi-major axis is constant on the post-burn coasting trajectory, the velocities at burn cutoff and target intercept can be related to one another (again with $r_d = r_p$ since burn cutoff position is not explicitly controlled), satisfying the relationship in Equation (18).

$$v_d^{\ 2} = 2\mu\left(\frac{1}{r_d}-\frac{1}{r_T}\right)+v_T^{\ 2} \tag{18}$$

Breaking up the magnitudes into vertical and horizontal components, and substituting Equation (1) for the vertical component at intercept, yields Equation (19).

$$v_{d,r}^{\ 2}+v_{d,h}^{\ 2} = 2\mu\left(\frac{1}{r_d}-\frac{1}{r_T}\right)+\left[(c_1+c_2 v_{T,h})^2+v_{T,h}^{\ 2}\right] \tag{19}$$

Since the orbital angular momentum ($h = r\,v_h$) is a constant on the coasting trajectory from r_d to r_T, then the horizontal velocity at burn cutoff and intercept can be related using Equations (20) and (21).

$$v_{T,h} = \rho\, v_{d,h} \tag{20}$$

$$\rho = \frac{r_d}{r_T} \tag{21}$$

Substituting Equation (20) into Equation (19) yields the FR-LTVC constraint equation.

$$\frac{2\mu}{r_d}(1-\rho)+(c_1+c_2\rho\, v_{d,h})^2+(\rho^2-1)v_{d,h}^{\ 2}-v_{d,r}^{\ 2}=0 \tag{22}$$

The solution to the free-range LTVC problem involves trying to minimize Equation (23), with subscripts *c* and *d* denoting the current and desired (cutoff) velocity components, respectively.

$$v_{go} = \sqrt{(v_{d,r} - v_{c,r})^2 + (v_{d,h} - v_{c,h})^2} \tag{23}$$

The optimization problem to be solved is to minimize Equation (23) (or rather its square) subject to the constraint in Equation (22). Observe that Equation (22) defines a conic section which constrains the in-plane desired velocity. Therefore, this is a problem of finding a point on a conic section (v_{dh}, v_{dr}) nearest the point ($v_{c,h}$, $v_{c,r}$). This insight allows for an efficient numeric solution that is stable, robust, and exhibits desirable convergence properties. For a detailed analysis of this problem and its robust numeric solution, including a useful method of normalizing the problem, please refer to *Some Geometric Relations and a Proposed Solution Process for the Free Range Linear Terminal Velocity Constraint Problem* by S. Robinson.[14] What follows is only a brief description outlining the solution approach.

Use a Lagrange multiplier to adjoin the constraint from Equation (22) to the square of Equation (23).

$$J = (v_{d,h} - v_{c,h})^2 + (v_{d,r} - v_{c,r})^2 + \lambda\left[\frac{2\mu}{r_d}(1-\rho) + (c_1 + c_2 v_{d,h})^2 + (\rho^2 - 1)v_{d,h}^2 - v_{d,r}^2\right] \tag{24}$$

Now take the partials with respect to $v_{d,h}$ and $v_{d,r}$ and set them equal to zero.

$$\frac{\partial J}{\partial v_{d,h}} = 2(v_{d,h} - v_{c,h}) + 2\lambda\left[c_1 c_2 + (c_2^2 + \rho^2 - 1)v_{d,h}\right] = 0 \tag{25}$$

$$\frac{\partial J}{\partial v_{d,r}} = 2(v_{d,r} - v_{c,r}) - 2\lambda v_{d,r} = 0 \tag{26}$$

These equations are solved to find the components of the desired velocity.

$$v_{d,h} = \frac{v_{c,h} - \lambda c_1 c_2}{1 + \lambda(c_2^2 + \rho^2 - 1)} \tag{27}$$

$$v_{d,r} = \frac{v_{c,r}}{1 - \lambda} \tag{28}$$

These relations are used in conjunction with a guarded Newton-Raphson iteration loop to find the value of λ which satisfies the constraint in Equation (22). Extensive offline testing has shown that this approach is a robust solution to the problem with desirable convergence characteristics.[14]

The left plot in Figure 6 shows the results of a quick convergence study done with one hundred million randomly generated cases. In this study all cases converged in less than 31 iterations. The right plot in Figure 6 shows the observed cumulative distribution function for the number of iterations. More than 99.9% of the random cases converged within 15 iterations. The cases which required more than 15 iterations for convergence all involved such large changes in velocity that it is difficult to imagine these cases occurring in an actual orbital scenario. More specifically, the cases with large numbers of iterations all involved initial conditions near the lines of symmetry for the constraint conic defined by Equation (22). Initial points lying directly on the lines of symmetry may not have unique solutions. For cases near reasonable mission profiles, the convergence is rapid and well behaved. For example, in the case of a notional CM raise burn from the EM-1 flight profile, the free-range targeting algorithm converged in only 3 iterations to the correct solution.

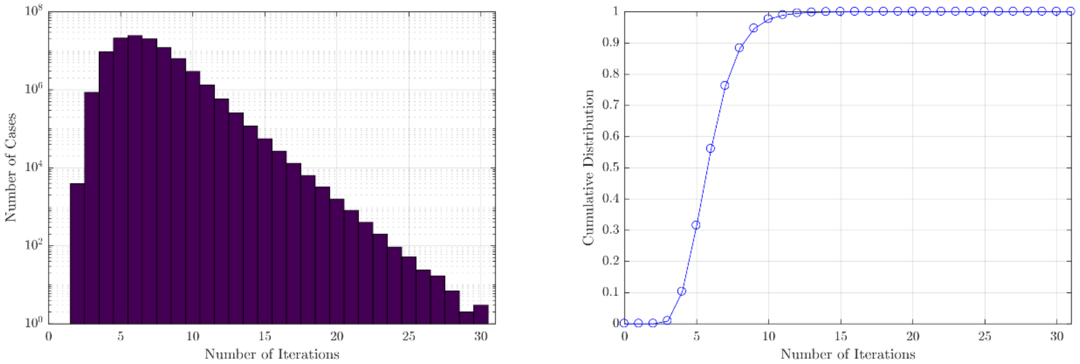


Figure 6. Number of Iterations Needed for Convergence (Left) and Cumulative Distribution Function (Right).

4. Transit Guidance Option

The Transit guidance option uses an algorithm that solves the classical Lambert time-of-flight point-to-point transfer. The iteration parameters used by the Orion Lambert algorithm were first published by Lancaster, Blanchard, and Devaney in 1966, and are valid for elliptical, parabolic, and hyperbolic transfers.[15, 16] Their work served as the basis for the on-board Lambert algorithm developed for Space Shuttle rendezvous burn targeting in the late 1970s. This algorithm, valid for elliptical and circular transfers, flew on the Space Shuttle from 1984 to 2011.[2]

In 1968 Battin published an independent development of the same iteration parameters but with a different time-of-flight equation that used a hypergeometric function.[17] Later, Battin published an improvement to his time-of-flight equation.[18, 19] After performing a survey of Lambert solution methods, Gooding wrote in 1990 that "the outstanding contribution to the subject has been the pair of papers by Lancaster and his colleagues."[20] Gooding developed a more comprehensive algorithm using the Lancaster, Blanchard, and Devaney iteration parameters. This included algorithms for estimating the first guess of the independent iteration parameter and use of a Halley iteration that provides third order convergence. Gooding's new developments achieved a desired level of solution accuracy throughout the iteration space in a fixed number of iterations.

In 2007 Gooding's FORTRAN algorithm was coded in MATLAB to serve in the Rendezvous Targeting (RTARG) software for Orion in the Constellation Program. This algorithm underwent extensive convergence characteristics testing in 2011. The RTARG software was later removed from the Orion software after the Constellation Program was canceled. In the spring of 2016, work began on a Lambert algorithm for the Orion OrbGuid Transit option. The MATLAB implementation of Gooding's algorithm was chosen since it had been tested for Orion during the Constellation Program. The non-dimensional nature of the iteration parameters permit generic testing of the iteration scheme over a wide range of values without defining a trajectory in terms of position and velocity vectors. In addition, the elliptical portion of the iteration space was familiar to personnel due to convergence characteristics testing conducted on the Space Shuttle on-board Lambert algorithm in 1993 and 1994.

The MATLAB algorithm underwent extensive modification to replace decades old FORTRAN coding practices with code that met Orion software standards. The multi-revolution logic was removed since it was not needed for exploration missions. Part of Gooding's starter value algorithm was replaced by a development of Izzo, and the Halley iteration was replaced with a Householder iteration.[21] The algorithm can compute elliptical, parabolic, and hyperbolic transfers.

To further simplify the algorithm, Battin's universal time-of-flight Equation (36) was used instead of Gooding's time-of-flight equations.[18] In the following equations the magnitudes of the initial and final position vectors are r_1 and r_2, and c is the chord of the transfer. The independent iteration parameter x (Equation [30]) is a function of the semi-major axis a of the transfer orbit and the semi-major axis of the minimum energy transfer a_m in Equation (29). The transfer geometry parameter λ (Equation [31], also called q) is a function of the semi-parameter of the transfer triangle s (Equation [29]), the transfer angle θ, and the magnitudes of the initial and final position vectors r_1 and r_2. The Orion implementation of Equation (35) uses the hypergeometric function evaluation software developed for the OrbGuid Encke-Beta trajectory predictor.[7] The initial and final times of the transfer are represented by t_1 and t_2 (Equation [36]).

$$s = 2a_m = 0.5(r_1 + r_2 + c) \qquad (29) \qquad\qquad x = \sqrt{1 - \frac{a_m}{a}} \qquad (30)$$

$$\lambda = \frac{1}{s}\sqrt{r_1\, r_2}\, cos\frac{\theta}{2} \qquad (31) \qquad\qquad y = \sqrt{1 - \lambda^2(1 - x^2)} \qquad (32)$$

$$\eta = y - \lambda x \qquad (33) \qquad\qquad S_1 = 0.5(1 - \lambda - x\eta) \qquad (34)$$

$$Q = \frac{4}{3}\, F(3{,}1{,}\frac{5}{2}{,}S_1) \qquad (35) \qquad\qquad \sqrt{\frac{\mu}{a_m^3}}\,(t_2 - t_1) = \eta^3 Q + 4\lambda\eta \qquad (36)$$

Testing of the Lambert algorithm was performed at several levels. Lambert was exercised during tests of the integrated Orion guidance, navigation, and control software. These tests involved flying portions of the EM-1 mission profile.

Extensive off-line testing of the Lambert algorithm was also performed. This permitted tests to be created that exercised the algorithm over a wider range of transfers than the integrated guidance, navigation, and control software tests. Furthermore, the off-line tests could collect more data and execute for longer times. Internal data not captured in integrated testing could be collected in off-line testing. The first type of off-line Lambert testing was executed in conjunction with testing of the Kepler algorithm used by the Encke-Beta predictor.[7] These tests exercised the complete Lambert algorithm by input of the transfer time and the initial and final position vectors. Test results have compared favorably with a different Lambert algorithm developed by Arora and Russell.[22]

The second type of off-line testing focused on determining the convergence characteristics of the iteration scheme. Logic concerning starter values for the independent iteration parameter, the Householder iteration, the time-of-flight equation, and the convergence test was exercised. Inputs to these tests included the normalized desired transfer time (left side of Equation [36]) and the transfer geometry parameter λ. While this level of testing does not exercise all of the Lambert logic, it does more extensively test the convergence characteristics of the iteration scheme. This type of off-line testing was originally performed during the Space Shuttle Program in 1993 and 1994 in response to a Lambert targeting problem that occurred during the STS-49 mission and another that occurred four days before the launch of STS-51.[23]

Figure 7 illustrates where the four EM-1 Transit burns are located in the Lambert iteration space. The horizontal axis is the independent parameter x and it is a function of the desired semi-major axis of the transfer (Equation [30]). $x = 0$ is a minimum energy elliptical transfer, -1 < x <

+1 represents elliptical transfers, $x = 1$ is a parabolic transfer, and x > +1 represents hyperbolic transfers. The circular transfer locus is also shown. The other lines on the plot represent the transfer geometry parameter q (Equation [31], also called λ). q = +1 is a 0 degree transfer, q = 0 is a 180 degree transfer, and q = -1 is a 360 degree transfer. The vertical axis is the normalized time-of-flight T (left side of Equation [36]).

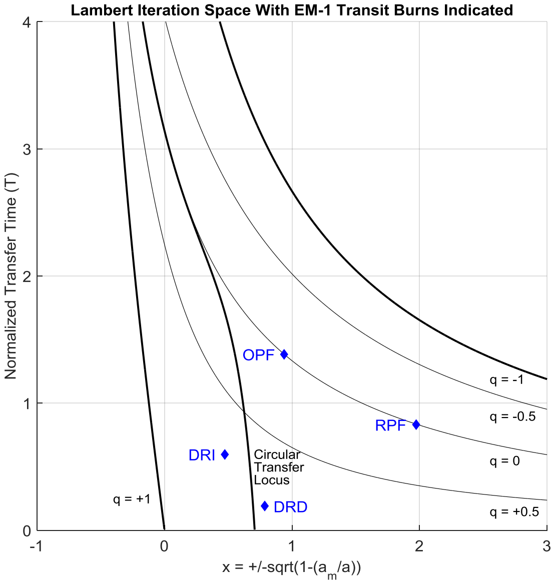


Figure 7. Lambert Iteration Space with EM-1 Transit (Lambert) Burns Labeled.

Figure 8 is a plot of the number of iterations required for the algorithm to converge on a solution for the independent parameter x given a desired normalized transfer time T and a transfer geometry parameter q. In all cases the iterations are complete in three or less iterations. Most cases converge in two iterations (blue circles). Cases of convergence on the first iteration (green circles) occur for some parabolic transfers ($x = 1$).

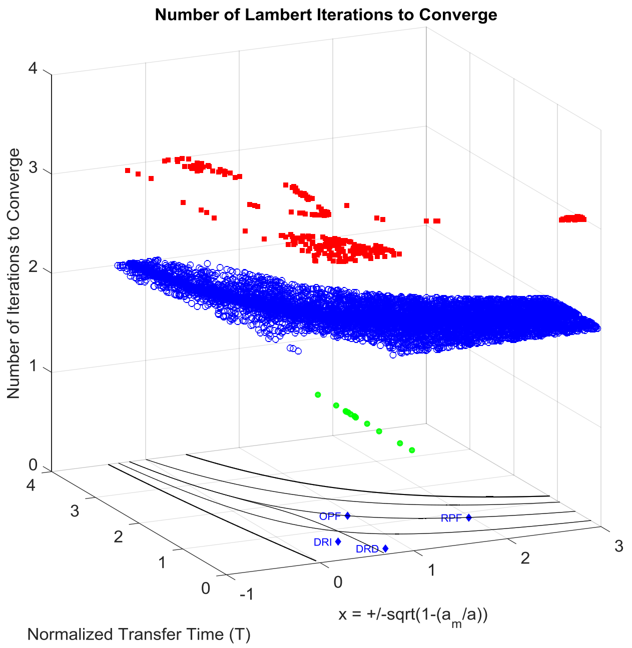


Figure 8. Lambert Iteration Space Number of Iterations to Converge.

5. Constrained Intermediate Terminal Intercept (CITI) mode

The CITI algorithm had its genesis during the Apollo Program with the return-to-Earth targeting to achieve a specified flight-path angle at EI, but was not used due its late development. It was derived by William M. Robertson, although the name was only recently applied.[24, 25]

This algorithm has application to intercept problems requiring a constraint on the flight path angle at an intermediate point. The algorithm finds the velocity required to intercept a target vector while achieving a desired flight-path angle at an intermediate altitude as illustrated in Figure 9. The fundamental variable in the solution is the flight path angle at cutoff (γ_d). Marscher derived two equations for all conic transfers between to arbitrary radii.[26] With burn cutoff being one radii (again with $r_d = r_p$ since burn cutoff position is not explicitly controlled), these relations are shown in Equations (37) and (38).

$$\frac{P}{r_d}=\frac{1-\cos\theta}{\frac{r_d}{r}-\cos\theta+\Gamma_d\sin\theta} \qquad (37)$$

$$\frac{P}{r_d}=\frac{2(\frac{r_d}{r}-1)}{(\frac{r_d}{r})^2(1+\Gamma^2)-(1+\Gamma_d{}^2)} \qquad (38)$$

P is the semi-latus rectum, θ is the transfer angle from r_d to r, Γ_d is the tangent of γ_d, r is an arbitrary radius, and Γ is the tangent of the flight-path angle at r. Evaluating Equation (37) at the intercept position (r_T) and Equation (38) at the intermediate position (r_i) and eliminating P/r_d yields the following quadratic in Γ_d.

$$\Gamma_d{}^2 + 2(\frac{r_d}{r_i}-1)\cot(\frac{\theta}{2})\Gamma_d + \left[1-(\frac{r_d}{r_i})^2(1+\Gamma_i{}^2)+\frac{2}{1-\cos\theta}(\frac{r_d}{r_i}-1)(\frac{r_d}{r_T}-\cos\theta)\right]=0 \tag{39}$$

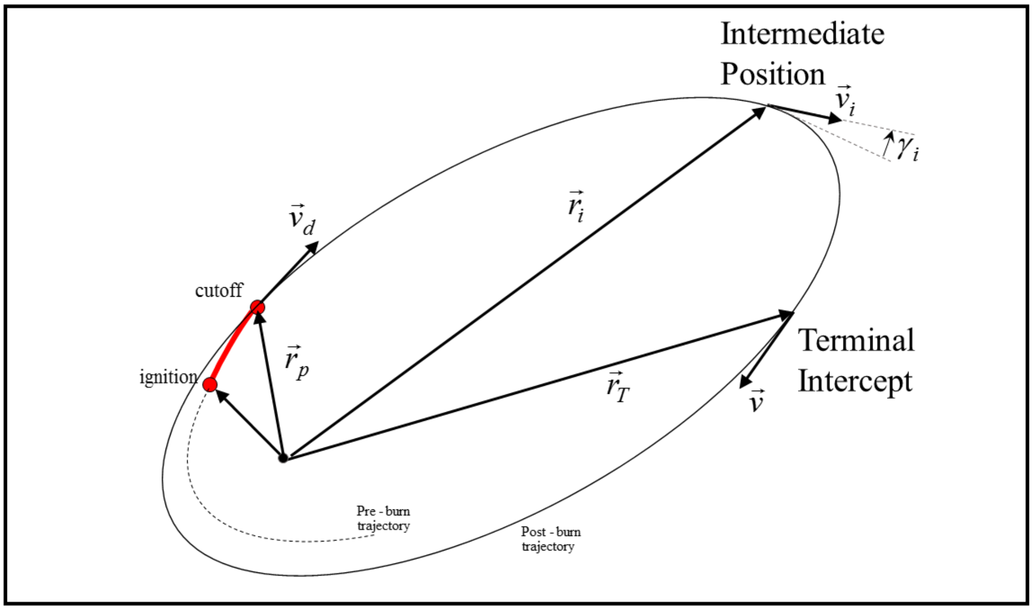


Figure 9. CITI Transfer Geometry.

Once Γ_d is determined, substitution into Equation (37) at the intercept position yields the normalized semi-latus rectum. Substitution of this result into another equation derived by Marscher yields the normalized semi-major axis.[26]

$$\frac{r}{a} = 2 - \frac{P}{r}(1+\Gamma^2) \tag{40}$$

The desired velocity magnitude at burn cutoff can now be obtained from the vis-viva integral Equation (17), completing determination of the desired velocity vector at burn cutoff. Details of the quadratic roots are provided on the basis of the relationship between r_d , r_i, and r_T , six regions in all.[24] The scenarios of current interest for Orion are $r_i > r_d > r_T$, and $r_d > r_i > r_T$. The first scenario applies to lunar and Earth orbit maintenance burns, while the second applies to direct return aborts during the outbound leg to the Moon and deorbit burns from LEO. An example of an Earth orbit problem, that parallels the LTVC solution, is a deorbit burn where an Earth-surface target is defined for ballistic intercept while specifying the desired flight path angle at the EI altitude of 400 kft.

HIGH-ORDER GRAVITY TARGET BIASING

Orion's exploration mandate necessitates higher autonomy. OrbGuid's high-order gravity option takes advantage of increased computing power to bias guidance targets on-board to take into account higher order gravitational harmonics and third body perturbations. This biasing is accomplished using a single prediction of the conic transfer solution from burn cutoff per each evaluation of the conic transfer solutions. The solution is predicted to the pierce plane that contains the target intercept position vector and is perpendicular to the orbit plane. The target vector is then biased by subtracting the miss vector. After several predictor/corrector guidance iterations, the conic burn solution with the biased targets rapidly converges to the high-order predicted solution.

Additional biases are also computed depending on the guidance option. For the two LTVC options, the linear slope constants Equation (1) are biased based on the radial velocity miss at intercept. For the transit option, the conic transfer time is biased by the transit-time miss from the desired transit time. For CITI, the propagation from burn cutoff to intercept is broken into two segments. The first segment propagates to the intermediate position vector to obtain the bias of the intermediate radius magnitude and intermediate flight-path angle. The second propagation segment provides for biasing the target intercept vector. An illustration of the position and constraint biasing is provided in Figure 10 for the LTVC solution.

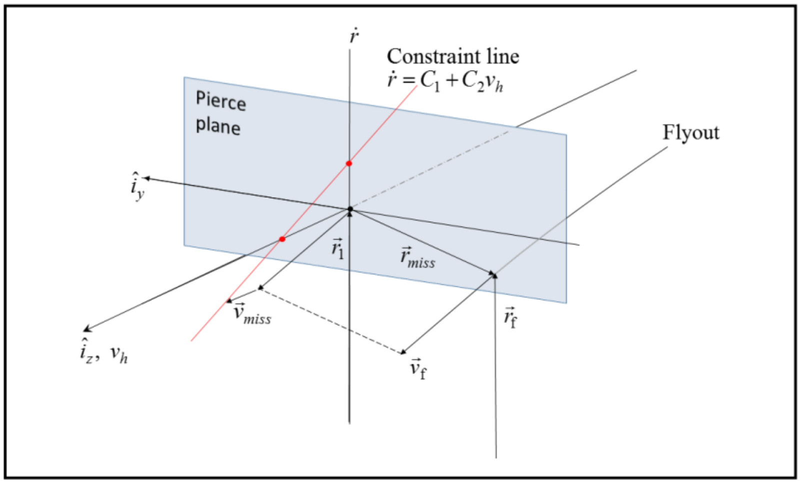


Figure 10. Intercept Target Biasing for Non-Central Body Effects.

OUT-OF-PLANE CONTROL

OrbGuid provides several options for specifying the objective of the out-of-plane component for the desired intercept velocity solutions: in-plane only, target intercept, velocity null, and Earth-fixed target plane intercept. Solving for the in-plane-only solution is useful for 180° transfers and cases where maintenance of the orbit plane is not important, and it is accomplished by projecting the target vector into the orbit plane. Target intercept is accomplished by adding an out-of-plane velocity component that nulls the out-of-plane position miss at intercept found by the predicted solution. The following equation for the out-of-plane velocity miss is derived from basic trigonometric identities.

$$v_{miss,y} = \frac{v_{horiz}\, r_{T,miss,y}}{r_T \sin\theta} \tag{41}$$

In this equation v_{horiz} is the horizontal component of the velocity at cutoff, r_T is the target intercept radius, θ is the central transfer angle from cutoff to the target, and $r_{T,miss,y}$ is the out-of-plane position miss at target intercept.

Out-of-plane velocity null is useful for partially correcting the orbit plane with a single burn, and it is accomplished by projecting the desired velocity into the desired orbit plane. The position remains uncorrected, but the plane error is limited to the current out-of-plane position miss.

Finally, the Earth-fixed target plane intercept is useful for bringing the orbit plane over a target such as a landing site. It applies to the LTVC and CITI burn options. This method requires an estimate of the elapsed time between downrange target intercept and the planar intercept target. Using this parameter, the planet-fixed vector can be converted into inertial coordinates, and the downrange intercept target moved into the appropriate orbit plane by using Equation (41) to relate the out-of-plane velocity correction required to null the cross-track position miss at landing.

In addition to directly specified out-of-plane intercepts, OrbGuid has options to protect against minimum mass or free-fall time constraints. Minimum mass includes propellant, and the mass protection prevents Orion from exceeding propellant reserves in order to meet out-of-plane targets. Similarly, a certain amount of free-fall time is required to accomplish docking mechanism jettison, SM jettison, CM orientation, and the CM raise burn (if required). In the case of a propulsion system downmode, these protections allow for an immediate downmode while allowing the vehicle to achieve the smallest out-of-plane possible while preserving the critical in-plane component.

OrbGuid accomplishes both mass and free-fall time protection by scaling the $\vec{v}_{go}$ magnitude to account for the limit. For mass, the maximum v_{go} magnitude is given by Equation (42).

$$v_{go,\max,mass} = v_{ex} \ln \frac{m}{m_{burnout,\min}} \tag{42}$$

For free-fall time, if t_{go} defines the burn time and *tti* defines the coast time from burn cutoff to intercept, then $t_{\text{go}}+tti$ defines the elapsed time to intercept, and $t_{\text{go}}+tti\text{-}tti_{\min}$ defines the allowable burn time that satisfies the minimum free-fall time. Using the rocket equation (43), the maximum v_{go} is given by Equations (44) and (45).

$$v_{go} = -v_{ex} \ln(1 - \frac{\Delta t}{\tau})) \tag{43}$$

$$tti_{\arg} = 1 - \frac{(t_{go} + tti) - tti_{\min}}{\tau} \quad (44) \qquad\qquad v_{go,\max,tti} = -v_{ex} \ln(tti_{\arg}) \quad (45)$$

The maximum v_{go} magnitude is the lesser of the mass and time-to-intercept values. If the unconstrained v_{go} is greater, and the in-plane burn is achievable, then the out-of-plane burn component is given by Equation (46).

$$v_{go,\max,oop} = sign(v_{go,oop})\sqrt{v_{go,\max}^2 - v_{go,ip}^2} \tag{46}$$

SUMMARY

For NASA’s future manned exploration missions, the Orion Multi-Purpose Crew Vehicle requires a burn guidance capability that is able to support the current exploration missions and that is adaptable to future, as yet to be defined, exploration mission profiles. The orbit powered flight guidance algorithm design described in this manuscript leverages many decades of successful flight heritage design with proven utility and reliability while incorporating lessons learned in the areas of advanced precision, autonomy, and architecture design. The result is an algorithm and suite of burn options that support a wide variety of orbit transfer scenarios. Yet the design is extendable to new burn options, as demonstrated by the relatively recent addition of the free-range LTVC burn option. The design is poised to provide the burn guidance required to support the mission challenges ahead.

REFERENCES

[1] Jeffrey Gutkowski, *EM-1 Distant Retrograde Orbit (DRO) Conceptual Flight Profile (CFP)*, EG-CEV-16-1, NASA Lyndon B. Johnson Space Center, Houston, TX, January 2016.

[2] John L. Goodman, "Powered Guidance Development for Apollo and the Space Shuttle," *Guidance, Navigation, and Control 2016, Advances in the Astronautical Sciences*, Vol. 157, Univelt, San Diego, CA, 2016, pp. 561–572.

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

[4] S. W. Thrasher and T. J. Fill, "Orion's Exo-atmospheric Burn Guidance Architecture and Algorithm," *AIAA GN&C Conference*, Portland, OR, July 13, 2011.

[5] S. K. Scarritt, T. J. Fill, and S. B. Robinson, "Advances in Orion's On-Orbit Guidance and Targeting System Architecture," Paper No. AAS 15-016, *AAS Guidance and Control Conference*, Breckenridge, CO, January-February 2015.

[6] T. J. Fill, *PEG Prediction Corrections*, Shuttle Memo 10E-80-11, The Charles Stark Draper Laboratory, Inc., Cambridge, MA, March 1980.

[7] Shane B. Robinson, Sara K. Scarritt, and John L. Goodman, "Encke-Beta Predictor for Orion Burn Targeting and Guidance," *Guidance, Navigation, and Control 2016, Advances in the Astronautical Sciences*, Vol. 157, Univelt, San Diego, CA, 2016, pp. 709–731.

[8] V. R. Bond and M. Allman, *Modern Astrodynamics*, Princeton University Press, New Jersey, 1996.

[9] S. W. Shepperd, *Linear Terminal Velocity Constraint Conic Required Velocity Routine Revisited*, Shuttle Memo 10E-77-24, The Charles Stark Draper Laboratory, Inc., Cambridge, MA, April 1977.

[10] Edgar C. Lineberry, *Invariant Orbital Elements For Use In The Description of Motion About an Oblate Earth*, JSC-09287, JSC Internal Note No. 74-FM-84, NASA/JSC Mission Planning and Analysis Division, December 4, 1974.

[11] Edgar C. Lineberry, *Orbital Transfers About an Oblate Earth*, JSC-09878, JSC Internal Note No. 75-FM-61, NASA/JSC Mission Planning and Analysis Division, August 12, 1975.

[12] R. Leroy McHenry, "Analytic Orbit Plane Targeting For Orbit Transfers About an Oblate Planet," *Proceedings of the 1991 AAS/AIAA Astrodynamics Conference*, Durango, CO, August 19-22, 1991, published in *Astrodynamics 1991, Advances in the Astronautical Sciences*, Vol. 76, Univelt, Inc., San Diego, CA, 1992, pp. 1305–1318.

[13] T. Fill, *Free-Range Linear Terminal Velocity Constraint (FR-LTVC) Maneuver Option*, CEV Flight Dynamics Team Technical Brief FltDyn-CEV-16-8, NASA Lyndon B. Johnson Space Center, February 2016.

[14] S. Robinson, *Some Geometric Relations and a Proposed Solution Process for the Free Range Linear Terminal Velocity Constraint Problem*, EG6 Technical Brief FltDyn-CEV-17-3, NASA Lyndon B. Johnson Space Center, January 2017.

[15] E. R. Lancaster, R. C. Blanchard, and R. A. Devaney, "A Note on Lambert's Theorem," *AIAA Journal of Spacecraft and Rockets*, Vol. 3, No. 9, September 1966, pp. 1436–1438.

[16] E. R. Lancaster and R. C. Blanchard, *A Unified Form of Lambert's Theorem*, NASA-TM-X-63355, Goddard Space Flight Center, September 1968.

[17] Richard H. Battin, "A New Solution for Lambert's Problem," *19th Congress of the International Astronautical Federation*, New York, NY, October 13-19, 1968.

[18] Richard H. Battin, "Lambert's Problem Revisited," *AIAA Journal*, Vol. 15, No. 5, May 1977, pp. 707–713.

[19] Richard H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, AIAA, Reston, VA, 1999, pp. 295–306.

[20] R. H. Gooding, "A Procedure for the Solution of Lambert's Orbital Boundary-Value Problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 48, 1990, pp. 145–165.

[21] Dario Izzo, "Revisiting Lambert's Problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 121, Issue 1, January 2015, pp. 1–15.

[22] Nitin Arora and Ryan P. Russell, "A Fast and Robust Multiple Revolution Lambert Algorithm Using a Cosine Transformation," Paper AAS 13-728, *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Hilton Head, SC, August 11-13, 2013, in *Astrodynamics 2013, Advances in the Astronautical Sciences*, Vol. 150, Univelt, Inc., San Diego, CA, 2014.

[23] John L. Goodman, *Lessons Learned From Seven Space Shuttle Missions*, NASA Contractor Report NASA/CR-2007-213697, NASA Johnson Space Center, January 2007.

[24] William M. Robertson, *Closed Form Solution of a Certain Common Conic De-orbit Problem*, 23A Space Guidance Analysis Memo No. 4-73, Charles Stark Draper Laboratory, Cambridge, MA, July 24, 1973.

[25] T. Fill, “Guidance Trade Study – Ascent SM Aborts,” Presentation, Charles Stark Draper Laboratory, Cambridge, MA, May 2009.

[26] W. Marscher, *A Unified Method of Generating Conic Sections*, MIT/Draper Laboratory Report R-479, Cambridge, MA, February 1965.