

ADVANCES IN ORION'S ON-ORBIT GUIDANCE AND TARGETING SYSTEM ARCHITECTURE

Sara K. Scarritt*, Thomas Fill†, and Shane Robinson‡

NASA's manned spaceflight programs have a rich history of advancing onboard guidance and targeting technology. In order to support future missions, the guidance and targeting architecture for the Orion Multi-Purpose Crew Vehicle must be able to operate in complete autonomy, without any support from the ground. Orion's guidance and targeting system must be sufficiently flexible to easily adapt to a wide array of undecided future missions, yet also not cause an undue computational burden on the flight computer. This presents a unique design challenge from the perspective of both algorithm development and system architecture construction. The present work shows how Orion's guidance and targeting system addresses these challenges. On the algorithm side, the system advances the state-of-the-art by: (1) steering burns with a simple closed-loop guidance strategy based on Shuttle heritage, and (2) planning maneuvers with a cutting-edge two-level targeting routine. These algorithms are then placed into an architecture designed to leverage the advantages of each and ensure that they function in concert with one another. The resulting system is characterized by modularity and simplicity. As such, it is adaptable to the on-orbit phases of any future mission that Orion may attempt.

INTRODUCTION

The exploration mandate of the Orion Multi-Purpose Crew Vehicle necessitates a level of autonomy and flexibility previously unparalleled in manned spaceflight. As future missions extend farther and farther from Earth, causing increasing lags or even losses in communication with the ground, Orion's on-board systems must be capable of operating the vehicle without the aid of Mission Control. Furthermore, it must be able to adapt to numerous different mission environments and objectives without tying up valuable flight computer resources. To address these demands, the Orion Guidance and Targeting system leverages decades of successful flight heritage from the Apollo and Shuttle programs, while at the same time incorporating cutting-edge algorithms and architecture design. The selected algorithms are highly flexible in nature and easily extensible to a wide variety of guidance and targeting problems. The system architecture is tailored to exploit the intrinsic advantages offered by each algorithm, particularly the way in which they interact with one another and with the rest of the GN&C system.

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

†Principal Member Technical Staff, Strategic & Space Guidance and Control Group, Charles Stark Draper Laboratory, 555 Technology Square, MS 70, Cambridge, MA 02139.

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

ON-ORBIT GUIDANCE AND TARGETING ARCHITECTURES IN US MANNED SPACE-FLIGHT

This section is meant to provide a brief history of onboard guidance and targeting for US manned spacecraft to provide context and show heritage of Orion's guidance and targeting architecture.

Apollo On-Orbit Guidance and Targeting Architecture

The Apollo on-orbit guidance and targeting architecture¹ relied heavily on ground support. The limited computational resources available meant that only very limited mission planning, or targeting activities, could be performed on-board the spacecraft. The on-board guidance routines has two modes: 1) External ΔV Guidance, 2) Lambert Aim Point Guidance. These two guidance modes were both based on cross product steering²⁻⁴ and differed only in the method for generating the required velocity vector. Lambert aim point guidance was capable of performing rendezvous maneuvers, or executing a return to earth maneuver in emergency situations. Outside of rendezvous, During nominal on operations each burn was computed by mission control, and the $\overrightarrow{\Delta V}$ vector was transmitted to the craft for execution using the external ΔV mode.

Space Shuttle On-Orbit Guidance and Targeting Architecture

The Space Shuttle flight regime was divided into three operational sequences (OPS). OPS-1 covered ascent and orbit insertion, OPS-2 covered on-orbit operations, and OPS-3 covered deorbit, entry and landing. In all of the orbital major modes within each OPS, the same basic construct was employed. For insertion and deorbit phases, there were two basic transfer modes – linear terminal velocity constraint (LTVC) and external ΔV . The LTVC transfer targets an intercept position while achieving an intercept velocity vector which possesses horizontal and vertical components that satisfy a predetermined linear constraint. The external ΔV nominally requires a constant inertial direction of thrust until a desired ΔV vector is achieved. The LTVC mode was the primary transfer mode for insertion and deorbit. Both OPS phases utilized essentially the same executive guidance architecture involving three key elements: a Maneuver Display Processing task for responding to crew requests, scheduling the pre-burn computations and transitioning to active guidance; a pre-maneuver display support task that essentially provided pre- and post-processing of the pre-burn maneuver computations; and the Powered Explicit Guidance (PEG) task, which is a predictor-corrector algorithm utilized to account for finite burn effects in the execution of the either transfer mode, and in the process, solved for the guidance outputs of steering profile and engine cutoff time to achieve the desired transfer targets. Burn guidance was terminated by transition to the next major mode.

The OPS-2 guidance framework and algorithms were identical to the other two OPS phases except the primary transfer mode was external ΔV . A rendezvous maneuver mode, “Lambert Targeting”, was also available, where essentially the external ΔV was continuously updated during the burn to hit a targeted position at some specified time. Lambert maneuvers were initially computed using an separate targeting algorithm known as the “Orbit Targeting Specialist Function”, providing the initial external ΔV for a rendezvous burn. During the burn the remaining external ΔV to be achieved was updated by calling a basic Lambert Conic-Velocity-Required” routine with a simple “initial position offset” biasing scheme to account for finite burn effects. The mechanization of the on-orbit algorithm was sufficiently simple, and the burns of relatively short duration, that no PEG task was allocated towards its implementation.

All of the orbit guidance with each OPS phase was executed at 1 Hz to provide closed-loop updates of the guidance steering command and engine cutoff time outputs. Much of this successful guidance heritage has been carried over into Orion’s guidance architecture. Numerous references are available for greater detail on the Shuttle guidance development and implementation.⁵⁻⁷

ORION’S GUIDANCE AND TARGETING SYSTEM

The On-orbit Guidance and Targeting (GDO) domain provides guidance and targeting solutions for all powered burns during the on-orbit phase of the mission, including Trans-Earth Injection (TEI), Lunar Destination Orbit (LDO) operations, Lunar Transit Abort, and Deorbit. Within the GDO domain, Configuration Software Units (CSUs) are used to perform specific functions within a mission phase. A CSU houses the algorithms necessary to complete a specific activity within a mission phase and responds to one or more mode commands that configure the CSU to run one or multiple algorithms housed within the CSU. Because Orion is designed to support a variety of missions, it would be highly impractical to include separate CSUs for every possible type of burn that Orion might perform. Instead, the Orion Guidance and Targeting architecture integrates all onboard guidance and targeting functionality into two flexible CSUs: the Two-Level Targeter (TLT) and Orbit Guidance (OrbGuid). Figure 1 is a high-level representation of this integrated architecture. These CSUs are generic in nature and have the capability to handle a wide range of burn scenarios. The interface between them is determined by the current GDO internal mode command, and outputs to the flight control system are configured by the GDO Junction Output Box (JOB).

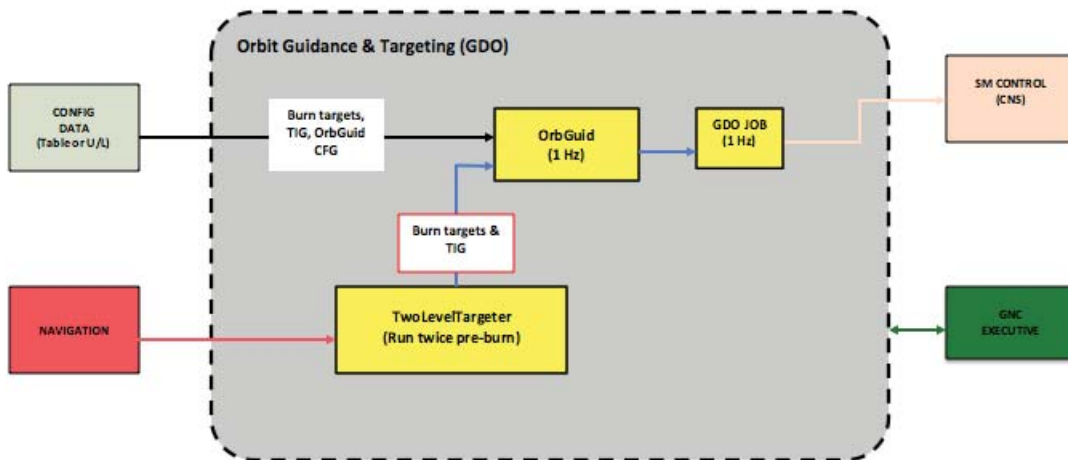


Figure 1. Orion Guidance & Targeting Architecture

The Orion Guidance and Targeting system represents a shift from the traditional paradigm of guidance and targeting functions. The role of “targeting,” in the context of the Orion architecture, encompasses aspects of both traditional single-burn targeting and of path planning. The Orion on-orbit targeting system incorporates both the current vehicle state information and a full set of future

burn opportunities and trajectory constraints; it then targets all of the future burns simultaneously, essentially adjusting the entire upcoming trajectory in order to meet all of the active constraints. The TLT will be run prior to each scheduled burn so that the resulting targeted trajectory includes the most recent navigation data. The targeting system will configure the converged trajectory output into appropriate burn targets to be passed to OrbGuid. Responsibility for directing each individual burn to achieve these burn targets falls under the purview of “guidance.” The OrbGuid CSU will immediately prior to ignition to determine the velocity that must be achieved by the burn in order to meet the burn targets - a function traditionally described as targeting - and then run cyclically during the burn to compute steering commands and cutoff time based on feedback from the navigation system.

The TLT and OrbGuid CSUs are the backbone of the Orion GDO architecture. The next sections provide an overview of the CSUs themselves, including descriptions of the logic flow and a brief discussion of the underlying algorithms. These sections highlight the intrinsic advantages of each algorithm and show how the Guidance and Targeting system leverages those advantages.

Two-Level Targeter Overview

The TLT CSU handles all on-orbit translation burn targeting, from final stage separation to entry interface. It is comprised of three main elements: the initialization script, the two-level targeting algorithm itself, and a post-processing script. A top-level flow diagram of the CSU is shown in Figure 2. The initialization script takes the input and parameter buses coming into the CSU and parses the data into the TLT internal state structure. The initialization also determines the number of maneuvers, maneuver locations, and the number of active constraints, and passes that to the main algorithm along with the internal states.

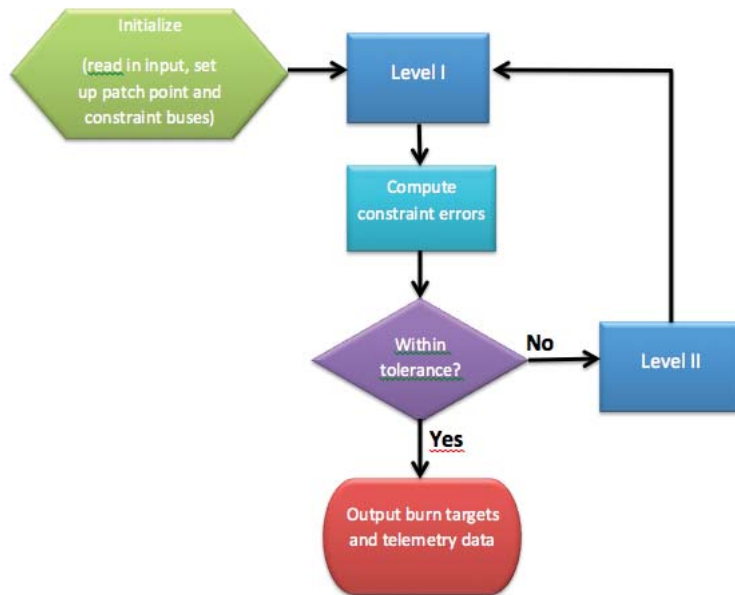


Figure 2. TLT logic flowchart

The two-level targeting algorithm simultaneously targets an arbitrary number burns with the ob-

jective of meeting a set of trajectory constraints to within a specified tolerance or tolerances. The algorithm is primarily based on linear systems theory; it uses a time-varying linearized dynamical model and a minimum norm solution to compute solution updates. These linear updates are implemented in the nonlinear system in an iterative corrections process that repeats until a feasible solution is identified. The algorithm is independent of the vehicle dynamics, requiring only that they follow the basic form $\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X})$. Thus, it can be applied to multiple different gravitational regimes. The constraint formulation is similarly adaptable, allowing the selection of available trajectory constraint types to be easily expanded.

The targeter requires a startup arc represented by a series of N “patch states.” These states, also termed “patch points,” consist of a position, velocity, time, and associated burn and/or constraint parameters for that state. They are selected by the user as representative waypoints along the trajectory. The algorithm consists of two main steps, the Level I process and Level II process, which adjust either the velocity or the position and time, respectively, of each patch state in order to satisfy the trajectory constraints. A single iteration of the targeter consists of first cycling through the Level I process (which is itself iterative) in order to ensure that the current trajectory solution is continuous in position, then correcting for any constraint values that lie outside tolerance via the Level II process. The patch state adjustments are made via a differential corrections process based on the linearized dynamical model. Certain patch states within the arc are designated as burn states; the converged position, velocity, and time at a particular burn state, and at the patch state immediately following it, provides the final targeting solution for that burn. An earlier development of the Two-Level Targeter (TLT) was employed during the design of the Genesis trajectory. That derivation, which is the basis for the current design, is well documented,⁸⁻¹⁰ as are the subsequent modifications to the algorithm to adapt it to the onboard targeting process.¹¹⁻¹³

The TLT CSU output is the burn targets that are required to achieve the desired trajectory as determined by the targeting algorithm. The various types of burn targets will be discussed in an upcoming section. The post-processing script extracts all the output burn target data and telemetry data from the converged patch state set and configures it into the proper output structure. If the impulsive version of the targeter is running, then this script also computes a time-of-ignition (TIG) bias correction value to account for finite burn effects in the burn execution. This computation, which is based on Shuttle heritage, seeks to center the impulsive TIG at the centroid of the expected burn arc by shifting the targeted TIG earlier. If the targeter is executing in finite burn mode, modeling the full thruster dynamics within the linearized system, the output TIG is the actual desired engine on-time and this computation is not necessary.

Orbit Guidance Framework Overview

Orion’s burn guidance is divided into an ascent and post-ascent phases. The Orbit Guidance CSU provides burn guidance for all of the post-ascent orbit phases. Its 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 desired target conditions, including the desired TIG, are provided by either ground uplink or by onboard targeting systems such as the Two-Level Targeter. Parameter data on the planned thruster performance and vehicle mass are also part of the target set.

The OrbGuid algorithm structure derives its heritage from the Space Shuttle insertion/deorbit powered flight guidance. Efforts to unify the various Shuttle ascent, insertion, on-orbit and deorbit powered guidance phases around a core predictor-corrector algorithm, named 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 unify an even broader range of orbital guidance maneuvers than did Shuttle. Through a menu of various desired velocity routines within the corrector framework, Orion can apply the same guidance algorithm across to all of its orbital powered burns such as orbit insertion, rendezvous, deorbit, CM raise burns, externally specified ΔV burns, lunar transfer and earth-return burns. OrbGuid's desired velocity routines also include enhancements over Shuttle such as a broader set of burn maneuver types, explicitly accounting for higher-order gravity perturbations over the maneuver to the target, and closed-loop updates of the burn residuals during the post-burn trim.

OrbGuid consists of two main parts, a top-level internal executive wrapped around PEG and the PEG algorithm itself. Figure 3 shows the flow diagram for the internal top-level logic. OrbGuid handles initialization and reinitialization 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 input time and 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 (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, setting the maximum number of PEG iterations to a value sufficient to allow PEG to run to full convergence, and setting the solution tolerance to a tight enough value for a precise solution. The solution tolerance specifies the acceptable miss between the predicted and desired velocity states at the end of the burn. The PEG algorithm is then called to converge on the burn solution which also yields the steering profile and the desired engine shutdown time. PEG is a semi-analytic algorithm where the form of the steering law allows for certain analytic approximations that enable solution 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,¹⁴ and a numeric integration of a neighboring coasting trajectory¹⁵ to predict the effects of gravity over the burn. A correction process is then employed to null the miss between the predicted and desired velocity states at burn cut-off using the velocity-to-be-gained vector, v_{go} , as the iteration variable. Over a small number of iterations, the algorithm converges on the 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. The details of the PEG solution are available in the literature.¹⁶

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 runs up to a maximum of two iterations per guidance cycle to update the steering solution for the latest vehicle state and maintain convergence to within a specified tolerance.

An output processor computes predicted maneuver characteristics for crew displays, including predicted apogee and perigee at burn cut-off, time-to-go to burn completion, time interval from burn cut-off to target intercept, and v_{go} in body-fixed coordinates.

At a pre-determined time interval before desired engine cut-off, OrbGuid holds the steering command direction to avoid rapid slews as the v_{go} vector trends to a zero magnitude. The control system

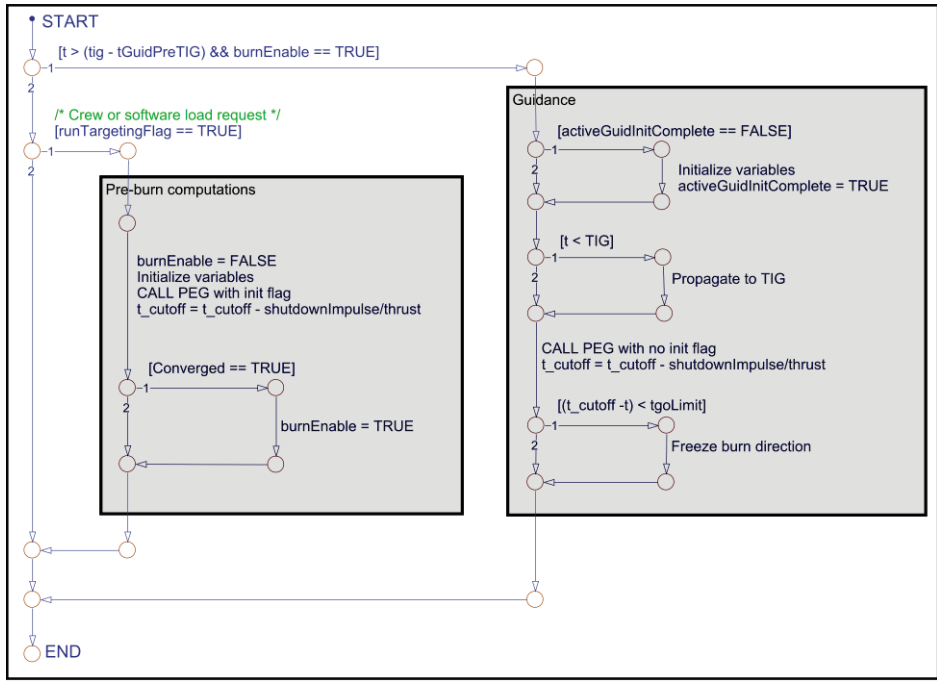


Figure 3. OrbGuid logic

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 the residual v_{go} required to achieve the burn objectives.

If the residual v_{go} from OrbGuid exceeds a threshold pre-determined for the specific burn, then the vehicle enters an automatic trim burn activity, subject to crew authorization. OrbGuid passes the v_{go} solution to the control logic once per guidance cycle until the desired burn accuracy is achieved. Then the vehicle enters an attitude hold while the crew monitors the OrbGuid v_{go} output. The crew has the option to further clean up the burn using the translation hand controller to execute RCS pulses. The crew can opt to bypass the automatic trim and perform a manual trim instead in the same manner. Only after the residual v_{go} satisfies the crew and the crew proceeds to the next activity will the vehicle cease to execute OrbGuid.

OrbGuid’s versatility comes mostly through the different desired velocity routines in the PEG corrector – each being related to a specific maneuver mode. The extensible nature of the PEG framework provides the ability to readily add new maneuver types as the need arises. The current menu of maneuver modes are detailed in the following 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 maneuver type (or mode) parameter controls which routine to execute, and the target set contains the parameters necessary to evaluate a solution. Another parameter, the planar guidance switch, controls how OrbGuid handles the out-of-plane burn component. The

separation of in-plane and out-of-plane components is essential near 180° transfer singularities, but it also enables OrbGuid to protect against fuel and coasting time limits or to target the plane of the landing site in the case of a deorbit burn. What follows is a brief description of the maneuver modes currently implemented within OrbGuid.

External Delta Velocity Mode

In many cases, an external source, such as a separate targeting routine or uplink from mission control, provides a desired velocity change to be achieved by thrust, the ext- ΔV vector, which is then executed by OrbGuid. External ΔV guidance begins with v_{go} being initialized to the ext- ΔV vector from the input target set, so no correction is necessary. To stay within the PEG predictor-corrector framework, the assignment of $v_d = v_p$ is made as the desired velocity routine, resulting in convergence on the first iteration. Fitting the external ΔV into PEG this way reduces the need for an additional guidance CSU. It also makes the turning rate feature available. Options are provided to specify the ext- ΔV vector in inertial coordinates or in a local-vertical local-horizontal (LVLH) coordinate frame defined by the vehicle ignition state.

Linear Terminal Velocity Constraint (LTVC) Mode

The LTVC problem is concerned with finding a velocity to intercept a downrange position vector target while constraining the radial and horizontal velocity components by a linear relationship at the target. 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 linear relationship between radial and horizontal velocities at the target is given by

$$\dot{r} = C_1 + C_2 v_h$$

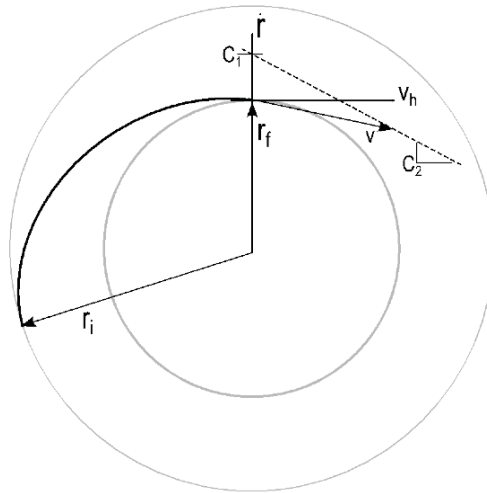


Figure 4. LTVC solves for the velocity to take a vehicle from r_i to r_f and meet the velocity constraint at r_f .

LTVC works well for the orbit insertion problem, which targets a desired apsis altitude from an apsis, and the velocity at the opposite apsis should be horizontal. This corresponds to $C_1 = C_2 = 0$. It also works well for a deorbit burn, where the target is defined at the entry interface (EI) altitude, and the trajectory chosen with a particular entry interface flight-path angle γ . For a fixed flight-path

angle, $C_1 = 0$ and $C_2 = \arctan\gamma$. In practice, the two constraints are chosen to control velocity dispersions at entry interface.

Bond and Allman give a historical background of LTVC development and provide a straightforward derivation of the conic formulation.^{17,18} OrbGuid has a high-order propagated option built on top of the precision LTVC formulation. Precision LTVC refers to LTVC that accounts for the J_2 gravity perturbation. Lineberry derived an analytic solution for in-plane precision LTVC.^{19,20} McHenry derived a formulation for both in-plane and out-of-plane J_2 perturbations.²¹ Both solutions result in a minimal increase in algorithmic complexity over the conic formulation. The precision solution degrades to the conic formulation for the J_2 coefficient set to zero, so the precision formulation is implemented exclusively.

High-order LTVC: OrbGuid's high-order LTVC option takes advantage of increased computing power to bias targets on-board. This biasing is accomplished using a single propagation of the precision LTVC solution per each call to a high-order (HOG) wrapper routine. The solution is propagated to the plane that contains the target vector and is perpendicular to the orbit plane. The target vector is then biased internally by subtracting the miss vector, and the C_1 parameter is biased by subtracting the radial velocity miss. Figure 5 illustrates the biased targets. The HOG routine stores these two biased quantities as internal states. After several calls to the routine during PEG convergence, precision LTVC with biased targets rapidly converges to the high-order propagated solution. Once again, the HOG solution degrades to the conic formulation if the gravity model is set to the central-body inverse-squared gravity field. So the HOG propagation, combined with the precision LTVC solutions, are utilized exclusively.

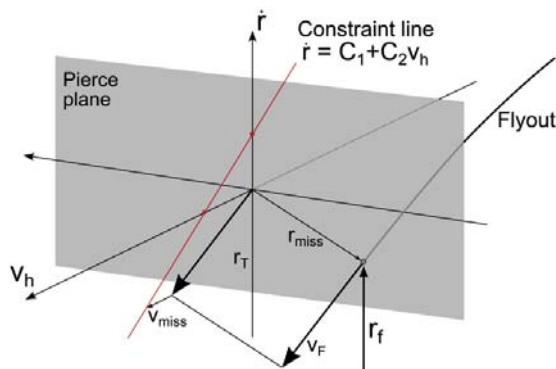


Figure 5. The target is biased in the opposite direction of the miss

Out-of-plane Solution: OrbGuid uses several methods to approach the out-of-plane component to the LTVC solution: in-plane only, target intercept, velocity null, and planet-fixed target plane intercept. Solving for the in-plane solution only 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-orbital-plane position miss found by the propagated solution.

$$v_{i,miss,y} = \frac{v_{i,horz} r_{f,miss,y}}{r_0 \sin\theta}$$

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 position miss. Finally, the planet-fixed target plane intercept option is useful for bringing the orbit plane over a planetary target such as a deorbit landing site. This method requires an estimate of transit time between the LTVC downrange target and the planar intercept target. Using this parameter, the planet-fixed target vector can be converted into inertial coordinates and the LTVC downrange target moved into the appropriate orbit plane defined by the predicted cutoff position and the inertial planet target vectors.

In addition to directly specified out-of-plane intercepts, OrbGuid LTVC has the option to protect against minimum mass or minimum post-burn-coast-time-to-target-intercept constraints. Minimum mass includes fuel, and the mass protection prevents Orion from exceeding fuel reserves in order to meet out-of-plane targets. Similarly, the minimum coast time between cutoff and target intercept is useful for earth deorbit where a certain amount of free-fall time is required to accomplish docking mechanism jettison, SM jettison, CM orientation, and CM burn (if required). In the case of a 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.

Transit (or Lambert) Desired Velocity Mode

The well-known Lambert boundary value problem involves finding the velocity required to transfer a vehicle between two position vectors with a specified transit time in a central body gravity field. OrbGuid uses Gooding's solution.²² This mode provides one of two direct mappings of the TLT outputs directly into OrbGuid for guided execution. The TLT burn solution provides an ignition time and state as well as the desired end state in the form of the next patch point state. If time is the primary transfer objective, the time of the first patch state, and both the second patch state time of arrival and inertial position provide the TIG and targets for this mode.

Similar to the High-order LTVC mode, a HOG Transit option is available to bias the Lambert targets on-board. This biasing is accomplished using a single propagation of the conic solution per each call to the same HOG routine as used for the LTVC biasing. The solution is propagated to the plane that contains the target vector and is perpendicular to the orbit plane. The target vector is then biased internally by subtracting the miss vector, and the transit-time parameter is biased by subtracting the transit-time miss. Once again, Figure 4 illustrates the biased position target. The high-order HOG routine stores these two biased quantities as internal states. After several calls to the routine during PEG convergence, conic Lambert with biased targets rapidly converges to the high-order propagated solution. The common use of the HOG routine provides the option to use the same methods to approach the out-of-plane aspects of the Lambert solution: in-plane only, target intercept and velocity null.

Patched Mode

The problem of orbital transfer between two inertial position vectors with a common focus is explored thoroughly in Battin.²³ The solution to this problem is a direct output of the TLT in the form of the position and velocity states of two successive patch points. The OrbGuid implementation of a patched mode provides a second direct mapping of the TLT outputs directly into OrbGuid for guided execution. When velocity at the second patch point is the primary objective, the time of the first patch state, and the position and velocity at the second patch point provide the TIG and targets for this mode.

Rather than resorting to a new conic solution for this problem (such as can be had by noting the

components of the terminal velocity states along skewed radial and chordal axes are the same), OrbGuid leverages the LTVC formulation. In this formulation, the target terminal velocity is converted to an equivalent linear terminal velocity constraint by way of

$$C_1 = 0$$

$$C_2 = \frac{v_T \cdot \hat{i}_x}{v_T \cdot \hat{i}_z}$$

where \hat{i}_x and \hat{i}_z are the local vertical and horizontal directions, respectively, at the target position. This implementation will yield the same theoretical solution under ideal conditions while benefiting from the target biasing for higher-order gravity perturbations. Additionally, this approach serves to avoid maneuver instability issues by not over constraining the guidance problem that would otherwise exist if a specific velocity target value was imposed at target intercept. This mechanization also provides the flexibility to use the same methods to approach the out-of-plane aspects of the transfer: in-plane only, target intercept and velocity null. Again, the HOG LTVC option is available to bias the target position and C_1 on board.

Constrained Intermediate Terminal Intercept (CITI) Mode

The CITI algorithm was derived by Robertson,²⁴ although the name was only recently applied.²⁵ This algorithm has application to intercept problems requiring a constraint on the flight path 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 6. Several maneuver scenarios have been identified for application of the the CITI mode desired velocity routine, including a number of lunar orbit maneuvers.

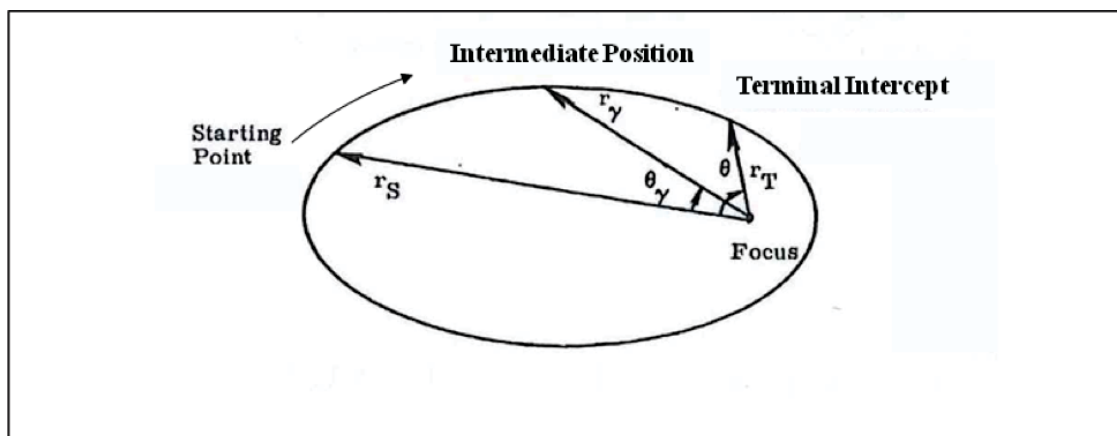


Figure 6. CITI orbital geometry

Similar to the LTVC problem, the target intercept position can be specified as an inertial vector, or as a combination of altitude at the target and transfer angle from TIG. The target intercept position vector can also be specified by a planetary-fixed (surface) position vector. The intermediate position magnitude can be specified as either an altitude relative to an equatorial radius, ellipsoidal (latitude dependent) radius, or as a radius magnitude. Similar to the previous intercept mode, a HOG transit option is available to bias the intermediate radius magnitude, the intermediate flight-path angle, and the target intercept vector. This biasing is accomplished using a single propagation of the

conic solution per each call to the HOG wrapper routine. The solution is propagated first to the intermediate position vector, and then on to the plane that contains the target intercept vector and is perpendicular to the orbit plane. The respective quantities are then biased by subtracting their respective miss with the desired values. The high-order HOG wrapper routine stores these three biased quantities as internal states. After several calls to the routine during PEG convergence, the conic CITI routine with biased targets rapidly converges to the high-order propagated solution. The common use of the HOG routine provides the option to use the same methods to approach the out-of-plane aspects of the CITI solution: in-plane only, target intercept and velocity null.

CONCLUSIONS

To support NASA's future manned exploration missions, the Orion Multi-Purpose Crew Vehicle requires an unprecedented level of autonomy and adaptability. Orion's onboard Guidance and Targeting system utilizes a combination of flexible algorithms and novel architecture design to meet this demand. The design leverages flight heritage from previous manned and unmanned programs while simultaneously advancing the state-of-the-art in preparation for the challenges ahead.

REFERENCES

- [1] G. M. Levine, "Guidance System Operations Plan for Manned CM Earth Orbital and Lunar Missions using Program Colossus 3: Section 5 - Guidance Equations (Rev. 14)," Technical Report R-577, MIT Charles Stark Draper Laboratory, March 1971.
- [2] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA Education Series, Reston, Virginia: American Institute of Aeronautics and Astronautics, revised ed., 1999.
- [3] F. H. Martin and R. H. Battin, "Computer-Controlled Steering of the Apollo Spacecraft," *Journal of Spacecraft and Rockets*, Vol. 5, April 1968, pp. 400–407, 10.2514/3.29267.
- [4] W. E. Moore, "Lambert's Problem and Cross-Product Steering," Technical Report No. 67-8, Analytical Mechanics Associates, 1967.
- [5] D. Zimpfer and P. Hattis, "Space Shuttle GN&C Development History and Evolution," AIAA, 2010.
- [6] "Space Shuttle Orbiter Operational Level C Functional Subsystem Software Requirements Part A Guidance Ascent/RTLs," Tech. Rep. STS 83-0002D OI-21, Rockwell International, January 1991.
- [7] "Space Shuttle Orbiter Operational Level C Functional Subsystem Software Requirements Part A Guidance On-Orbit/Deorbit," Tech. Rep. STS 83-0003-26B, Rockwell International, August 1996.
- [8] R. W. K.C. Howell, B.T. Barden and M. Lo, "Trajectory design using a dynamical systems approach with application to GENESIS," *Proceedings of the AAS/AIAA Astrodynamics Conference*, Sun Valley, ID, 1997, pp. 1665–1684.
- [9] K. H. R.S. Wilson, B.T. Barden and B. Marchand, "Summer Launch Options for the Genesis Mission," *Advances in the Astronautical Sciences*, Vol. 109, 2002, pp. 77–94.
- [10] R. Wilson and K. Howell, "Trajectory Design in the Sun-Earth-Moon System Using Lunar Gravity Assists," *Journal of Spacecraft and Rockets*, Vol. 35, No. 2, 1998, pp. 191–198.
- [11] K. H. B.G. Marchand and R. Wilson, "Improved Corrections Process for Constrained Trajectory Design in the n-Body Problem," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007, pp. 884–897.
- [12] C. S. M.W. Weeks, B.G. Marchand and S. Scarritt, "Onboard Autonomous Targeting for the Trans-Earth Phase of Orion," *Journal of Guidance, Control, and Dynamics*, Vol. 33, May 2010, pp. 943–956.
- [13] A. B. W. T. S.K. Scarritt, B.G. Marchand and M. Weeks, "Finite Burn Linear Targeting Algorithm for Autonomous Path Planning and Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 35, September 2012, pp. 1605–1615.
- [14] T. J. Fill, "PEG Prediction Corrections," Shuttle Memo 10E-80-11, The Charles Stark Draper Laboratory, Inc., Cambridge, MA, March 1980.
- [15] B. T. L. A. C. B. T. J. McHenry, R.L., "Space Shuttle Ascent Guidance, Navigation, and Control," *The Journal of the Astronautical Sciences*, Vol. 27, January-March 1979.
- [16] S. Thrasher and T. Fill, "Orion's Exoatmospheric Burn Guidance Architecture and Algorithm," *AIAA GN&C Conference*, Portland, OR, July 2011.
- [17] V. Bond and M. Allman, *Modern Astrodynamics*. Princeton, NJ: Princeton University Press, 1996.

- [18] S. Shepperd, "Linear terminal velocity constraint conic required velocity routine revisited," Shuttle Memo 10E-77-24, The Charles Stark Draper Laboratory, Cambridge, MA, April 1977.
- [19] E. C. Lineberry, "Invariant Orbital Elements For Use In The Description of Motion About an Oblate Earth," Tech. Rep. JSC-09287, NASA/JSC Mission Planning and Analysis Division, December 1974.
- [20] E. C. Lineberry, "Orbital Transfers About an Oblate Earth," Tech. Rep. JSC-09878, NASA/JSC Mission Planning and Analysis Division, August 1975.
- [21] R. L. McHenry, "Analytic Orbit Plane Targeting For Orbit Transfers About an Oblate Planet," *Advances in the Astronautical Sciences*, Vol. 76, 1992, pp. 1305–1318.
- [22] R. Gooding, "A Procedure for the Solution of Lambert's Orbital Boundary-Value Problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 48, January 1990, pp. 145–165.
- [23] R. Battin, *Astronautical Guidance*, ch. Chapter 4. McGraw-Hill, 1964.
- [24] W. M. Robertson, "Closed Form Solution of a Certain Common Conic De-orbit Problem," Analysis Memo 4-73, 23A Space Guidance, July 1973.
- [25] T. Fill, "Guidance Trade Study - Ascent SM Aborts," Presentation, Charles Stark Draper Laboratory, May 2009.