AAS 21-561

# REVISITING TRAJECTORY DESIGN WITH STK ASTROGATOR PART 2

Cody Short[*], Linda Kay-Bunnell[†], Doug Cather[‡] and Nathaniel Kinzly[§]

Humankind's surrogate presence in space continues to increase year-over-year. Indeed, space missions continue to generally grow in scale and complexity as increasingly more vehicles are placed in various regimes. Large constellations of small spacecraft as well as missions in lunar and cislunar space have expanded dramatically in recent years. For all of these efforts, effective software tools to support design, analysis and operations are critical. This paper is intended to serve as the second installment in a continuing series concerning the Systems Tool Kit (STK) Astrogator capability set from AGI, an Ansys Company.

## INTRODUCTION

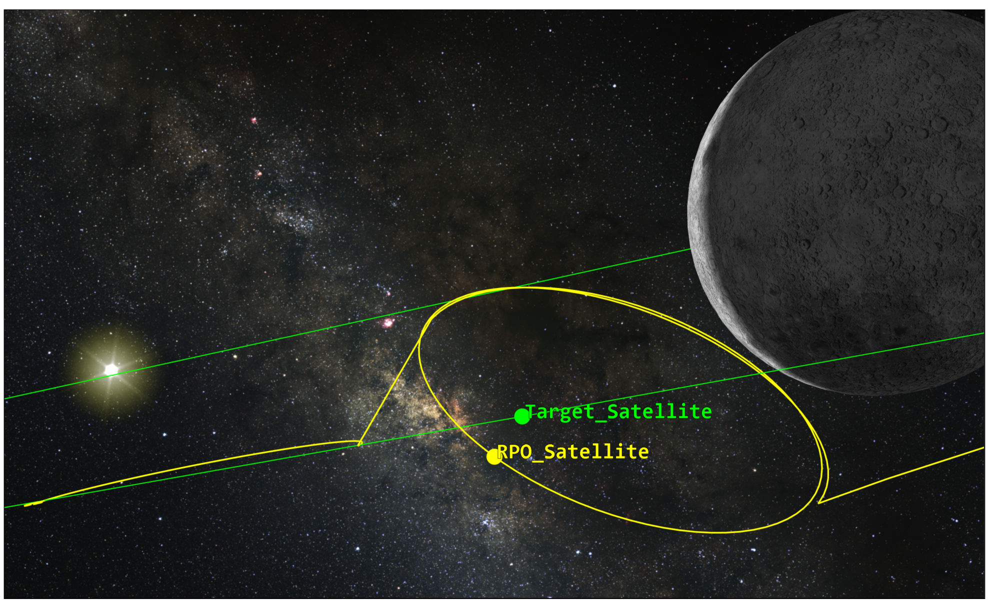


**Figure 1**: A sequence of proximity operations in lunar orbit produced in Astrogator

[*]STK Astrogator Product Owner and Technical Lead, AGI, an Ansys Company, Exton, PA 19341
[†]Senior Test Engineer, AGI, an Ansys Company, 220 Valley Creek Blvd., Exton, PA 19341
[‡]Space Operations Technical Director, AGI, an Ansys Company, Colorado Springs, CO 80920
[§]Astrodynamics Software Engineer, AGI, an Ansys Company, 220 Valley Creek Blvd., Exton, PA 19341

This paper will cover recent additions to the STK Astrogator toolset, such as the now out-of-the-box availability of rendezvous and proximity operations (RPO) templates. An example of a mission composed using these RPO strategies is depicted in Fig. 1 above, where an *inspector/chase vehicle* (yellow) is depicted performing operations in close proximity to a *target vehicle/resident space object* (RSO, green) which is itself in orbit about the Moon. These RPO capabilities, which represent decades of mission operations experience based upon a preponderance of proven theory and strategy,[1,2] were previously available in a limited beta for users upon request. These pre-packaged strategies allow operators to jump into RPO activities with dramatically reduced spin-up time—making these capabilities available to all Astrogator users represents a tremendous step toward increasing the immediate effectiveness of both the toolset and its users.

The paper also covers various other additions aside from the RPO strategies, and gives some insight into the Astrogator development process. A brief summary of expanded state transition matrix (STM) options, part of an ongoing effort to increase dynamical systems capabilities within Astrogator, is offered. This new STM-related functionality receives expanded treatment in a companion paper by Kinzly et al.[3] High level orchestration approaches enhanced by closer integration of Python tools including embedded Jupyter notebooks within STK, improved ephemeris handling and expanded support for user data in plugins are also covered. In addition to these discussions of "what's new", this paper also serves to revisit a few points from the previous installment in the series, covering minor corrections and updates. Additionally, a high-level overview of the Astrogator development approach with particular focus on quality assurance (QA) processes is explored. A specific anecdotal observation of this workflow demonstrates the effectiveness of using QA efforts to drive the development process, a reflection of a common strategy known as test-driven software engineering that is being incremetally adopted where possible.

The document is organized as follows. In the first section, STK Astrogator, a brief introduction to the software is given; this section reproduces similar background material from the initial installment of the paper series[4] as well as an intermediate paper.[5] Next, a brief section, Remarks on the Previous Installment, revisits two points from Part 1 and supplies an updated table of Astrogator missions. Then, a discussion and enumeration of the Rendezvous and Proximity Operations (RPO) strategies that have been incorporated into Astrogator is given. A brief introduction to the CR3BP as well as few notes regarding its implementation in Astrogator are made in the subsequent section, Circular Restricted Three-body Problem; this section follows from the previous reference[5] as well. Then, an introduction to the State Transition Matrix (STM), which has recently received additional attention for use in Astrogator is made. A section describing Additional Advancements relevant to Astrogator follows. The penultimate section, Test-driven Development and Quality Assurance Processes, discusses the quality assurance process for vetting Astrogator and, more generally, STK. This section also provides an example of a test-driven strategy that has piloted well in Astrogator development. Finally, the paper is concluded with Concluding Remarks and Future Work along with a brief Acknowledgments section and References.

## STK ASTROGATOR

Astrogator's roots lie in a strong lineage of tools with incarnations dating back to 1989.[6*] Many elements and algorithms preceded these formal offerings, and others have arisen over time. The tool has been used for analysis, design and operations on missions ranging from LEO to GEO,[6] from the Sun[7] to Arrokoth,[8,9] and many places in between. Most of these applications faced unique requirements, and software enhancements resulted in response.[4,5,10–12] With particular respect to operational capacities, Astrogator has been employed to support numerous programs. The earliest mission utilizing the software was the Wilkinson Microwave Anisotropy Probe (WMAP) mission.[13] Many current missions in various phases of the mission design lifecycle utilize Astrogator in some capacity. Several specific missions for which Astrogator has been used, most for operations, are tabulated in Table 1 found in the next section where they have been categorized by mission regime.

The role of Astrogator in the spacecraft mission design lifecycle is largely reflected in the trajectory design and analysis phases, although it is also relevant to other stages such as early concept design and selection as well as operations. Astrogator's integration within STK leads to significant synergies that enable trade studies where the environment is modeled accurately to capture many important contributions to the system. For example, inherent system awareness of periods of time when a spacecraft has line-of-sight intervisibility (or access) to other resources will affect other aspects of the mission design. Considerations such as the access example are also critical during operational phases of a spacecraft mission. Further, the integration with STK allows for feedback through graphics and data product reporting.

While the STK system lends some support, Astrogator displays several independent strengths. In turn, Astrogator extends the capabilities of the STK system. Particular strengths of Astrogator are reflected in the principles of modularity and configurability. These principles are implemented in the STK Component Browser and associated component technology.[14] The various components in the browser represent individual aspects of the mission model. A particular component may reflect part of the force model or characteristics associated with a central body for the system. Some components represent calculations and provide mechanisms for reporting various data, while others interact together to produce a functional unit or even a sequence of segments to define an Astrogator satellite. These capabilities offer extensive "out-of-the-box" options, while also providing plugin points for users to supply their own models when needed. Additional mechanisms exist to execute the simulation. The Mission Control Sequence (MCS) is an interactive environment for designing and configuring the evolution of spacecraft trajectories. The MCS uses various segments/objects to capture the design—each of these pieces is a *component*.

Often, the interactive desktop environment is employed to construct and configure a particular solution strategy. Indeed, this rapid design architecture accompanied by graphical feedback enables users to quickly arrive at a particular solution and then refine that solution. It is sometimes the case that the design begins with external analysis and a particular

[*] Much of the text in this section follows Short, Haapala and Bosanac[5] and Short, Ghosh and Claybrook[4]

trajectory is migrated into Astrogator for refinement, but it is also possible for the design to progress through levels of model fidelity entirely within STK/Astrogator. The interactive environment is well-poised to support both types of operations. While the MCS provides an interactive environment, both internal and external scripting capabilities allow for even greater configurability and automation. Regular tasks that require re-execution of a previously computed Astrogator trajectory under updated conditions are typically automated through a combination of internal scripting and external API wrapping. Astrogator can be operated by scripting in MATLAB, through compiled Microsoft COM programs or native and extended Windows/Linux scripting. Thus, the high-fidelity solutions previously computed in support of some operational goal can be programmatically adjusted and recomputed as required. Additionally, large-scale research analyses are supported through the same concepts. This option between operating modes allows for a large breadth of applicability, and some combination of Astrogator's capabilities can be employed to address most astrodynamical problems. Before taking a closer look at some of the specific aspects of the tool in terms of their functionality and technical implementation a few remarks on Part 1 of this paper series follow.

## REMARKS ON THE PREVIOUS INSTALLMENT

Two aspects of the previous installment in this paper series require revisiting. First the discussion in that paper of the Astrogator differential corrector conflated the concept of floating end-point targeting associated with the Astrogator predecessor, Swingby,[15] with what is implemented in the differential corrector. The differential corrector solves for a set of equality constraints and consequently cannot be described as a *floating end-point* solver. Such approaches, however, may be accomplished with the various optimization search options in Astrogator. Next, the table of missions provided in Part 1 of the paper series has been updated to include feedback and additional information. This updated table is included here as Table 1[*].

## RENDEZVOUS AND PROXIMITY OPERATIONS

Situations that involve relative motion between more than one spacecraft, once more isolated occurrences, are becoming more and more common. Scenarios involving rendezvous operations such as servicing of assets or transferring crew and cargo happen on a relatively regular basis. Operations bringing vehicles into close proximity with one another without rendezvous are also common. Such proximity operations may occur in either a cooperative or non-cooperative sense. Various spacecraft have been tracked performing proximity operations near other satellites on orbit, those flying under a common flag as well as not.[16,17] While the motivation behind such operations is often not disclosed, it is generally accepted as important for the associated behavior to be understood with capabilities available to quickly model and analyze the corresponding motion. In response to this growing need,

[*]Table 1 is compiled from input of various AGI personnel and in cooperation with AGI partner Space Exploration Engineering, LLC principals Mike Loucks, John Carrico and Lisa Policastri. The authors welcome feedback with respect to errors and omissions. Corrections and/or additions will be reflected in later installments of this paper series.

| LEO | | GEO/GTO, HEO | Lunar, Cislunar, Libration | Deep Space |
|---|---|---|---|---|
| ORS-1 | OCO-2 | **AsiaSat 3 Rescue**<br>GTO | **WMAP**<br>Sun-Earth L2 | **NEAR**<br>Asteroid |
| Chandra | JPSS | **Artemis**<br>GEO | **LRO**<br>Moon | **CONTOUR**<br>Comet |
| STSS 1&2 | STS | **DSCS III**<br>GEO | **LCROSS**<br>Moon | **ISEE-3**<br>Comet |
| AMOS | GeoEye 1-2 | **UFO – All**<br>GEO | **ARTEMIS-THEMIS**<br>Earth-Moon L1, L2, Moon | **New Horizons**<br>Pluto, Arrokoth |
| Cygnus | OrbView 2-3 | **FLTSAT – All**<br>GEO | **LADEE**<br>Moon | **Messenger**<br>Mercury |
| Skysat 3-14 | Suomi NPP | **MUOS – All**<br>GEO | **DSCOVR**<br>Sun-Earth L1 | **STEREO**<br>Heliocentric |
| CloudSat | Glory | **GEOStar Series Ascent**<br>GEO | **SELENE**<br>Moon | **MAVEN**<br>Mars |
| STPSat-2 | KOMPSat 1-3 | **COMS**<br>GEO | **Beresheet**<br>Moon | **MOM**<br>Mars |
| IKONOS | ICESat2 | **MEV**<br>GEO | **IBEX**<br>Lunar Resonant, HEO | **Parker Solar Probe**<br>Heliocentric |
| | | **Van Allen Probes/RBSP**<br>HEO | **SOHO**<br>Sun-Earth L1 | **Akatsuki**<br>Venus |
| | | **ERG**<br>HEO | **SLIM**<br>Moon | |

**Table 1**: Missions utilizing STK Astrogator in ops and other phases of the mission lifecycle

preconfigured template solutions to multiple rendezvous and proximity operations (RPO) strategies have been constructed and are included now by default with Astrogator.

RPO strategies are often relatively complex and require time, experience and an appropriate skill set to develop. Indeed, some of the many skills required may be listed:

- Fluency in the associated fundamental astrodynamics and orbital mechanics
- Proficiency in an appropriate simulation framework such as Astrogator
- Familiarity with numerical methods for trajectory construction and their nuances
- Knowledge and experience for orchestrating independent mechanisms through scripting or other high-level methods

Fortunately, many of these required proficiencies for constructing RPO strategies can be offset by leveraging existing resources. The various options now included with Astrogator follow such an approach having been produced from extensive experience in the associated domain. Thus, the approaches can generally be applied quickly and efficiently even in situations where expert guidance in all the aforementioned areas may not be available.

Starting in STK 12.2 multiple strategies have been packaged as pre-configured Astrogator sequences and made available as part of the STK installation. These include capabilities to address RPO mission phases that have been categorized and listed in Tables 2, 3 and 4. Additional strategies are both under consideration as well as development to be included

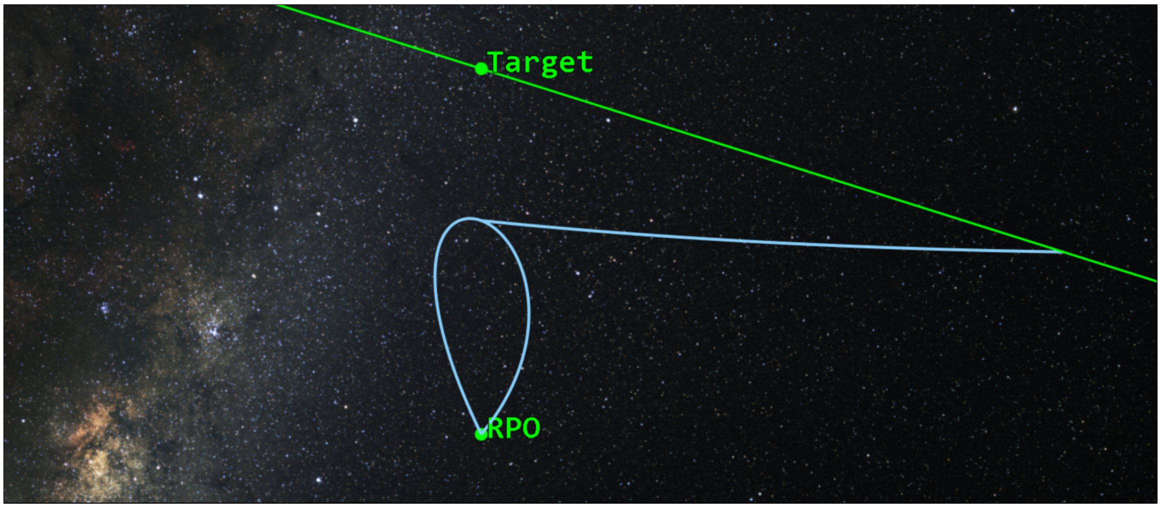


**Figure 2**: An RPO sequence moving the chase vehicle (RPO) from the VBar of the target vehicle (Target) into a *teardrop* orbit relative to the target

in future versions of Astrogator as pre-configured sequences. Goals for including these templates with Astrogator include reduced spin-up time for operators and analysts as well as providing starting points for users to customize and extend the provided techniques.

While each of the sequences are described briefly in the tables below and the general categorization is generally self-explanatory, some commentary is warranted. The configuration sequences are intended to automate STK and Astrogator settings that are commonly useful for performing RPO analysis. Operations such as setting a reference vehicle, updating spacecraft parameters and so on are all things that can be manually accomplished in the Astrogator user interface. However, these configuration sequences, as well as the various other RPO sequences, have settings pre-selected that are more suitable for RPO work. The categories denoted "Forced Motion", "Differential Forces" and "Matched Forces" are associated with the underlying models and approaches an operator would use to address corresponding problems. Natural motion solutions are often useful while certain operations require active control to maintain a relative motion strategy. Of course, when differential forces are acting on the vehicles, appropriate control strategies are necessary to counteract natural drift. A few rendezvous sequences are provided and others are under development. Finally, some sequences represent more specialized operations such as approaching or departing the *VBar* and *RBar*[*] as well as in a more general sense. One set of sequences not described in the tables are a set of supporting sequences that also ship with STK and are used for autosequences and other purposes within some of the listed strategies. More detailed information is also available from the Astrogator online documentation.[18]

[*]The so-called VBar is the direction along the target vehicle's orbit consistent with its velocity, while the RBar is the radial outward direction from the central body to the target vehicle. Each have both positive and negative senses with the target representing the origin.

**Table 2**: RPO Configuration, Differential Force and Forced Motion Strategies

| Category | Sequence/Strategy | Purpose/Description |
|---|---|---|
| Configuration | Coast | Propagate the spacecraft state under given forces until certain RPO-framed conditions are met |
| | Set Initial State | Pre-configured to set the chase vehicle's initial state in a frame relative to the reference vehicle |
| | Set Reference Vehicle | Facilitate the selection of the vehicle with respect to which RPO strategies will be applied |
| | Set DeltaV | Automate the process of configuring a relative maneuver and the selection of properties |
| | Update Spacecraft Parameters | Set particular spacecraft parameters from an RPO-relevant set of options |
| Differential Forces | Maintain NMC | Maintain a natural motion circumnavigation about target vehicle in the presence of differential forces |
| | Maintain VBar | Maintain a relative position along the VBar of the target satellite experiencing differential forces |
| Forced Motion | FM Circumnav | Actively control the chase vehicle to perform a forced motion circumnavigation about the target |
| | FM Waypoints | Navigate the vehicle about (or nearby) the target through a series of waypoints |
| | Follow Sun | Maintain a position along the target–Sun line for desirable lighting conditions on the target vehicle |
| | Hop | Move from one position relative to the reference vehicle to another using a single maneuver |
| | Hop Min DV | Like the *Hop* sequence but searching for the optimal time to minimize the maneuver cost |
| | Hop and Stop | Like the *Hop* sequence with a second maneuver to stop relative motion upon arrival at the destination |
| | Perch Equal Spacing | Maintain a point relative to the target using forced motion waypoints bounded within a small box |
| | Perch Max Error | Like the *Perch Equal Spacing* sequence with direct control over allowable error in holding position |
| | Teardrop | A relative orbit along the RBar with a maneuver at the turn-around point (see Fig. 2) |
| | VBar Hop | Relocate from one position on the VBar to another relative to a target vehicle using half of an NMC |

**Table 3**: RPO Specialized RBar and VBar Strategies

| Category | Sequence/Strategy | Purpose/Description |
|---|---|---|
| Specialized RBar/VBar | NM Circumnav to RBar | Follow a natural motion circumnavigation and stop on the $\pm$ RBar |
| | NM Circumnav To VBar | Follow a natural motion circumnavigation and stop on the $\pm$ VBar |
| | RBar Approach | Actively control the chase vehicle to approach the target vehicle along the RBar |
| | RBar Hop | Using half of a teardrop relative orbit, hop from one location along the RBar to another |
| | RBar to NM Circumnav | Initiate a natural motion circumnavigation about the target vehicle from the RBar |
| | RBar to VBar | Using a portion of an NM circumnavigation, move from the RBar to the VBar |
| | VBar Approach | Actively control the chase vehicle to approach the target vehicle along the VBar |
| | VBar to NM Circumnav | Initiate a natural motion circumnavigation about the target vehicle from the VBar |
| | VBar to RBar | Using a portion of an NM circumnavigation, move from the VBar to the RBar |

**Table 4**: RPO Matched Force, Rendezvous and Specialized Strategies

| Category | Sequence/Strategy | Purpose/Description |
|---|---|---|
| Matched Forces | NM Circumnav | Calculate the initial conditions and place the chase vehicle in that state for a natural motion circumnav |
| | VBar | Place the chase vehicle in a VBar relative orbit with respect to the target |
| Rendezvous | Exit Eccentricity Vector | Initiate a drifting orbit relative to the target satellite while maintaining the eccentricity vector |
| | Exit GEO | Initiate a circular drifting GEO orbit, using two maneuvers, to depart the vicinity |
| | GEO to GEO Drifting | Rendezvous one GEO satellite with another using a three-maneuver method |
| | GEO to GEO No Lead | Rendezvous one GEO satellite with another using a five-maneuver method |
| Specialized | Match Plane Single Burn | Match the chase satellite's orbit plane to that of the target satellite with a single maneuver |
| | Phase Change | Adjust the phasing of the chase vehicle with respect to the target by drifting toward or away from it |
| | Stop Plane Cross | Facilitate the process of constructing a relative orbit that terminates on particular plane crossings |
| | Stop Relative Rate | Stop all relative motion between the chase and target vehicles at the current time |
| | Stop Relative Motion | Propagate the chase vehicle until it reaches a zero crossing in a relative rate |

## CIRCULAR RESTRICTED THREE-BODY PROBLEM

The introduction of the Circular Restricted Three-body Problem (CR3BP) model within the STK/Astrogator software architecture was completed after the first installment of this paper series, and a previously presented companion paper[5] provides a more comprehensive discussion of the model. This section provides some of the fundamental background and comments regarding the implementation in Astrogator. Some text in this section follows from the previous reference and is consistent with definitions from well-known sources.[19,20]

A useful and approximate model of a multi-body system is one that sufficiently captures the dominant features of the dynamical interactions. Formulating a dynamical model that reflects the gravitational interactions of three bodies produces a model that is sufficiently complex to reveal many important characteristics while remaining tractable. However, the general three-body problem possesses no closed-form analytical solution.[21] Thus, additional simplifications, such as those consistent with CR3BP, offer significant insight.

The CR3BP models the gravitational influence of two larger, massive *primaries* (e.g., the Earth and the Moon) evolving on circular orbits on a third, much smaller body of negligible mass (e.g., a spacecraft). These two primary bodies are designated as $P_1$ and $P_2$.

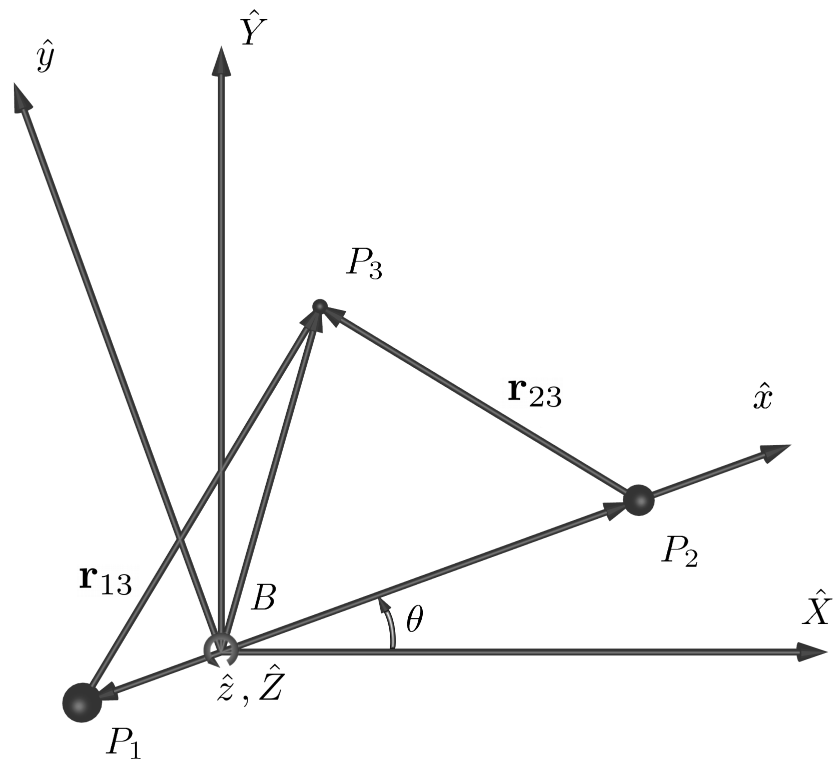


**Figure 3**: A pictorial schematic of the circular restricted three-body problem

The position variables $x$, $y$ and $z$ describe the position of the third body $P_3$, the spacecraft, with respect to the barycenter $B$ of the primary system and are defined in the rotating frame

$(\hat{x}, \hat{y}, \hat{z})$. This rotating frame is depicted in Figure 3 relative to an inertial reference frame $(\hat{X}, \hat{Y}, \hat{Z})$. The system mass parameter is represented by $\mu_{\text{CRP}} = \frac{m_2}{m_1+m_2}$, a function of the masses or mass parameters of the primary bodies. Additionally, distances between the third body and each of the massive primaries are denoted $r_{i3}$. In a coordinate frame that rotates coincident with the circular motion of the primaries, a system of differential equations that describes the motion of the third body is written

$$\ddot{x} = \frac{\partial U^*}{\partial x} + 2\dot{y}, \quad \ddot{y} = \frac{\partial U^*}{\partial y} - 2\dot{x}, \quad \ddot{z} = \frac{\partial U^*}{\partial z}, \tag{1}$$

with the pseudo-potential function defined as

$$U^* = \frac{1-\mu_{\text{CRP}}}{r_{13}} + \frac{\mu_{\text{CRP}}}{r_{23}} + \frac{1}{2}(x^2+y^2). \tag{2}$$

First derivatives in $x$ and $y$ appear in the equations of motion as a result of the Coriolis acceleration, and typical formulations incorporate nondimensionalization by characteristic system quantities in length, time and mass.

The implementation of the CR3BP equations of motion for use in Astrogator relies on instantaneous transformations to produce a hybrid system that evolves under these equations of motion. At each time step, Astrogator redefines the system, to account for all aspects of the motion of the secondary. If the secondary body in the three-body system follows natural motion (i.e., non-circular), the associated motions will be incorporated into the instantaneous transformations inducing librations and pulsations in the rotating frame inconsistent with the CR3BP dynamics. However, if the secondary motion for the system is appropriately defined (i.e., moving on a circular orbit about the primary, etc.) the result is consistent with the circular restricted model. Thus, the implementation of the CR3BP in Astrogator assumes a guided framework where the environment is properly and precisely configured. Aside from these considerations, the numerical implementation is generally consistent with typical numerical integration approaches. The process is principally one of frame transformations performed in a specific order to prepare the position and velocity state variables to be utilized as inputs for the dynamical model. Once appropriately cast, the model is evaluated and the necessary quantities retained. Subsequent transformations are performed based on the propagator definition.

## STATE TRANSITION MATRIX

Many analysis and design processes require information not only of a particular solution but also of the behavior about that solution in the design space. Frequently, the question of what happens if reality is slightly perturbed from the original design reference must be considered. Such perturbations may arise as a consequence of insufficiently modeled forces or partially characterized responses to system input. For example, high-fidelity models for spacecraft motion can never perfectly account for all forces acting on the vehicle—these models can yield extremely accurate and precise results but, ultimately, they remain *models*. Additionally, spacecraft maneuvers introduce some measure of uncertainty into the trajectory upon execution. Maneuver performance may become well known and anticipated over the course of multiple burns, but initial maneuvers will often perform, at least slightly, inconsistent with expectations resulting in too much or too little adjustment to the spacecraft path or attitude. Consequently, it is often extremely valuable to have some mechanism for assessing and characterizing the behavior adjacent to the original solution. One option for such assessment is an examination of the linear flow associated with a given reference solution to the underlying model using the state transition matrix.

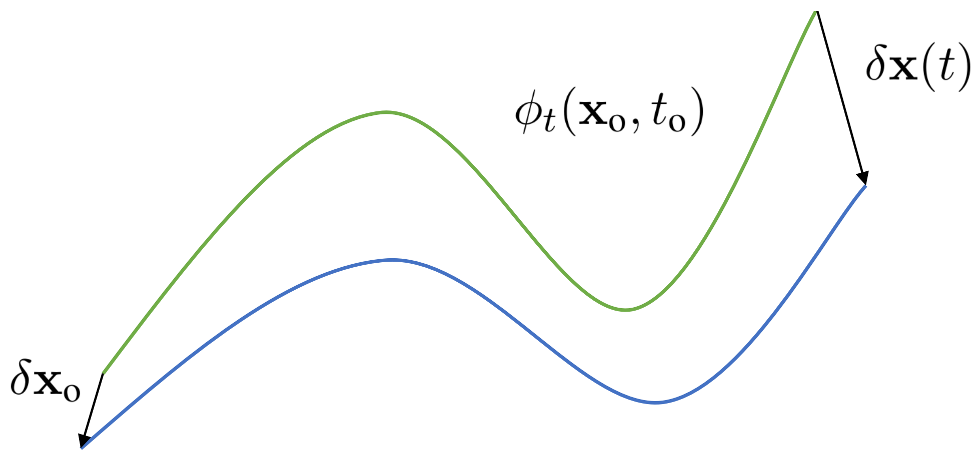


**Figure 4**: Nominal (green) and perturbed (blue) trajectories with initial and final variations

The state transition matrix (STM) is a matrix that captures the effects of an initial state variation $\delta\mathbf{x}_o$ along a solution $\phi_t(\mathbf{x}_o, t_o)$ resulting in a final state variation $\delta\mathbf{x}(t)$ after a duration $t$. This concept is captured in Fig. 4 where the nominal path is drawn in green above a perturbed path in blue along with vectors indicating the initial and final variations along the path. The STM can be produced by numerical integration of the variational equations corresponding to the underlying model, frequently posed as a system of first-order differential equations.

$$\dot{\mathbf{x}} = f(\mathbf{x}, t), \qquad \mathbf{x}(t_o) = \mathbf{x}_o \tag{3}$$

The solution $\phi_t(\mathbf{x}_o, t_o)$ necessarily satisfies the differential equations such that

$$\dot{\phi}_t(\mathbf{x}_o, t_o) = f(\phi_t(\mathbf{x}_o, t_o), t)\,, \qquad \phi_{t_o}(\mathbf{x}_o, t_o) = \mathbf{x}_o \tag{4}$$

It follows from Equation 4 that the derivative of the solution $\phi_t(\mathbf{x}_o, t_o)$ with respect to the initial state, denoted $\mathbf{\Phi}_t(\mathbf{x}_o, t_o)$, will result in the variational equations[22]

$$\dot{\mathbf{\Phi}}_t(\mathbf{x}_o, t_o) = \frac{\partial f(\phi_t(\mathbf{x}_o, t_o), t)}{\partial \mathbf{x}} \mathbf{\Phi}_t(\mathbf{x}_o, t_o)\,, \qquad \mathbf{\Phi}_{t_o}(\mathbf{x}_o, t_o) = \mathbf{I} \tag{5}$$

Where $\mathbf{\Phi}_t(\mathbf{x}_o, t_o)$ is the STM and $\mathbf{I}$ is the identity matrix. Referring back to Fig. 4, the associated depiction can be captured notationally as: $\delta\mathbf{x}(t) = \mathbf{\Phi}_t(\mathbf{x}_o, t_o)\,\delta\mathbf{x}_o$.

The computation and availability of the STM in Astrogator has undergone a series of incremental advancements, each increasing the usefulness of the associated capabilities for analysis. Recent iterations have brought these options more fully to the fore and a detailed discussion of these capabilities is offered in a companion paper by Kinzly et al.[3] The associated paper discusses the history of the STM in STK Astrogator, elaborates on the mechanisms that supply the capability and provides some analysis and validation of the underlying implementation.

## ADDITIONAL ADVANCEMENTS

The software system represented by Astrogator within the Systems Tool Kit is constantly experiencing direct and ancillary development. Frequently, advancements to another part of the larger toolset directly benefit Astrogator and its user base. A few examples of such direct and indirect improvements are briefly described below.

### Improved Automation Capabilities with Python

While the introduction of the STK Python API is not relegated to functionality associated only with STK Astrogator, it is generally of interest to Astrogator users. STK has long offered multiple APIs utilizing various platforms and technologies, and the newest API for use with Python became available with STK 12.1. This API offers various advantages including cross-platform support and the opportunity to utilize the many Python libraries to enhance simulation orchestration and scenario analysis. Further, in STK 12.2, a built-in mechanism for leveraging Jupyter notebooks directly within STK to interact with and control the functionality of the system was introduced. Astrogator users are among the users of STK that make significant use of any and all APIs and these new Python capabilities expand that toolset. The Programming Help for STK[23] contains additional information about these advancements and also provides documentation for the Integrated Jupyter Notebooks.[24] Additional Python integration within STK and Astrogator is in the pipeline for future implementation.

### Improved Ephemeris Handling

Over the past several releases of STK Astrogator, steps have been taken to improve ephemeris handling. Astrogator satellites that contain many hundreds of segments previously suffered from poor load, save and post-run processing times. Adjustments to the save/load and ephemeris lookup processes has dramatically improved each of these operations. As an example, a low-thrust spiral trajectory from LEO to GEO composed of roughly 60,000 segments produced by autosequence implementation of control strategies exhibited load times and run post-processing times on the order of an hour. This has been reduced to a few minutes. Additional steps to make such improvements are always underway, and the STK/Astrogator developers are always open to feedback.

### Expanded Support for user Data in Plugins

Astrogator offers a system of user variable propagation along with its ephemeris. A user may initialize such data members from one of the segments that produces an initial state, update them in various places and supply plugin code to evolve the variables in conjunction with the Astrogator state. Previously, such user variable support was restricted to force model (EOM function) plugins. With STK 12.2, both drag and solar radiation pressure (SRP) plugins now support the propagation of user variables. The introduction of user variables for other plugin points is also an item in the Astrogator development pipeline.

## TEST-DRIVEN DEVELOPMENT AND QUALITY ASSURANCE PROCESSES

The main goal of the STK quality assurance process is to maintain and improve the quality of the software. The process also aims to ensure that new features are added efficiently, customer upgrade requests are satisfied, and user interface options are maintained and expanded. These goals are achieved when program managers, development teams, and test teams work together to drive the following steps of the QA process:

- Define and review requirements for new features or requested improvements
- Design and implement the necessary code changes
- Test and verify the results
- Identify and confirm release tests and release the software
- Summarize lessons learned for continued improvement

Every person plays an integral role, and it is critical to satisfy each milestone in the QA process before new software can be released. The remainder of this section will focus on the testing and verification step of the process, specifically how it applies to Astrogator and the recently added features discussed in this paper.

Astrogator is rigorously tested and built on a daily basis through STK's continuous integration and build process, occurring in a dedicated build farm of 60 machines. A complete daily install of STK is produced from this process and automatically exercised through the regression test suite. Regression test runs are currently processing over 700 test scripts and executing several million commands through the API, in parallel, on a test bank of 40 machines. With each new feature or capability added to the software comes a new test, or set of tests, added to the regression suite. At present, 108 tests are dedicated to Astrogator's growing catalog of features, with 39 of those being recently added to cover the newly incorporated RPO sequences alone.

Quality assurance and testing for the CR3BP and STM capabilities follows the more typical QA path. Numerical results are validated by the developer as part of the software development process. Once the feature is fully incorporated into STK, the test engineer reproduces and verifies the developer's results through a specified test procedure and manual testing. This test procedure is then used as the basis for creating an automated test, and is often expanded upon to include scenarios an STK user might employ out in the field. Regression test results are monitored daily and compared against a baseline result to identify differences or failures, which are investigated immediately.

Incorporating the RPO strategies to make them available as part of the STK installation package provided the ideal opportunity to apply a more test-driven approach to the development process,[25] since the strategies already existed as pre-configured Astrogator sequences. This allowed an automated test to be created for each RPO sequence before any code changes were made to incorporate them. The initial test results were validated against the existing RPO documentation, and in this way an acceptable baseline was created. As code changes were implemented, the RPO regression tests helped identify what the next step in the development approach should be. This often included updating the tests as well. In this way, an iterative process between code updates and test updates helped quickly and efficiently deliver the robust RPO templates that are now available to all Astrogator users. This test-driven approach proved to be time-saving and productive, and other areas of STK development are being considered where this approach can be applied.

## CONCLUDING REMARKS AND FUTURE WORK

Future installments of this paper series are intended given sufficient interest from the community. Further, two of the key recent advancements discussed in this paper, the introduction of the CR3BP model and advancements to STM functionality, have both been addressed in greater depth.[3,5] A similar, more-detailed discussion of the RPO capability additions is also warranted and intended as a future companion paper to the series.

As space missions continue to require more from designers, operators and analysts to the point that there is more work than there are people to do the work, having efficient and effective tools is key. Further, established reliability and confidence in these tools is paramount. Consequently, some insight into the associated processes for developing and maintaining the software is useful. Up-to-date information regarding the status of recent and current developments in the toolset is also important to help users understand what is new as well as the associated proving level of those capabilities. Tools under continuous development are most effective when users are aware of the associated advances, and a driving goal of this paper series is to help communicate how the Astrogator toolset may best be used.

## ACKNOWLEDGMENTS

As always, the authors wish to thank the documentation, support and training teams at AGI for their incredible capacity to distill and convey the complex concepts of Astrogator to the end users. Frequently, the workflows to solve particular astrodynamics problems can be daunting—these processes are made approachable by the very talented folks on the front lines. Further, the authors are grateful for reviews of various portions of the text by Vince Coppola, Sylvain Dupont, Ryan Lee, Johnathon Thompson and Frank Urzynicok.

## REFERENCES


[1] T. A. Lovell, S. G. Tragesser, and M. V. Tollefson, "A Practical Guidance Methodology for Relative Motion of LEO Spacecraft based on the Clohessy–Wiltshire Equations," *AAS/AIAA Space Flight Mechanics Meeting*, Maui, Hawaii, February, 2004.

[2] J. Sullivan, S. Grimberg, and S. D'Amico, "Comprehensive Survey and Assessment of Spacecraft Relative Motion Dynamics Models," *Journal of Guidance, Control and Dynamics*, Vol. 40, No. 8, 2017, pp. 1837–1859.

[3] N. Kinzly, S. Chikine, J. Carrico, M. Loucks, and C. Short, "Directly Leveraging the State Transition Matrix in STK Astrogator," *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, Montana [virtual], August, 2021.

[4] C. Short, P. Ghosh, and A. Claybrook, "Revisiting Trajectory Design with STK Astrogator, Part 1," *AAS/AIAA Astrodynamics Specialist Conference*, Portland, Maine, August, 2019.

[5] C. Short, A. Haapala, and N. Bosanac, "Technical Implementation of the Circular Restricted Three-body Model in STK Astrogator," *AAS/AIAA Astrodynamics Specialist Conference*, South Lake Tahoe, California [virtual], August, 2020.

[6] J. Carrico and E. Fletcher, "Software Architecture and Use of Satellite Tool Kit's Astrogator Module for Libration Point Orbit Missions," *Libration Point Orbits and Applications*, Girona, Spain, June, 2002.

[7] Johns Hopkins University Applied Physics Laboratory, "Solar 60: Parker Solar Probe Mission Design," July 2018. Online: https://www.youtube.com/watch?v=rd7YwYOuDf4, Accessed: 30 July 2021.

[8] Y. Guo and R. Farquhar, "New Horizons Mission for the Pluto–Kuiper Belt Mission," *AAS/AIAA Astrodynamics Specialist Conference*, Monterey, California, August, 2002.

[9] E. Lu, M. Loucks, and J. Carrico, "The Distribution of Required Deflection Impulses as a Function of Time Before Impact for Earth Impacting Asteroids," *IAA Planetary Defense Conf.*, Tokyo, Japan, 2017.

[10] M. M. Berry and V. T. Coppola, "Correct Modeling of the Indirect Term for Third Body Perturbations," *AAS/AIAA Astrodynamics Specialist Conference*, Mackinac Island, Michigan, August, 2007.

[11] M. M. Berry and V. T. Coppola, "Integration of Orbit Trajectories in the Presence of Multiple Full Gravitational Fields," *AAS/AIAA Space Flight Mechanics Meeting*, Galveston, Texas, January, 2008.

[12] M. M. Berry, "Comparisons Between Newton–Raphson and Broyden's Methods for Trajectory Design Problems," *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Alaska, August, 2011.

[13] M. A. Mesarch and S. F. Andrews, "The Maneuver Planning Process for the Microwave Anisotropy Probe (MAP) Mission," *AAS/AIAA ASC*, Monterey, California, August, 2002.

[14] AGI Marketing, "Astrogator Specifications Sheet," July 2021. Online: https://www.agi.com/resources/documents/astrogator, Accessed: 16 July 2021.

[15] J. P. Carrico, H. L. Hooper, L. Roszman, and C. Gramling, "Rapid Design of Gravity Assist Trajectories," *ESA Symposium on Spacecraft Flight Dynamics*, Darmstadt, Germany, September–October 1991.

[16] T. Harrison, K. Johnson, T. G. Roberts, T. Way, M. Young, and M. C. Faga, "Space Threat Assessment 2020," tech. rep., Center for Strategic and International Studies (CSIS), 2020.

[17] T. Harrison, K. Johnson, J. Moye, T. Way, and M. Young, "Space Threat Assessment 2021," tech. rep., Center for Strategic and International Studies (CSIS), 2021.

[18] AGI Documentation Team and Subject Matter Experts, "Astrogator RPO Documentation," March 2021. Online: https://help.agi.com/stk/#gator/rpo_intro.htm, Accessed: 20 July 2021.

[19] V. Szebehely, *Theory of Orbits: The Restricted Problem of Three Bodies*. Academic Press, 1967.

[20] K. C. Howell, "Three-Dimensional, Periodic, 'Halo' Orbits," *Celestial Mechanics*, Vol. 32, No. 1, 1984.

[21] F. Diacu, "The Solution of the *n*-body Problem," *Math. Intell.*, Vol. 18, No. 3, 1996, pp. 66–70.

[22] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems*. Springer–Verlag, New York, 1989.

[23] AGI Documentation Team and Subject Matter Experts, "STK Python API Documentation," March 2021. Online: https://help.agi.com/stkdevkit/index.htm#python/pythonIntro.htm, Acc.: 20 July 2021.

[24] AGI Documentation Team and Subject Matter Experts, "STK Integrated Jupyter Notebooks for Python Plugin Documentation," March 2021. Online: https://help.agi.com/stk/#pythonIntegratedJupyterNotebooksPlugin.htm, Accessed: 20 July 2021.

[25] K. Beck, *Test-driven Development by Example*. Addison-Wesley, 2003.