AAS 22-102

A NEW GUIDANCE TECHNIQUE FOR DISCRETE-EVENT DRAG MODULATION FOR AEROCAPTURE MISSIONS

Ethan R. Burnett[*], Samuel W. Albert[†], and Hanspeter Schaub[‡]

This paper introduces a new guidance algorithm for discrete-event drag-modulated aerocapture, inspired by the concepts of variation-of-parameters and osculating orbits. The algorithm is implemented and tested in a high-fidelity simulation developed by NASA's Jet Propulsion Laboratory for the case of an Earth flight demonstration of aerocapture using a rigid drag skirt. The algorithm takes the spacecraft navigation-estimated position and velocity and computes a jettison time for the drag skirt, and the performance of the new algorithm is compared to a numerical predictor-corrector (NPC) approach.

INTRODUCTION

Aerocapture has been studied as a favorable means of reducing the necessary propellant for interplanetary missions to bodies with significant atmospheres, such as Venus, Mars, Titan, and the ice giants.[1,2] Instead of the traditional technique of performing an orbital insertion burn upon arrival at the target body, the space vehicle passes through the atmosphere, reducing planetocentric velocity significantly while safely protected by an aeroshell. After the hypersonic atmospheric encounter, the vehicle is captured into a bounded orbit about the target body, and a periapsis-raise maneuver completes the capture orbit. In practice, for a passive capture scenario, the high degree of uncertainty about atmospheric conditions (particularly the profile of density vs. altitude) can force large errors in the planned capture orbit apoapsis. As a result, various control methods have been proposed to reduce the target apoapsis error. Discrete-event drag modulation presents a solution that is operationally and mechanically simple, in which the vehicle changes its area-to-mass ratio during the hypersonic flight via jettison of a drag device. For properly chosen entry conditions, the final apoapsis is a function of the jettison timing, and thus a properly timed jettison event can significantly reduce the capture apoapsis error.[3]

For single-stage drag modulation applications to aerocapture problems, there are various methods for computing the timing of the discrete control event. A frequently explored approach is to use a numerical predictor-corrector (NPC) algorithm to estimate the drag modulation event timing using the best available dynamic model of the atmospheric density and an estimate of the current spacecraft state.[4] One way to do this without dedicated sensors is to estimate the density from the aerodynamic acceleration, comparing it to the expected density at that altitude. The mismatch between observed and expected density then informs the dynamic model in the predictor on the

[*]Ph.D., Aerospace Engineering Sciences, University of Colorado Boulder
[†]Research Assistant, NSTRF Fellow, Aerospace Engineering Sciences, University of Colorado Boulder
[‡]Glenn L. Murphy Chair of Engineering, Smead Department of Aerospace Engineering Sciences, University of Colorado, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO 80309-0431. AAS Fellow, AIAA Fellow.

approximate scaling of the atmospheric density for the remainder of the pass through the atmosphere. However, this method is somewhat susceptible to navigation error because if the density look-up uses an incorrect altitude estimate, it results in a bias in the computed density scale factor. In addition, the method can be susceptible to noise and oscillations in the computed density scale factor. Furthermore, for most conceivable implementations, it is difficult to ensure that a guidance computer will always be able to support the NPC algorithm at a runtime frequency that is compliant with mission requirements. This is because the NPC algorithm is not computationally well-posed – it is difficult to estimate or bound the number of iterations required for convergence in a given call to the algorithm.

This work discusses an alternate approach that does not directly deal with computing a density scale factor at all, and additionally is computationally well-posed, such that the runtime is independent of the atmospheric descent dynamics and can be easily characterized and bounded. The approach relies on estimating the state at some time before entry, using it to generate nominal capture trajectories from a range of atmospheric density profiles, and additionally computing the sensitivity of each associated trigger time to initial height, velocity, and flight path angle. The measured state during descent is used to identify the nearest nominal trajectory, and the dynamic model for that trajectory is used to back-propagate the estimated state to a new perturbed entry condition, which informs a correction to the estimated jettison time. For the case that the perturbed entry conditions are only small deviations, the correction can be achieved via pre-computed linear sensitivities. For larger deviations, nonlinear corrections are also possible with sufficient sampling. This approach benefits from being easily scaleable with on-board computational capability, because the computation time for the nominal trajectories can be bounded. The subsequent execution of the algorithm also relies on a single back-propagation per algorithm call, instead of multiple forward propagations.

This work introduces a new guidance algorithm based on quasi-initial conditions (QICs),[5] called QIC guidance. The details of QICs are reviewed, and the performance of QIC guidance is compared to an NPC algorithm implementation for simulations of an aerocapture demonstrator mission, and their relative performance and trade-offs are discussed. The algorithms are tested in a high-fidelity simulation environment that was written using the Dynamics Simulator for Entry, Descent, and Surface Landing (DSENDS) software developed by the DARTS lab at NASA's Jet Propulsion Laboratory (JPL).[6] The highly realistic dynamic modeling enables rigorous testing of both algorithms.

PRELIMINARIES

Aerocapture with ADEPT

The application of interest in this work is drag modulation timing for aerocapture using the Adaptable Deployable Entry and Placement Technology (ADEPT).[7] This device uses a conical deployable structure (the "drag skirt") with a 3D woven carbon fabric skin for drag modulation and thermal protection, and the thermally protected understructure supports the aerodynamic deceleration forces generated on the leading surface. The drag skirt is jettisoned at a particular time computed to achieve a desired final capture orbit apoapsis, but the effects of atmospheric uncertainty after jettison result in some final orbit error. The management of the effects of atmospheric uncertainty is partly delegated to judicious planning of the atmospheric descent profile. However, the jettison time algorithm is an equally important factor in successful aerocapture. An ideal jettison time algorithm will efficiently compute a time that minimizes any deterministic orbit error and the integrated effects of atmospheric density deviations up until the jettison time.

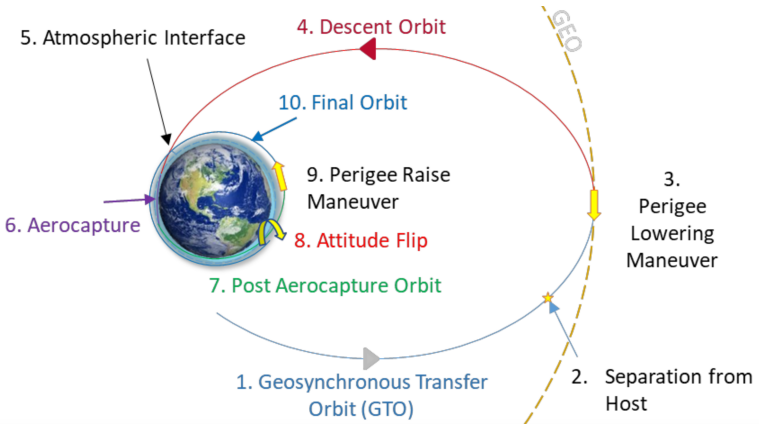


Figure 1: Aerocapture Earth Test

A multidisciplinary team from NASA and academia, led by JPL, has been studying aerocapture for small satellites.[8] The simulations used in this work were tested mainly using the existing software infrastructure developed by that team. The particular simulation used in this work is an Earth-based flight demonstration of aerocapture using the ADEPT architecture. The test scenario is depicted in Figure 1, borrowed from Reference 8. The descent trajectory is an elliptic orbit with an apoapsis radius at GEO and a periapsis in the Earth's atmosphere. The target orbit is a much lower Earth orbit, and the goal of the aerocapture test is to successfully lower the apoapsis radius to a target value, and then to raise the periapsis out of the atmosphere via a subsequent burn. The timing of the jettison of the drag device is critical for achieving the target apoapsis.

Problem Dynamics

The algorithms in this work assume that the space vehicle is subject to the dynamics of two-body planetary gravity, J_2, and the atmospheric drag acceleration:

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^2}\hat{\boldsymbol{r}} - \frac{3\mu J_2 R^2}{2r^4}\left(\left(1 - 5\left(\hat{\boldsymbol{r}}\cdot\hat{\boldsymbol{K}}\right)^2\right)\hat{\boldsymbol{r}} + 2\left(\hat{\boldsymbol{r}}\cdot\hat{\boldsymbol{K}}\right)\hat{\boldsymbol{K}}\right) - \frac{\rho v_{\mathrm{f}}^2}{2\beta}\hat{\boldsymbol{v}}_{\mathrm{f}} \tag{1}$$

where ρ is the variable atmospheric density, R is the planetary equatorial radius, $\hat{\boldsymbol{K}}$ is the polar axis unit vector, and $\beta = m/(C_D A)$ is the ballistic coefficient of the vehicle. The quantities m, C_D, and A are the mass, drag coefficient, and reference area of the vehicle, respectively. The quantity $\boldsymbol{v}_{\mathrm{f}}$ is the flow velocity, or the velocity of the spacecraft with respect to the planetary atmosphere rotating with angular velocity $\boldsymbol{\omega}_p$:

$$\boldsymbol{v}_{\mathrm{f}} = \dot{\boldsymbol{r}} - \boldsymbol{\omega}_p \times \boldsymbol{r} \tag{2}$$

Lift effects are not considered – ADEPT is axisymmetric, and passive stabilization keeps its axis of symmetry close to alignment with $\hat{\boldsymbol{v}}_{\mathrm{f}}$. The DSENDS simulation models the net atmospheric forces on the spacecraft beyond the drag force (i.e., including key aerodynamic yaw, pitch, and roll moments) by interpolation of catalogued aerodynamic simulation data. These effects are secondary, and propagation of the dynamics given by Eq. (1) gives reasonable approximations of the trajectories simulated with DSENDS.

Quasi-Initial Conditions as a State Representation

The new guidance algorithm developed in this paper relies on a coordinate representation known as "quasi-initial conditions". The fundamental concept of the quasi-initial condition is illustrated in Figure 2. Initial conditions (here projected into initial entry flight path angle γ and velocity v) are

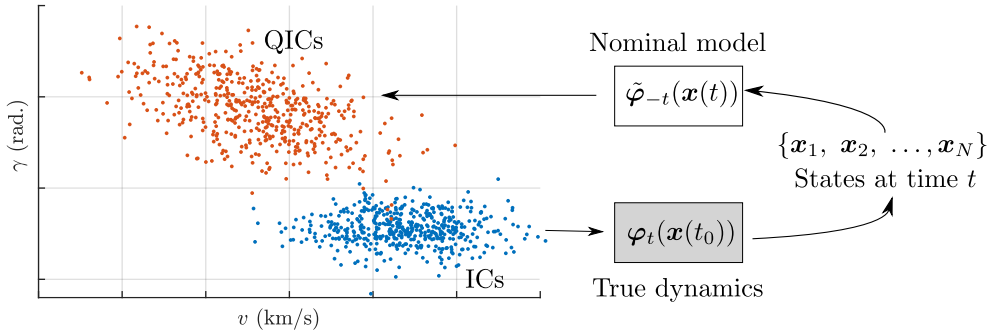


Figure 2: Quasi-Initial Conditions

propagated via the true dynamics to an evolving distribution of states. At any time t, these states can be back-propagated via a nominal model that is similar to the true dynamics, but deterministic and autonomous. The details and levels of fidelity of the nominal model are a choice left to the astrodynamicist. Because the nominal model is similar to the true dynamics, an initially Gaussian distribution of initial conditions yields a relatively slowly evolving distribution of quasi-initial conditions (QICs). Furthermore, the QICs constitute a rigorous and complete representation of the evolving capture orbit state, because the state at any time t can be recovered from the QICs by forward propagation of the nominal model. The propagation through the nominal model can thus be considered as a nonlinear coordinate transformation.

Reference 5 explores using the QICs as a state representation for aerocapture, and finds that propagated initially Gaussian distributions in these quantities have lower measures of non-Gaussianity than coordinate sets not rendered in the QIC space (i.e. not back-propagated). As a result, the effective dynamics in the QIC space can be considered to have a lower degree of dynamical nonlinearity than in other aerocapture state representations. Thus, the initially linear dispersions take longer to become significantly nonlinear. Crucially, large distributions of states are back-propagated to a more compact distribution of quasi-initial conditions. It is the persistent compactness and the slower distortion of the state distribution in the QICs that enables the new algorithm discussed in this paper. The algorithm, to be discussed, is inspired by the quasi-initial condition concept, as well as the classical variation-of-parameters concept employed in classical astrodynamics.[9]

ALGORITHMS FOR COMPUTING JETTISON TIME

Numerical Predictor-Corrector Algorithm

The classical NPC algorithm for discrete-event drag modulation is illustrated in Figure 3. The aerodynamic acceleration $\boldsymbol{a}_{\text{aero}}$ is estimated by the IMU acceleration signal. If it exceeds a specified threshold value, the NPC algorithm is called. First, the nominal density at the navigation-estimated

altitude is used with the navigation-estimated state to compute an estimate of the dynamic pressure:

$$q_{\text{est}} = \frac{1}{2}\rho_{\text{nom}}(r)v_{\text{f}}^2 \tag{3}$$

where the estimated dynamic pressure is used to obtain an estimate of the drag coefficient via interpolation of stored data of C_D vs. q.

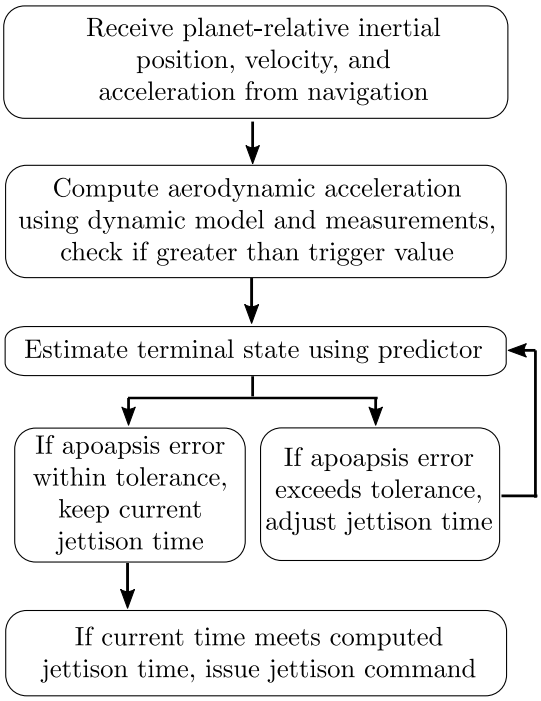


Figure 3: Numerical Predictor-Corrector Diagram

Next, the density is estimated from a re-arranged expression for aerodynamic drag force:

$$\rho_{\text{est}} = 2\frac{m_1 a_{\text{aero}}}{A_1 C_{D,1} v_{\text{f}}^2} \tag{4}$$

where m_1, A_1, and $C_{D,1}$ are the pre-jettison mass, area, and interpolated drag coefficient, respectively. The density estimate is used to compute the i^{th} density scale factor F_i:

$$F_i = \rho_{\text{est}}(t_i)/\rho_{\text{nom}}(r(t_i)) \tag{5}$$

This value is then filtered via a low-pass filter:

$$\overline{F}_i = (1-k)\overline{F}_{i-1} + k \cdot F_i \tag{6}$$

As the gain k is decreased, this filter will increasingly reject small disturbances. Sensible values of k depend on the frequency of density scale factor measurement updates. The density scale factor can also be filtered with a moving average filter, detailed below:

$$\overline{F}_N = \frac{1}{N}\sum_{i=1}^{N} F_i \tag{7}$$

where N is a memory parameter, and again the chosen value of N should be tuned based on the density scale factor update frequency.

For a given call of the NPC algorithm, the current value of the density scale factor is used to scale the on-board model of density vs. altitude in the following manner:

$$\rho_{\text{pred}}(r) = \overline{F}_i \rho_{\text{nom}}(r) \tag{8}$$

Using Eq. (8), systematic atmospheric density modeling error can be partially attenuated, but because true atmospheric density variations are the results of a highly complex process that is impossible to model on-board, the future state will naturally differ from the predicted state by an unpredictable amount.

In the predictor step, the trajectory is numerically propagated to atmospheric exit using the current value of the jettison time, and the resulting apoapsis radius is computed. The resulting final error in the apoapsis radius informs a correction to the jettison time. If the apoapsis error is positive, the jettison time is extended, and if the apoapsis error is negative, the jettison time is reduced. There are various means of implementing this logic.[10] Once the apoapsis error satisfies a tolerance value (typically no more than 50 km in our case), the jettison controller's trigger time is updated if needed. This concludes a single call to the NPC algorithm.

The indeterminate number of costly numerical propagations per call to the algorithm is an undesirable property of the NPC algorithm. Furthermore, the heavy reliance on navigation-estimated density leads the algorithm to be quite susceptible to navigation error. The density scale factor computation ends up partially cancelling out the effects of navigation error, but not completely.

An alternative approach is to use the estimated descent trajectory from the navigation state history to determine the scale of atmospheric perturbations. If multiple nominal trajectories are pre-propagated with varying scalings of density vs. altitude, then identifying the scale of atmospheric perturbation becomes as simple as comparing the current state to the states at a similar time in all of the nominal trajectories. This approach might be less susceptible to navigation error, removes the indeterminacy of the NPC algorithm, and deals with the highly uncertain density scale factor in an entirely implicit manner via careful examination of the estimated state in the context of other pre-computed descent profiles.

QIC Guidance Algorithm (Linear Correction)

The new QIC guidance algorithm using linear correction is illustrated in Figure 4. This approach is quite different from the NPC algorithm. Using an accurate state estimate from well before atmospheric entry, the nominal trajectories are generated and the state data is saved at some frequency. For this application, the state used is 10 minutes before entry, 8000 state vectors are saved at equally spaced times for $t \in [0, 1000]$ s, and the data is interpolated as needed. Each of these trajectories is generated using a unique density profile. In this work, the density profiles were each constant scalings of an on-board nominal model of density vs. altitude. The same propagation model used in the NPC predictor was used to generate the nominal trajectories, and the jettison times were computed in a manner similar to the NPC algorithm. The model accounts for the atmospheric drag as well as the J_2 term of the gravity field.

If the chosen density profiles are of the form $\rho_k(r) = F_k \rho_{\text{nom}}(r)$ where F_k is an element of a set of selected scale factor values, then the descent profiles smoothly vary and produce a wide variation in possible jettison times. This is illustrated in Figure 5, with a small set of F values and the nominal

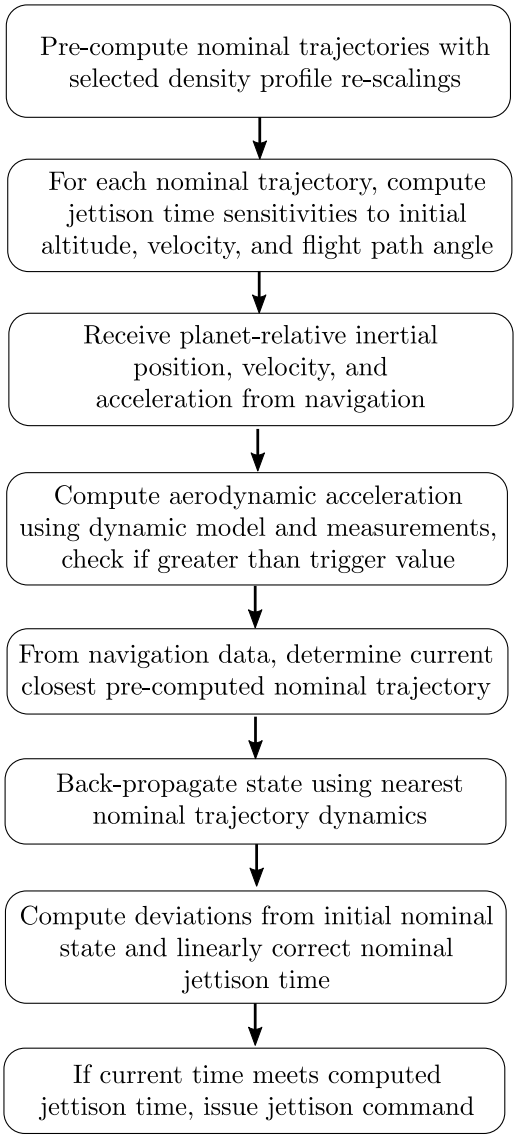


Figure 4: QIC Guidance Diagram

jettison points indicated by small square markers. Note that in this figure, the semimajor axis is used to concisely portray the descent profile data. Other quantities could be used, but this was convenient because the simulations run were for a highly elliptic Earth-based aerocapture hardware test, so the semimajor axis was always a well-defined measure of the orbit. More generally, the orbit energy could be used, with the hyperbolic measure $E = v_\infty^2/2$ smoothly transitioning through zero to the elliptic definition $E = -\frac{\mu}{2a}$ as the orbit captures.

For each nominal trajectory, it is also necessary to compute the sensitivity of the jettison time to the initial conditions. While the initial condition generally must be described by six quantities, three dominant quantities can be isolated for this application, whose variations at the initial time are by far the most important in impacting the jettison time. These are the altitude h_0, velocity magnitude

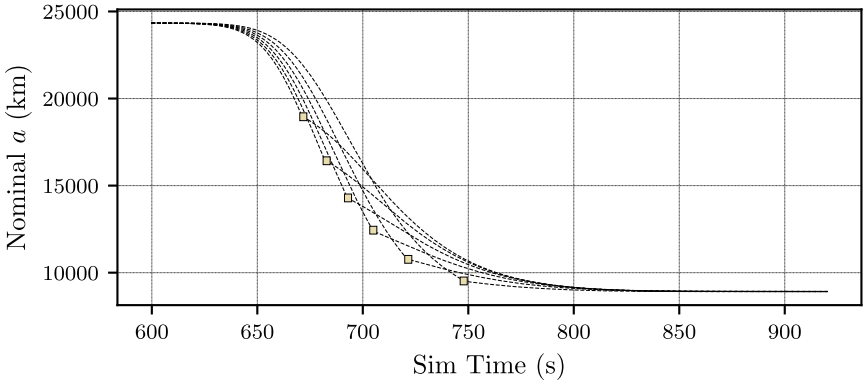


Figure 5: Nominal Trajectories and Jettison Events

v_0, and flight path angle γ_0, defined below:

$$h_0 = \|\boldsymbol{r}_0\| - R_p \tag{9}$$

$$v_0 = \|\boldsymbol{v}_0\| \tag{10}$$

$$\gamma_0 = \sin^{-1}\left(\hat{\boldsymbol{v}}_0 \cdot \hat{\boldsymbol{r}}_0\right) \tag{11}$$

where $\hat{\boldsymbol{r}}_0 = \boldsymbol{r}_0/r_0$ and $\hat{\boldsymbol{v}}_0 = \boldsymbol{v}_0/v_0$ are unit vectors and R_p is the planetary radius. In our application, the sensitivities to these three quantities respectively were on the order of 0.01 s/m, 1 s^2/m, and 1000 s/deg. The sensitivity to flight path angle is by far the most extreme. The sensitivities are computed via finite differencing, by adding small variations to the initial altitude, velocity, and flight path angle and computing the resulting change in jettison time. For this work, the resolution in jettison time is limited by the frequency that the guidance algorithm is run at. It is generally run at 8-10 Hz for the results in this paper.

To compute the sensitivities in our application via finite-differencing, the fundamental step sizes used were 150 m for initial altitude, 1.0 m/s for initial velocity, and 0.002 degrees for the initial flight path angle. These values were typically small enough for accurate linearization, but large enough to compute jettison time differences larger than the timing resolution.

Once all of the nominal trajectories, nominal jettison times, and sensitivities are computed, the most taxing execution in the algorithm is complete. The spacecraft is to wait until the atmospheric acceleration trigger value is reached, before entering the loop starting with the fifth block in Figure 4.

To determine from navigation data which nominal trajectory is closest, the simplest method is to compute some kind of difference metric between the navigation data and the stored trajectory data at the same time. For this work, that was of the following form:

$$d = c_1\left(a - a^*\right)^2 + c_2\left(\dot{a} - \dot{a}^*\right)^2 \tag{12}$$

where c_1 and c_2 are constants to tune the scale of relative importance of differences in semimajor axis and semimajor axis rate. The semimajor axis a is estimated from the present navigation state, and the semimajor axis rate is determined by differencing of the current estimated semimajor axis and a prior one. Note that a highly similar metric could be defined using the orbit energy and orbit

energy rate. With each call to the algorithm, Eq. (12) is computed for each nominal trajectory, and the trajectory yielding the lowest value of d is selected.

The nearness of the navigation state to a nearby nominal trajectory implies a similarity in atmospheric intensity for the two trajectories. Thus, the navigation state is to be back-propagated with the same density table that was used to generate the nearest nominal trajectory. This yields a new initial state (a quasi-initial condition) that differs from the initial state used to generate the nominal trajectories, and the initial state difference informs the correction in jettison time via the sensitivities. Essentially, the back-propagation step represents an osculating condition – the newly generated descent profile intersects the navigation estimate in state space at the measurement time. This approach was inspired by the well-known osculating orbit concept from orbital mechanics, except instead of defining a Keplerian orbit to intersect the present state, a perturbed trajectory is used instead.

The back-propagation details are now discussed. Assuming that the dynamic model is autonomous, the nonlinear dynamics of the propagator satisfy the following mapping relationship

$$\boldsymbol{\varphi}_{-t} \circ \boldsymbol{\varphi}_t(\boldsymbol{x}) = \boldsymbol{\varphi}_t \circ \boldsymbol{\varphi}_{-t}(\boldsymbol{x}) \tag{13}$$

where for example $\boldsymbol{\varphi}_t(\boldsymbol{x}_0)$ is the solution to $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ with $\boldsymbol{x}(0) = \boldsymbol{x}_0$. It is possible to approximate an jettison time correction via the following linear mapping from the deviated initial condition found via back-propagation of the navigation state $\boldsymbol{x}_{\text{nav}}(t)$:

$$t_j \approx t_j^* + \frac{\mathrm{d}t_j}{\mathrm{d}\boldsymbol{x}_0} \left(\boldsymbol{\varphi}_{-t}(\boldsymbol{x}_{\text{nav}}(t)) - \boldsymbol{x}_0^*\right) \tag{14}$$

Rewriting Eq. (14) in terms of the jettison time sensitivities and deviations in initial altitude, velocity, and flight path angle:

$$t_j \approx t_j^* + \frac{\mathrm{d}t_j}{\mathrm{d}h_0}\delta h_0 + \frac{\mathrm{d}t_j}{\mathrm{d}v_0}\delta v_0 + \frac{\mathrm{d}t_j}{\mathrm{d}\gamma_0}\delta \gamma_0 \tag{15}$$

Thus, the navigation state is back-propagated to time 0, and the new altitude, velocity, and flight path angle are computed, then the deviations are computed as $\delta h_0 = h_0 - h_0^*$, etc.

Because the jettison time correction is based on linear sensitivities, there is a limited region of validity for the initial deviations. If they get too large, the correction to the jettison time will tend to be excessive and erroneous. This could happen if the navigation state is still far from the nearest nominal trajectory. There are a few methods to address this case.

The first and simplest method would be to default to the nearest nominal jettison time if the linear correction on that time exceeds a certain amount. A second method would be to limit the absolute value of the linear correction to some maximal value. In a third approach, the navigation state could be partially weighed with the nearest nominal state, such that the state to be back-propagated $\boldsymbol{x}_b$ is sufficiently close to the nominal state:

$$\boldsymbol{x}_b = (1-\alpha)\boldsymbol{x}^*(t) + \alpha \boldsymbol{x}_{\text{nav}}(t) \tag{16}$$

where $0 \leq \alpha \leq 1$ is a tuning parameter, and the case of $\alpha = 0$ reproduces the no-correction result from the first method. In this case, the logic of the algorithm reduces to simply selecting the jettison time of the closest nominal trajectory. The choice of suitable values of α is going to be application-specific, so this approach is not explored numerically.

Future Improvements

While not explored in the results of this paper, one major improvement would be to replace the linear correction on the jettison time, Eq. (15), with a general nonlinear correction. As opposed to computing three independent sensitivities, a nonlinear correction would require sampling of the variations in the jettison time in a three-dimensional domain of initial height h_0, magnitude of the initial velocity v_0, and initial flight path angle γ_0. This would require far more computations than the linear correction. However, the initial condition could be defined spatially instead of temporally, as the point at which $\|\boldsymbol{r}\| - R_p = h_{\text{spec}}$ for a specified entry altitude h_{spec}. This would have the effect of removing the initial condition h_0 from consideration, reducing the dimension of important initial variations to 2, and also reducing the computational burden. With this change, the updated jettison time would be computed as below:

$$\begin{aligned} t_j &\approx t_j^* + f(v_0^*, \gamma_0^*, \delta v_0, \delta\gamma_0) \\ &= t_j^* + f^*(\delta v_0, \delta\gamma_0) \end{aligned} \tag{17}$$

where $f^* : (\delta v_0, \delta\gamma_0) \rightarrow \delta t_j$ is a general nonlinear function that must be approximated from discrete sampling of the 2D space of variations in jettison time t_j with variations in the dynamically dominant initial conditions v_0 and γ_0. The approximation of this function from discrete sampling could be accomplished by various 2D interpolation methods such as bicubic interpolation or splines. This would need to be done for each pre-computed nominal trajectory, taking the place of the linear sensitivities. Note that there is significant freedom in how this could be set up, and the choice of sampling and interpolating strategy would need to be explored further.

In general, nonlinear correction is needed when the back-propagation step produces quasi-initial conditions that differ too greatly from the nominal initial conditions for a linear correction to be valid. One interesting question is whether or not increasing the total number of pre-computed nominal trajectories can reduce the number of instances where a nonlinear corrective step is needed. It is also worth noting that if the set of initial trajectories is produced from smooth variations in the density profile, then the resulting nominal jettison time will also vary smoothly. This can be used to more rapidly pre-compute the nominal trajectories due to the good quality of the initial guess. Additional improvements to the prototype algorithm explored in this paper are likely possible.

It is additionally worth noting that while the approach explored in this paper is to generate nominal trajectories only by varying the scale factor on the atmospheric density profile, other functional variations to the density profile could also be explored. There is no need for the set of pre-computed nominal trajectories to vary in only one manner, and multi-dimensional variations could be applied.

RESULTS WITH PROTOTYPE ALGORITHM

For early tests, the new jettison time algorithm was applied to the nominal run and several of the dispersion runs in the DSENDS Earth-test aerocapture simulation. The results are summarized in the following sub-sections.

Nominal Run Test with Initial State Error Robustness Demonstration

From the nominal aerocapture case in the DSENDS simulation, the atmospheric density profile and initial state were used to generate 23 nominal trajectories for equally spaced density scale factors $F \in [0.5, 1.6]$, with increments of 0.05. The resulting semimajor axis profiles are plotted in Figure 6. Note that the resulting jettison times vary from 670 to 800 seconds. The larger the value

of F, the earlier the jettison time. For each nominal trajectory, the jettison time sensitivities are also computed. Note in Figure 6 that all trajectories achieve nearly the same post-aerocapture semima-

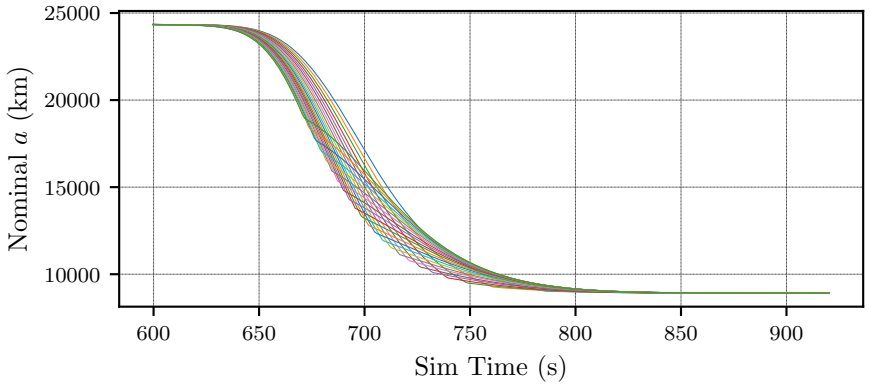


Figure 6: Pre-computed Trajectories for Nominal Aerocapture Test Case

jor axis because the targeting tolerance of apopapsis radius is only 15 km and there is very little variation in periapsis radius amongst the runs.

As an initial test of the new algorithm, the DSENDS nominal case is tested, and its initial conditions are used to generate all of the pre-computed trajectories. In addition, 9 dispersion runs are also tested using the new algorithm without recomputing the 23 nominal trajectories. Thus, for each dispersion run, the spacecraft initial conditions differ from the initial conditions used to compute the nominal map. As a baseline, for each run, the NPC algorithm-based jettison timer is also tested using the same dispersions and initial conditions. By running identical scenarios with the two different jettison timer methods, this small batch test thus isolates performance differences due only to differences in the algorithm implementations.

This initial test is designed to accomplish two goals. First and most importantly, with the nominal DSENDS case, the aim is to determine whether or not the QIC guidance algorithm *works* (i.e. whether or not it achieves comparable performance to the NPC algorithm). Additionally, with the dispersion cases, it is desired to obtain early results suggestive of whether or not the algorithm is *robust* – do errors between the true initial state and the initial state used to compute the nominal trajectories destroy the performance?

The results are summarized in Table 1. The QIC guidance algorithm achieves comparable but slightly worse error than the NPC algorithm for the the nominal DSENDS case (run 0). The additional error is not indicative of a systemic performance problem, as indicated by the off-nominal results in runs 1 through 9. Note that for these runs, the new algorithm doesn't even re-compute a new set of nominal trajectories based on the updated run-specific initial conditions. Despite this limitation, the new algorithm achieves an average final apoapsis error of 434.29 km, only slightly worse than the fully functioning NPC algorithm's average final apoapsis error of 386.91 km.

Note that of the 9 dispersion cases tested, in 6 of these cases, the new algorithm computed a linear correction to the jettison time that was outside of the linear regime of the sensitivities. In those cases (for which the jettison time is labeled with a star), the new algorithm simply defaulted to the nearest nominal trajectory jettison time, discarding the erroneous linear correction. Recall that this was a proposed robustness feature outlined in the end of Section 3.2. Despite this, it is highly encouraging that the simple default jettison time picking logic was still sufficient to successfully

Table 1: Nominal and Off-Nominal Performance Results, Nominal Map Only

Run	t_j, NPC	t_j, New Alg.	r_a error (km), NPC	r_a error (km), New
0	**704.91**	**705.24**	**−15.09**	**−82.00**
1	683.61	681.24	837.25	1282.32
2	719.01	721.0*	-25.37	-376.36
3	721.31	721.0*	-21.09	19.93
4	726.81	726.3*	-158.10	-88.78
5	662.94	676.6	1062.80	-835.78
6	695.51	695.6*	-731.73	-767.90
7	717.41	716.6*	361.56	511.19
8	718.71	716.14	-378.12	43.77
9	716.27	716.6*	-278.01	-334.91
Avg	–	–	386.91	434.29

produce acceptable jettison times, such that the average error using the (partially implemented) new algorithm was still comparable to the average error using the NPC algorithm.

Small-Batch Run Comparisons of Jettison Time Algorithm Performance

In this sub-section, several of the dispersion cases from the previous sub-section are re-run using properly generated pre-computed trajectory maps for the new algorithm, and the results are again compared to the NPC algorithm. The generation of these pre-computed maps is computed using prototype code not yet implemented in DSENDS, and is not completely automated, hence the small number of results here. Note that for the dispersion case results in Table 2, in all 8 dispersion cases

Table 2: Off-Nominal Performance Results, Updated Maps

Run	t_j, NPC	t_j, New Alg.	r_a error (km), NPC	r_a error (km), New
0	704.91	705.24	-15.09	-82.00
1	683.61	687.9*	837.25	-0.39
2	719.01	717.8*	-25.37	193.90
3	721.31	723.5*	-21.09	-319.00
4	726.81	726.2*	-158.10	-73.15
5	662.94	677.2*	1062.80	-922.82
6	695.51	691.1*	-731.73	170.84
7	717.41	717.5*	361.56	348.92
8	718.71	717.3*	-378.12	-145.23
9	716.27	–	-278.01	–
Avg	–	–	386.91	250.70

the linear correction to the jettison time was outside of the linear regime of the sensitivities (indicated by the asterisks on the quantities in column 3), so the algorithm defaulted to the jettison time of the nearest reference trajectory. Despite this, these results show that the new algorithm improves on the performance of the NPC algorithm in 6 of 8 dispersion cases, and the overall average apoapsis error is reduced by about 35%. This is a highly encouraging result. Future work will explore whether

increasing the number of pre-computed nominal trajectories fixes the tendency of the QICs to be outside the linear regime. Nonlinear correction will also be explored. In this work, the tendency of the linear correction to not be used prompted a stripped-down version of the algorithm to be tested, in which back-propagation and linear correction is not performed. This is the subject of the next section.

MONTE CARLO RESULTS WITH SIMPLIFIED ALGORITHM

A 5001-trial Monte Carlo analysis was performed using the DSENDS simulation of the Earth flight-test of drag-modulated aerocapture from a geosynchronous transfer orbit into lower Earth orbit,[7] targeting an apoapsis altitude of 5000 km. The simulation includes a variety of dispersions on the atmosphere, trajectory, and vehicle properties. Atmospheric density is modeled using the Earth Global Reference Atmospheric Model (Earth-GRAM) 2010.[11] The entry state is dispersed according to a navigation assessment performed at JPL that was then scaled to match the project requirement of entry flight path angle delivery error with a standard deviation value of $3\sigma = 0.2\,\mathrm{deg}$. A model of navigation uncertainty was also included.

The version of the algorithm tested in this analysis forgoes computing any sensitivities, and instead simply selects the jettison time of the nearest nominal trajectory. In this case the nearest trajectory is defined as the pre-computed nominal trajectory with the smallest distance parameter at the current time, with the distance parameter computed according to Eq. 12 with $c_1 = c_2 = 1$. Once selecting the new jettison time, the algorithm was allowed to update the current jettison time solution by a maximum of 5 seconds in either direction.

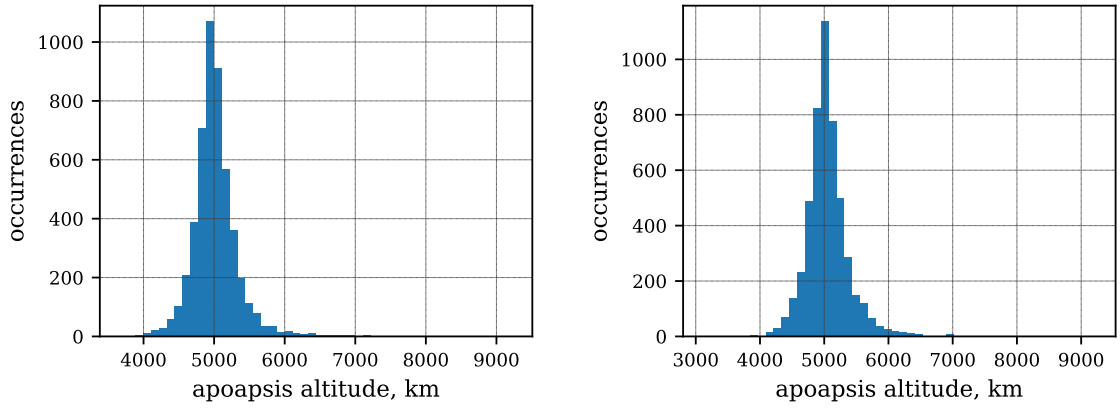
A large number of nominal trajectories were computed in order to achieve reasonable accuracy without any linear correction step. 85 trajectories were computed starting from the navigated state at simulation initialization approximately 10 minutes prior to entry, with density scale factors $F \in [0.58, 1.42]$ with increments of 0.01. For comparison, NPC guidance as described in Section was applied to the same 5001 cases, using a combination of bisection and Newton's method for root-finding.

The results of this Monte Carlo analysis are summarized in Table 3 and Figure 7. Clearly, even with the simplified implementation the QIC guidance has very similar performance to the NPC guidance for this scenario.

Table 3: Achieved apoapsis altitude, for 5001-trial Monte Carlo analysis, with 5000 km target. All values in km.

Alg.	Mean	Standard Deviation	1st-Percentile	99th-Percentile
QIC	5018.77	342.70	4257.60	6263.48
NPC	5052.46	358.01	4295.14	6263.48

Some of the outliers in both datasets occur because the dispersions applied in this scenario are such that, for a small percentage of the cases, the target apoapsis is unreachable. In these cases, the best the guidance algorithm can do is to command jettison either immediately after atmospheric entry (for steep, high-density cases) or just before atmospheric exit (for shallow, low-density cases). That said, for both algorithms the apoapsis data is well-centered with a reasonable standard deviation, considering that the architectural choice of single-event jettison drag-modulation inherently



(a) QIC guidance, simplified

(b) NPC guidance

Figure 7: Achieved apoapsis altitude histograms

limits control authority in the atmosphere. Neither guidance algorithm results in any failure (impact) cases out of the 5001 trials.

A quantitative comparison of computational effort between the NPC and QIC approaches is out of scope for this work. However, the absence of iterations or numerical propagation from the atmospheric flight phase of the simplified QIC algorithm makes it clear that the simplified QIC method requires less computation effort than the NPC approach. The primary computational burden of the simplified QIC approach is in the pre-compute step that occurs in the 10 minutes before atmospheric entry. For QIC guidance with the back-propagation and jettison time correction, there is always only one propagation per call, which is generally less than is needed for the NPC algorithm, and can also be computationally bounded.

CONCLUSIONS

This paper introduces a novel technique for discrete-event drag modulation for aerocapture guidance. Previously, algorithms based on a numerical predictor-corrector (NPC) approach have been primarily explored for the aerocapture guidance problem. The NPC approach is well-studied and easy to implement, but it is somewhat susceptible to navigation error. Additionally, the NPC algorithm is not computationally well-posed – it is difficult to estimate or bound the number of iterations required for convergence in a given call to the algorithm. The new QIC guidance approach discussed in this work is expected to be more robust to navigation error, and is additionally computationally well-posed. The approach relies on estimating the state at some time before entry, generating nominal capture trajectories from a range of atmospheric density profiles, and additionally computing the sensitivity of each associated trigger time to initial height, velocity, and flight path angle. The navigation-estimated state during descent is used to identify the nearest nominal trajectory, and the dynamic model for that trajectory is used to back-propagate the estimated state to a new perturbed entry condition, which informs a correction to the estimated jettison time, which can be linear or nonlinear.

In this work, early prototypes of the QIC guidance algorithm are implemented in a high-fidelity simulation environment that was written using the Dynamics Simulator for Entry, Descent, and Surface Landing (DSENDS) software developed by the DARTS lab at NASA's Jet Propulsion Labora-

tory (JPL).[6] The highly realistic dynamic modeling enables the performance of the new algorithm to be compared to an NPC approach in a flight-like simulation of an aerocapture demonstrator mission. The preliminary results show that the new algorithm is highly robust to errors in the spacecraft state before entry, and that the new implementation shows promise as a guidance method for discrete-event aerocapture. A large Monte Carlo study of a simplified version of the algorithm without back-propagation shows comparable performance to the previously implemented NPC algorithm. These early and highly promising results encourage further exploration of the QIC guidance algorithm for aerocapture applications.

ACKNOWLEDGEMENTS

The authors thank Aaron Schutte, Bill Strauss, and Dan Burkhart for supporting work performed at NASA JPL for this paper. This work was partially supported by a NASA Space Technology Research Fellowship. Additionally, the DSENDS environment in which these simulations were performed was developed and supported by Abhi Jain and others in the JPL DARTS Lab.

REFERENCES

[1] T. R. Spilker, M. Adler, N. Arora, P. M. Beauchamp, J. A. Cutts, M. M. Munk, R. W. Powell, R. D. Braun, and P. F. Wercinski, "Qualitative Assessment of Aerocapture and Applications to Future Missions," *Journal of Spacecraft and Rockets*, Vol. 56, No. 2, 2019, pp. 536–545, 10.2514/1.A34056.

[2] N. Vinh, W. Johnson, and J. Longuski, *Mars aerocapture using bank modulation*. 2000, 10.2514/6.2000-4424.

[3] Z. R. Putnam and R. D. Braun, "Drag-Modulation Flight-Control System Options for Planetary Aerocapture," *Journal of Spacecraft and Rockets*, Vol. 51, No. 1, 2014, pp. 139–150, 10.2514/1.A32589.

[4] D. M. Fawley and Z. R. Putnam, *Comparison of Real-time Guidance Options for Drag-Modulation Entry at Mars*. American Institute of Aeronautics and Astronautics, 2020/04/27 2020, doi:10.2514/6.2020-0848.

[5] M. J. Grace, E. R. Burnett, and J. W. McMahon, "Quasi-Initial Conditions as a State Representation for Aerocapture," *AIAA SciTech Forum*, 2022.

[6] J. M. Cameron, A. Jain, P. D. Burkhart, E. S. Bailey, B. Balaram, E. Bonfiglio, M. Ivanov, J. Benito, E. Sklyanskiy, and W. Strauss, *DSENDS: Multi-mission Flight Dynamics Simulator for NASA Missions*, 10.2514/6.2016-5421.

[7] A. Cassell, B. Smith, P. Wercinski, S. Ghassemieh, K. Hibbard, A. Nelessen, and J. Cutts, "ADEPT, A Mechanically Deployable Re-Entry Vehicle System, Enabling Interplanetary CubeSat and Small Satellite Missions," *Proceedings of the Small Satellite Conference, Session 3: Advanced Technologies I*, 2018.

[8] A. Austin, A. Nelessen, B. Strauss, J. Ravich, M. Jesick, E. Venkatapathy, R. Beck, P. Wercinski, M. Aftosmis, M. Wilder, G. Allen, R. Braun, M. Werner, and E. Roelke, "SmallSat Aerocapture to Enable a New Paradigm of Planetary Missions," *IEEE Aerospace Conference*, 2019.

[9] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*. Reston, VA: AIAA Education Series, 4th ed., 2018.

[10] E. Roelke and R. D. Braun, "Discrete-Event Drag-Modulated Guidance Performance for Venus Aerocapture," *Journal of Spacecraft and Rockets*, Vol. 58, No. 1, 2021, pp. 190–199, 10.2514/1.A34761.

[11] F. W. Leslie and C. G. Justus, "The NASA Marshall Space Flight Center Earth Global Reference Atmospheric Model - 2010 Version," Tech. Rep. NASA/TM-2011-216467, NASA, 2011.