

Guidance, navigation, and control solutions for spacecraft re-entry point targeting using aerodynamic drag

Sanny Omar, Riccardo Bevilacqua*

University of Florida, ADAMUS Laboratory, Gainesville, FL, 32611, USA



ARTICLE INFO

Keywords:

Spacecraft
Drag
Guidance
Tracking
Re-entry point targeting

ABSTRACT

As large numbers of increasingly smaller spacecraft continue to be launched, means of efficient and reliable orbital maneuvering and orbit disposal have become increasingly necessary. For spacecraft that do not contain thrusters, aerodynamic drag modulation using a retractable drag device or attitude changes presents itself as an efficient way to perform orbital maneuvers and control the re-entry location.

This paper introduces an aerodynamically based re-entry guidance generation algorithm for low Earth orbit spacecraft that exhibits significant accuracy, robustness, and efficiency. The paper also presents a novel guidance tracking algorithm whereby the drag device of a spacecraft is deployed or retracted relative to a nominal deployment profile (given in the guidance) based on the difference between the actual and desired state of the spacecraft. A full state feedback linear-quadratic-regulator control scheme is utilized with the Schweighart Sedgwick equations of relative motion to drive the relative position and velocity between the spacecraft and the guidance trajectory to zero. A problem-specific Extended Kalman Filter implementation is also introduced to remove noise from the GPS-derived relative motion estimate.

One thousand Monte Carlo simulations of the guidance generation algorithm with randomized initial conditions and desired re-entry locations are conducted, resulting in an average guidance error of 12.5 km and a maximum error below 106 km. The tracking of these aerodynamic decay guidances with the aforementioned algorithms is also simulated with drag force uncertainties up to a factor of two and navigation errors (noise and bias) comparable to that expected from a CubeSat GPS unit. Despite these simulated errors and uncertainties, this approach provides guidance tracking down to a re-entry altitude of 120 km with a final position error under 6 km for all cases. The algorithms detailed in this paper provide a way for any spacecraft capable of modulating its drag area to autonomously perform orbital maneuvers and execute a precise re-entry.

1. Introduction

Spacecraft orbit and re-entry control is traditionally conducted using powerful chemical engines capable of producing a nearly instantaneous change in velocity [1]. The advent of small spacecraft such as CubeSats [2] with minimal or no propulsion systems has fueled the development of creative orbit control methods including the use of aerodynamic drag. The concept of orbit control using aerodynamic drag has been considered for decades [3] and a number of researchers including the authors of this paper have worked on this problem [4–10]. Recently, Planet Labs was able to control a constellation of over 100 CubeSats using aerodynamic drag [11]. However, many of these aerodynamic orbit control algorithms are designed for bang-bang control (min or max drag only), only work with small initial spacecraft

separations, do not employ feedback control to correct for uncertainties, or result in long maneuver completion times. Given the increasing number of spacecraft in LEO, there is a concern about orbital debris mitigation, especially since most small satellites cannot perform propulsive de-orbit burns. Several teams have developed drag devices that increase the cross-sectional area of a satellite to expedite de-orbit [12–14], but these devices do not control the re-entry location of the host satellite.

Satellites containing components such as tungsten or titanium that may survive re-entry and pose a hazard to ground assets [1] need a way to control their de-orbit location in order to obtain a launch [15,16]. NASA debris mitigation guidelines [15] require that the casualty risk from re-entering debris must be less than 1 in 10,000 and the Inter-Agency Space Debris Coordination Committee (IADC)¹ has stated that

* Corresponding author.

E-mail address: bevivr@ufl.edu (R. Bevilacqua).

¹ www.iadc-online.org.

Nomenclature

a	Semi-major axis (km)	\mathbf{r}	Spacecraft position vector (km)
C_b	Ballistic coefficient (m^2/kg)	SS	Schweighart Sedwick relative motion model
e	Eccentricity	s_d	Ballistic coefficient scaling factor
ECI	Earth centered inertial	T	Period of orbit or sinusoidal noise term (s)
EKF	Extended Kalman Filter	t	Time (s)
i	Inclination ($radians$)	t_c	Time controllability of analytical solution (s)
J_2	Constant that describes Earth's oblateness	u	Argument of latitude ($radians$)
\mathbf{K}	LQR gain matrix	\mathbf{v}	Spacecraft velocity vector (km/s)
LEO	Low Earth Orbit	W_d	Work done by aerodynamic drag per unit mass (km^2/s^2)
LQR	Linear quadratic regulator	θ	True anomaly ($radians$)
n	Mean motion ($radians/s$)	μ	Earth's gravitational parameter (km^3/s^2)
\mathbf{Q}	LQR weighting matrix for state error	ρ	Density (kg/m^3)
\mathbf{R}	LQR weighting matrix for actuator effort	Ω	Right ascension of ascending node ($radians$)
R_e	Radius of Earth	ω	Argument of perigee ($radians$)
		ω_e	Earth rotation rate ($radians/s$)

debris surviving re-entry must not cause an undue risk to persons or property. Debris risk mitigation requirements are only expected to become stricter and it will become increasingly necessary for satellite operators to implement controlled re-entry capabilities. If a satellite cannot contain a propulsion system due to volume, mass, or power constraints, modulation of the aerodynamic drag experienced by the satellite can be utilized to control the de-orbit location. This drag modulation could be achieved using a retractable drag device [17,18] or by changes in the spacecraft's cross-sectional area. Prior works by Virgili [19] and Dutta [20] discuss algorithms for aerodynamically-based re-entry control, but these algorithms have some limitations (further elaborated on in Ref. [21]). Virgili's algorithm provides only an initial guess of a satellite ballistic coefficient profile that must be followed to de-orbit in the desired location. This initial guess must be used in a numerical optimizer that is computationally intensive and has no convergence guarantees. Dutta proposes directly using NASA's Program to Optimize Simulated Trajectories (POST2) numerical optimizer to calculate the desired ballistic coefficient profile. These algorithms are not suitable to run onboard a spacecraft due to the lack of convergence guarantees and Virgili and Dutta have not fully investigated the ability of a spacecraft with a retractable drag device to track the generated guidances in a realistic environment with model uncertainty and sensor noise.

The guidance generation algorithm introduced in this paper is based on the authors' previous work [21,22] but offers substantial novelties and improvements. The shortcomings of the previous algorithm and the new improvements are discussed in Sections 2 and 3. This work also introduces a high performance LQR (Linear Quadratic Regulator) [23,24] based guidance tracking algorithm (Section 4) that enables a spacecraft capable of active drag modulation to follow the guidance despite sensor noise and drag force uncertainties. While many feedback control algorithms in prior literature were designed for bang-bang control (drag device fully deployed or fully retracted), the presented tracking algorithm allows intermediate deployment levels, resulting in significant power savings. These bang-bang approaches could not be generalized for continuous control so a fundamentally different control architecture was required for this algorithm. The tracking algorithm utilizes the in-plane relative position and velocity (four states) through the Schweighart Sedgwick relative motion dynamics [25], resulting in improved performance over prior algorithms which only account for two states (generally mean anomaly and semi major axis) and do not consider J_2 perturbations in the dynamics. A means of analytically calculating the \mathbf{Q} and \mathbf{R} matrices for the LQR controller based on the desired system performance (a topic often neglected) is also presented. Section 5 presents a unique version of an EKF (Extended Kalman Filter) [26] to filter GPS measurement noise. Instead of filtering on the inertial satellite position and velocity, the EKF filters directly on the position

and velocity relative to the guidance. This allows the error covariance to become smaller in the radial direction, leading to more accurate state estimates. This formulation also helps to smooth out noise-like errors that result from the guidance being an imperfect representation of reality. By keeping guidance tracking in mind when designing the EKF, a more accurate state estimate and superior controller performance can be obtained than if the noise filter and the controller were developed in a completely decoupled manner. Finally, Section 6 discusses the results of simulations to validate various aspects of the tracking controller performance including the cases of actuator saturation, sensor noise, bias errors, model uncertainties, and actuation delays. A Monte Carlo campaign consisting of 1000 guidance generation and guidance tracking simulations was also conducted with randomized initial conditions and realistic models of sensor noise and drag estimation errors. The results of these simulations are shown in Section 6.4 and provide a complete validation of the guidance, navigation, and control algorithms needed for spacecraft de-orbit point targeting using aerodynamic drag. Such a comprehensive development and detailed validation of all GNC algorithms relevant to this problem is not available in prior literature.

2. Previous guidance generation algorithm

2.1. Previous algorithm overview

The algorithm discussed in this paper is based on the guidance generation procedure discussed in the authors' prior work [21,22] but offers significant improvements. The algorithm in Ref. [21] calculates an initial ballistic coefficient (C_{b1}), second ballistic coefficient (C_{b2}), and time value t_{swap} . The spacecraft maintains C_{b1} until time t_{swap} , C_{b2} until a specified orbit semi major axis a_{term} is reached, and a predefined ballistic coefficient C_{bterm} until the de-orbit altitude. If the proper control parameters (C_{b1} , C_{b2} , and t_{swap}) are chosen, the spacecraft will arrive at the desired latitude and longitude at the de-orbit altitude. Note that in Refs. [21] and [22] and in this paper, C_b is defined as

$$C_b = \frac{C_d A}{2m} \quad (1)$$

where C_d is the drag coefficient, A is a reference area (often the cross-wind surface area), and m is the mass of the spacecraft. In Ref. [22], an analytical solution is developed where, if given a numerically propagated spacecraft trajectory with some C_b profile, the de-orbit location of a spacecraft with the same initial conditions but a different C_b profile can be calculated. Using this analytical solution and an initial numerically propagated trajectory with some control parameters C_{b1} , C_{b2} , and t_{swap} , a means of calculating the C_b adjustments needed for proper targeting is developed. In this method, the t_{swap} value is first adjusted to achieve the increase in true anomaly necessary for latitude targeting

with the minimum possible longitude error. Next all three control parameters are adjusted using an analytical solution to yield proper longitude targeting. The trajectory is propagated with these new control parameters, and the control parameters are once again adjusted via the analytical solution based on the de-orbit location of the new simulated trajectory. This process continues as shown in Fig. 1 until a maximum number of iterations is reached or a numerically propagated trajectory is achieved with targeting error below a specified threshold.

While the prior guidance generation algorithm is extremely effective in many scenarios, there are some limitations. For one, there must be some difference between C_{b1} and C_{b2} for changes in t_{swap} to induce any change in orbital behavior. In the beginning of the simulation, C_{b1} and C_{b2} are set as far apart as possible to maximize the effectiveness of changes in t_{swap} , but over the course of several targeting iterations, it is possible to have C_{b1} and C_{b2} very close together, making latitude targeting impossible. This issue also hinders the ability to re-generate a new guidance with the initial guidance control parameters as an initial guess. If the spacecraft is following an initial guidance and is beyond the swap point (t_{swap}), the remainder of the guidance until the terminal point can be characterized by $C_{b1} = C_{b2}$ and an arbitrary t_{swap} . This means latitude targeting via t_{swap} variations will not be possible. Guidance re-generation must begin from scratch with C_{b1} and C_{b2} as far apart as possible which increases computation time and provides no convergence guarantee.

Another limitation of the prior algorithm is due to the trade-off between controllability and sensitivity. If targeting begins with insufficient time remaining before the de-orbit, it may not be possible to find a C_b profile that yields a desired de-orbit point (insufficient controllability). However, if maneuvering begins too early, the discrepancies between the analytical and numerical solutions will become large because the errors introduced by the assumptions made in the analytical solution will grow over time. If the remaining orbit lifetime is above a certain threshold, the analytical and numerical solutions will become so inconsistent with each other that the algorithm will not converge. In Ref. [21], guidance generation simulations using a high fidelity orbit model were set to begin when the spacecraft had roughly one week of orbit lifetime remaining. This allowed the algorithm to converge, but resulted in longitude errors of up to 1250 km due to the limited controllability.

3. Improved Guidance Generation Algorithm

3.1. Algorithm general form

The new guidance generation algorithm in this paper address all the shortcomings discussed in Sec. 2 using an improved analytical solution and a shrinking horizon strategy. As in the prior algorithm, the analytical solution calculates the control parameters (C_{b1} , C_{b2} , t_{swap}) needed for de-orbit point targeting. However, both latitude and longitude targeting are handled in the same calculation and the only restriction on the initial C_b values is that they must lie between the minimum and maximum allowable satellite ballistic coefficients ($C_{b_{min}}$ and $C_{b_{max}}$). This eliminates the issue of insufficient latitude controllability discussed previously and allows for a more complete exploration of the available control space, yielding more accurate solutions. The shrinking horizon

strategy capitalizes on the benefits of high controllability at high initial altitudes and the reduced sensitivity to drag force perturbations and errors in the analytical solution assumptions at low altitudes. The use of mean argument of latitude instead of true anomaly in the analytical solution as well as the implementation of the drag-work enforcement method help reduce the discrepancies between the analytical and numerical solutions and lead to an improved accuracy and higher convergence rates.

The general form of the shrinking horizon strategy is as follows. A trajectory is first propagated with a set of control parameters analytically calculated for proper targeting. Because of the high sensitivity, it is unlikely that the de-orbit point in the numerically propagated trajectory will correspond to the one predicted by the analytical solution. A predefined percentage of the beginning of the newly propagated trajectory (time t_g) is saved as the initial part of the guidance. The rest of the trajectory is then utilized to analytically calculate a new set of control parameters. Another trajectory is propagated with these parameters using the state of the prior trajectory at t_g as an initial condition and a predefined percentage of that trajectory is appended to the first part of the guidance. The process continues until a trajectory is propagated that has less than a specified amount of orbit lifetime remaining or lands within a specified distance of the target point. That entire trajectory is then appended to the previously calculated initial components of the guidance. At this point, the guidance generation algorithm is complete and a reference trajectory corresponding to a desired C_b profile has been created. The logic behind this algorithm is depicted graphically in Fig. 2. Fig. 3 illustrates the case of a general shrinking horizon problem where each computation of the problem solution (in this case the C_b profile) represents a steadily decreasing time between the initial condition and the solution horizon (in this case the de-orbit point). Fig. 4 illustrates how each numerical simulation begins at a lower semi major axis than the previous one, hence shrinking the horizon (remaining semi major axis) with each propagation. In Fig. 4, each thin, colored line represents a trajectory simulated all the way to the de-orbit point with some drag profile. The line sections with the thick black outlines represent the portions of the trajectories that are included in the guidance and to which the drag-work enforcement method is applied. In other words, the final guidance consists of the trajectory defined by the thick black outlines.

The drag-work enforcement method involves iteratively modifying the C_b during the first t_g seconds of the propagation of each trajectory to ensure that the total work done by aerodynamic drag at t_g is equal to the work that should be done by this point according to the analytical solution. Recall that the portion of the trajectory before time t_g will be added to the guidance and is represented by the thick black outlines in Fig. 4. Because the analytical and numerical solution experience the same total work done by drag at t_g , the discrepancy between them is reduced and the accuracy and convergence rate of the algorithm are significantly increased.

3.2. Analytical solution for control parameters

The goal of the analytical solution is to estimate the ballistic coefficient profile necessary to de-orbit in a desired location. The solution is developed by first deriving a mapping between the changes in the C_b

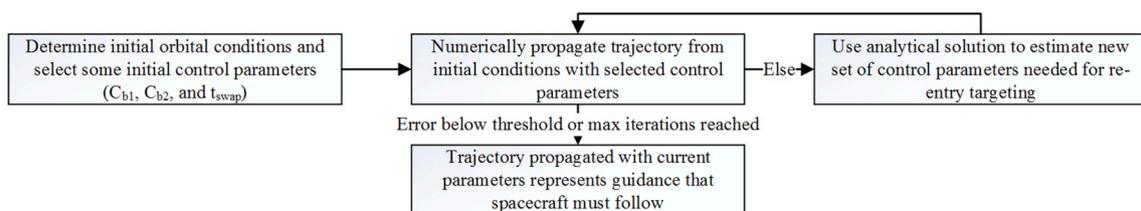


Fig. 1. Prior Targeting Algorithm Schematic [21].

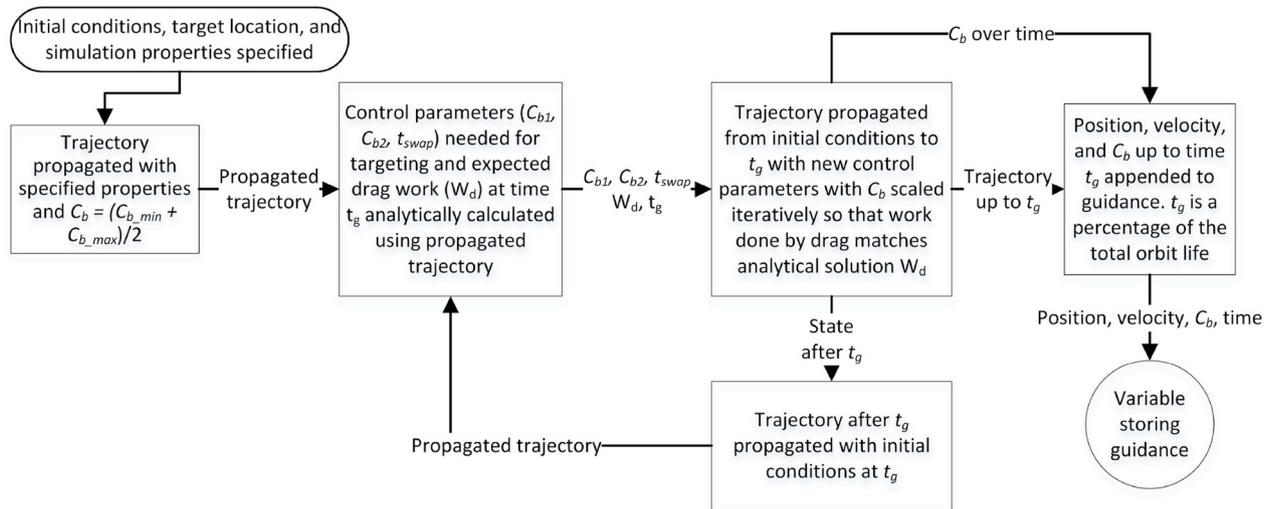


Fig. 2. Basic form of the improved guidance generation algorithm.

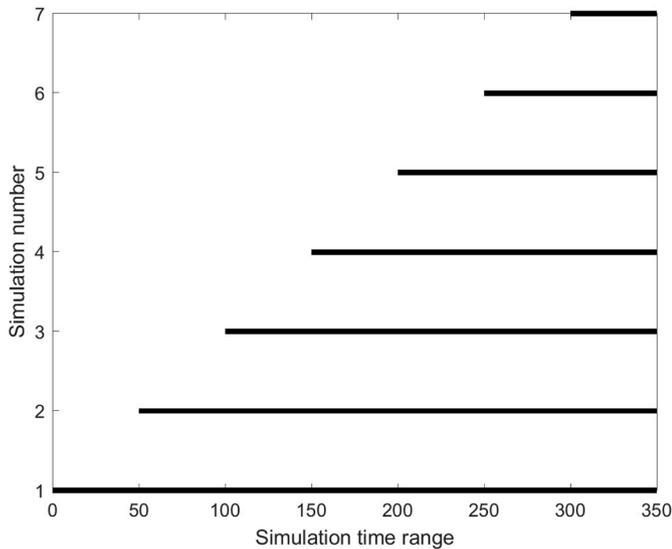


Fig. 3. Shrinking horizon control general form.

profile and the changes in the final orbital elements of a trajectory. This mapping is then inverted to compute the C_b profile changes necessary to induce the changes in the final orbital elements necessary to de-orbit in the desired location.

3.2.1. Mapping from initial conditions and drag profile to de-orbit location

The analytical re-entry point targeting solution is based on the relation [22] that the time and argument of latitude required for a spacecraft to decay from an initial to final semi major axis due to aerodynamic drag increase linearly with decreasing ballistic coefficient. As an example, assume satellite “A” with ballistic coefficient C_{bA} takes time Δt_A to achieve some change in semi major axis Δa and undergoes an argument of latitude change Δu_A during this drop. The time and argument of latitude change a satellite “B” with the same initial conditions and some different C_{bB} will undergo to achieve the same Δa is given by

$$\Delta t_B = \frac{C_{bA} \Delta t_A}{C_{bB}} \tag{2}$$

$$\Delta u_B = \frac{C_{bA} \Delta u_A}{C_{bB}} \tag{3}$$

Though this is completely true only if the satellites are in a circular orbit around a spherical Earth and density is a function of only semi-

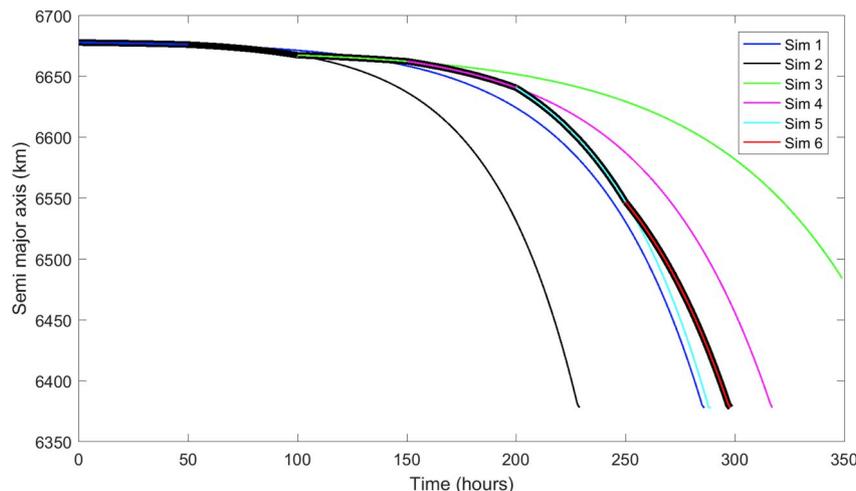


Fig. 4. Semi major axis over time for all trajectories in shrinking horizon guidance generation example.

major axis, it is a reasonable approximation for analytically characterizing the effect that a C_b change will have on an orbit. Note that even though Δu is an angle and can be defined between 0 and 2π , the cumulative value of Δu since the initial state must be used in Eq. (3). For example, if satellite “A” has made 5 complete orbits since the initial conditions, then $\Delta u_A = 10\pi$, not 0. One may notice that in Ref. [22], true anomaly is used instead of argument of latitude in Eq. (3). Though the theoretical results are the same, using argument of latitude provides some implementation benefits. First, the argument of the perigee is not well defined for circular orbits and in a high fidelity numerical simulation, will oscillate wildly due to J_2 perturbations. This makes it difficult to accurately compute the true anomaly. Because argument of latitude is the sum of true anomaly and argument of perigee however, it is well defined even in circular orbits and does not exhibit erratic behavior in a high fidelity simulation environment. Additionally, the mean argument of latitude, defined as the sum of the mean anomaly and argument of the perigee with the J_2 -induced short periodic oscillations subtracted out, can be utilized when processing the results of numerical simulations. This ultimately leads to more accurate analytical solutions, justifying the use of argument of latitude instead of true anomaly in this paper.

Eq. (2) and Eq. (3) are proven formally in Ref. [22] using the Gauss Variation of Parameters for a circular orbit around a spherical Earth, but they can also be understood intuitively using the work energy theorem. Because the only force that does work on the satellite in a circular, two-body orbit is drag, the total change in orbital energy over a period of time is equal to the work done by drag. Because the drag force is directly proportional to the ballistic coefficient, a factor x increase in C_b will result in a factor x increase in the rate of energy change. Because orbital energy can be defined in terms of semi major axis ($E = -\frac{\mu}{2a}$), this results in a factor x reduction in the time required to achieve a desired semi major axis change. Similarly, because the rate of change of argument of latitude in a circular orbit is equal to mean motion which is directly related to the semi major axis ($n = \sqrt{\frac{\mu}{a^3}}$), a factor of x reduction in the time spent decaying from some a_i to a_f will lead to a factor x reduction in the total change in argument of latitude experienced during that time.

Since the average rate of change of right ascension ($\dot{\Omega}_{avg}$) is independent of C_b , the change in Ω experienced during the orbital decay can be calculated by

$$\Delta\Omega = \dot{\Omega}_{avg}\Delta t \tag{4}$$

Inclination is assumed not to change significantly during the orbital decay, and eccentricity and argument of the perigee are both zero in a circular orbit.

As shown in Fig. 5, if the trajectory of a satellite with some initial set of control parameters has been numerically propagated, the de-orbit location of a new trajectory corresponding to the same initial conditions but a different set of control parameters can be analytically estimated by dividing the trajectories into regions of semi major axes where the C_b is not changing in either trajectory. Note that in Fig. 5, the solid line represents an initial numerically propagated trajectory and the dashed line represents a new trajectory whose de-orbit location we would like to calculate without numerically propagating the trajectory. Define phase i as the region between some initial and final semi major axis where the satellite in the numerically propagated trajectory maintains a fixed ballistic coefficient C_{bi0} . Define the changes in time, argument of latitude, and right ascension during region i of this trajectory as Δt_i , Δu_i , and $\Delta\Omega_i$. Eqs. (2)–(4) can be utilized to solve for the Δt_i , Δu_i , and $\Delta\Omega_i$ that would have occurred in phase i had the ballistic coefficient instead been C_{bi} as

$$\Delta t_i = \frac{C_{bi0}\Delta t_{i0}}{C_{bi}} \tag{5}$$

$$\Delta u_i = \frac{C_{bi0}\Delta u_{i0}}{C_{bi}} \tag{6}$$

$$\Delta\Omega_i = \left(\frac{\Delta\Omega_{i0}}{\Delta t_{i0}}\right)\Delta t_i \tag{7}$$

In Fig. 5, phase 1 occurs between t_0 and $t_{s_{old}}$ in the initial numerically propagated trajectory, phase 2 occurs between $t_{s_{old}}$ and $t_{eq_{old}}$, phase 3 occurs between $t_{eq_{old}}$ and $t_{term_{old}}$, and phase 4 occurs between $t_{term_{old}}$ and the de-orbit point. Note that in each phase, the ballistic coefficient is unchanging in both the new and initial trajectories, so Eqs. (5)–(7) can be used to find the changes in time, Ω , and u in each phase of the new trajectory. If Δt_i for only the new trajectory is known during phase i , Eq. (5) can be solved for Δt_{i0} and this value can be used to find Δu_{i0} and $\Delta\Omega_{i0}$. Such a case occurs in phase 2 of Fig. 5 where Δt_{20} cannot be calculated explicitly because $t_{eq_{old}}$ is not known, but $\Delta t_2 = t_{s_{new}} - t_{eq_{new}}$ can be calculated once $\Delta t_1 = t_{eq_{new}}$ has been found using Eq. (5). Given a set of initial orbit elements at time t_0 , the time and orbital elements of the new trajectory at the de-orbit point can be found by

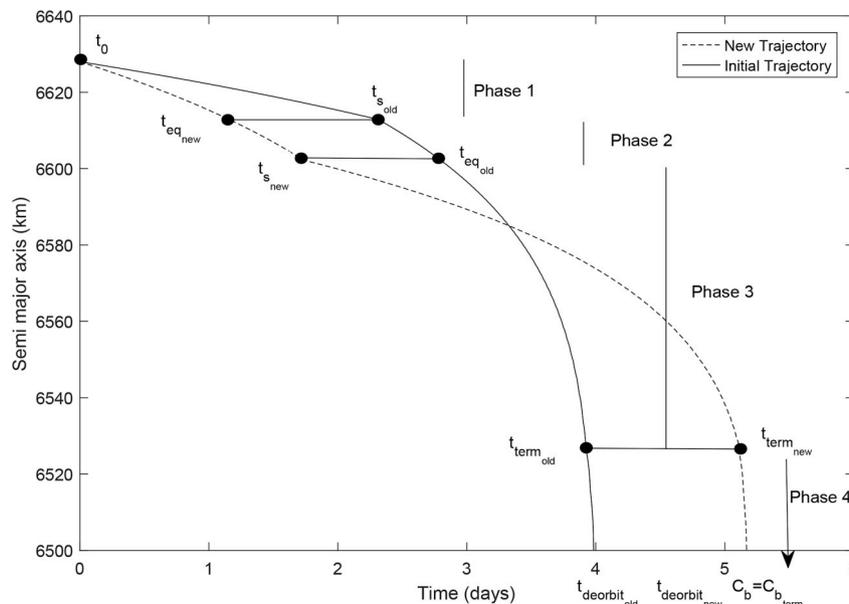


Fig. 5. Dividing orbits into phases for analytical de-orbit point calculation [21].

$$t_f = t_0 + \sum_{i=1}^4 \frac{C_{bi0} \Delta t_{i0}}{C_{bi}} \tag{8}$$

$$u_f = u_{init} + \sum_{i=1}^4 \frac{C_{bi0} \Delta u_{i0}}{C_{bi}} \tag{9}$$

$$\Omega_f = \Omega_{init} + \sum_{i=1}^4 \left(\frac{\Delta \Omega_{i0}}{\Delta t_{i0}} \right) \Delta t_i \tag{10}$$

$$\omega_f = e_f = 0 \tag{11}$$

$$i_f = i_{init} \tag{12}$$

The final semi-major axis of the new trajectory will be assumed identical to that of the numerically propagated trajectory because de-orbit occurs at a fixed altitude. These orbital elements can be converted to an ECI position and velocity and with knowledge of t_f , to a de-orbit latitude and longitude. Observe that during phase 4, both trajectories have the same C_b so they experience the same changes in orbital elements.

3.2.2. Computing the ballistic coefficient needed for Re-Entry point targeting

The analytical solution framework can be utilized to determine the ballistic coefficient changes necessary to achieve a desired change in the de-orbit location. The argument of latitude required to de-orbit at the desired latitude is calculated based on the transformation matrix between the perifocal and ECI frames given by eq. 4.49 in Ref. [27]. In the perifocal frame, the origin is at the orbit focus (center of Earth), the x -axis points toward the perigee, the z -axis is aligned with the orbit angular momentum vector, and the y -axis completes the right-handed coordinate system. The z -component of the ECI position vector is equal to the last row of the perifocal to ECI frame direction cosine matrix multiplied by the perifocal position vector.

$$r \sin(lat) = r \begin{bmatrix} \sin(\omega)\sin(i) & \cos(\omega)\sin(i) & \cos(i) \end{bmatrix} \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix} \tag{13}$$

Recognizing the trigonometric identity

$$\sin(u) = \sin(\omega + \theta) = (\sin(\omega)\cos(\theta) + \cos(\omega)\sin(\theta)) \tag{14}$$

Eq. (13) can be simplified to

$$u_1 = \sin^{-1} \left(\frac{\sin(lat)}{\sin(i)} \right) \tag{15}$$

Because the \sin^{-1} function only returns values between $-\pi/2$ and $\pi/2$, u_1 represents passage of the desired latitude on the ascending (northbound) portion of the orbit. The u value that yields latitude targeting on the descending (southbound) portion of the orbit can be calculated by

$$u_2 = \pi - u_1 \tag{16}$$

For either selected value of u , the total necessary change in argument of latitude from t_0 to the de-orbit point necessary for proper latitude targeting can be defined as

$$\Delta u_f = u + 2k\pi \tag{17}$$

where k is an integer. Selection of the proper value of k will be discussed later, but an initial k can be selected to yield u_f as close as possible to (the total argument of latitude change in the numerically propagated trajectory). This selection maximizes the accuracy of the analytical solution by minimizing the deviation from the propagated trajectory.

The longitude error that would result if Δu_f were achieved but the orbit lifetime remained the same can be calculated by determining the orbital elements of the numerically propagated trajectory at the de-orbit point, setting the argument of latitude to Δu_f , and calculating the

impact latitude and longitude using the original de-orbit time. For a given longitude error from this impact location and the desired impact location denoted by

$$\lambda_e = \lambda_{imp} - \lambda_{des} \tag{18}$$

the increase in orbit lifetime necessary to correct for this longitude error is calculated by

$$\Delta t_d = \frac{\lambda_e}{\omega_e - \dot{\Omega}_{avg}} \tag{19}$$

where ω_e is the rotation rate of Earth and $\dot{\Omega}_{avg}$ is the average rate of change of right ascension in the numerically propagated trajectory. Note that $(\omega_e - \dot{\Omega}_{avg})$ is the angular rate at which the Earth rotates relative to the orbital plane.

To solve for the C_b profile needed to achieve a desired orbit behavior, consider the case where t_{swap} occurs at the same semi-major axis in both the new and initial trajectories. In this case, phase 2 in Fig. 5 ceases to exist, and there are only two phases between t_0 and the terminal point (t_{term}). We will call phase 1 the region between t_0 and t_{swap} in both trajectories and phase 2 the region between t_{swap} and t_{term} . Furthermore, because both trajectories experience the same time (Δt_{4i}) and argument of latitude (Δu_{4i}) change during the terminal phase (phase 4 discussed earlier), the desired time (Δt_i) and argument of latitude (Δu_i) at the terminal point can be calculated as

$$\Delta u_i = \Delta u_f - \Delta u_{4i} = \Delta u_{i0} + \Delta u_d \tag{20}$$

$$\Delta t_i = \Delta t_{i0} + \Delta t_d \tag{21}$$

where Δt_{i0} and Δu_{i0} represent the time and change in argument of latitude to the terminal point in the initial numerically propagated trajectory. Δu_d is the increase in total change in argument of latitude needed for proper targeting. Eq. (20) and Eq. (21) allow us to modify only C_{b1} , C_{b2} , and t_{swap} to attempt to achieve the desired Δu_i and Δt_i . Δu_i and Δt_i are related to C_{b1} and C_{b2} by

$$\Delta u_i = \frac{C_{b10} \Delta u_{i0}}{C_{b1}} + \frac{C_{b20} \Delta u_{20}}{C_{b2}} \tag{22}$$

$$\Delta t_i = \frac{C_{b10} \Delta t_{i0}}{C_{b1}} + \frac{C_{b20} \Delta t_{20}}{C_{b2}} \tag{23}$$

Solving Eq. (22) for C_{b1} yields

$$C_{b1} = \frac{\Delta u_{i0} C_{b10} C_{b2}}{\Delta u_i C_{b2} - \Delta u_{20} C_{b20}} \tag{24}$$

Substituting Eq. (24) for C_{b1} in Eq. (23) and solving for C_{b2} yields

$$C_{b2} = \frac{C_{b20} (\Delta t_{20} \Delta u_{i0} - \Delta t_{i0} \Delta u_{20})}{\Delta t_i \Delta u_{i0} - \Delta t_{i0} \Delta u_i} \tag{25}$$

Finally, t_{swap} must be updated to enforce the condition that the swap point occurs at the same semi major axis in both trajectories. This is achieved by setting

$$t_{snew} = \frac{t_{sold} C_{b10}}{C_{b1}} \tag{26}$$

Note that variables with subscript “0” correspond to the initial numerically propagated trajectory.

3.2.3. Ensuring feasible parameter ranges

Applying the aforementioned method directly causes issues when there is insufficient ballistic coefficient controllability available to achieve the desired Δu_i and Δt_i . In such a case, because it is easier to control the along-track position than the cross-track position with aerodynamic drag, a set of control parameters should be selected that precisely achieve the desired Δu_i and achieve a Δt_i as close as possible to the desired value. With Δu_i calculated using Eq. (20), the minimum and maximum achievable Δt_i for this u_i can be calculated by first solving Eq. (25) for Δt_i to get

$$\Delta t_t = \frac{C_{b20}(\Delta t_{20}\Delta u_{10} - \Delta t_{10}\Delta u_{20}) + C_{b2}\Delta t_{10}\Delta u_t}{C_{b2}\Delta u_{10}} \tag{27}$$

For a maximum Δt_t , the largest C_{b2} and smallest C_{b1} that yield the correct Δu_t must be used. This allows the satellite to spend as much of its orbit as possible at higher altitudes where the orbital period is longer and hence the time required to achieve a given Δu is also longer. Substituting $C_{b_{max}}$ for C_{b2} in Eq. (24) gives the C_{b1} needed to achieve the desired Δu_t if $C_{b2} = C_{b_{max}}$. If the required C_{b1} is greater than $C_{b_{min}}$, then the combination of C_{b1} and C_{b2} is valid. If not, C_{b2} is too large and the greatest C_{b2} that yields a feasible C_{b1} can be calculated by solving Eq. (24) for C_{b2} and substituting $C_{b_{min}}$ for C_{b1} to get

$$C_{b2} = \frac{C_{b_{min}}\Delta u_{20}C_{b20}}{C_{b_{min}}\Delta u_t - \Delta u_{10}C_{b10}} \tag{28}$$

The maximum Δt_t can be found by substituting the calculated values of C_{b1} and C_{b2} into Eq. (27). Similarly, the minimum Δt_t requires the maximum valid C_{b1} and the minimum valid C_{b2} that yield the correct Δu_t so that the satellite can spend the greatest amount of time at low altitudes where the orbital period is shorter. These C_b values can be found as explained earlier by substituting $C_{b_{min}}$ for C_{b2} in Eq. (24) and increasing C_{b2} as necessary to ensure $C_{b_{min}} < C_{b1} < C_{b_{max}}$.

If the desired Δt_t is within the feasible range for the given Δu_t , then the control parameters required for proper targeting can be calculated using Eqs. (24)–(26). If not, Δt_t should be set to either the minimum or maximum feasible value to minimize the magnitude of the difference between the desired Δt_t and the best achievable Δt_t .

It follows from Eq. (17) that proper latitude targeting is achieved by any u_t satisfying

$$\Delta u_t = \Delta u_{it} + 2k\pi \tag{29}$$

where k is an integer and Δu_{it} is the Δu_t initially calculated for proper latitude targeting. In each analytical targeting iteration, a range of Δu_t values to test should be calculated. Based on the value of Δt_d , a range of Δu_t values that give proper latitude targeting can be calculated that will contain the Δu_t that minimizes the longitude targeting error. If Δt_d is positive, orbit lifetime must be increased and more orbits are needed, while if it is negative, lifetime must be decreased and fewer orbits are needed. The lower bound for the increase in orbit lifetime per orbit (T_l) is given by the orbital period of a satellite with zero altitude while the upper bound (T_u) applies to a satellite with a equal to the initial semi major axis of the guidance trajectory (a_i).

$$T_l = 2\pi\sqrt{\frac{R_e^3}{\mu}} \tag{30}$$

$$T_u = 2\pi\sqrt{\frac{a_i^3}{\mu}} \tag{31}$$

From Eq. (30) and Eq. (31), the range of Δu_t values to test can be calculated as follows where each Δu_t must satisfy Eq. (29). If $\Delta t_d \leq 0$

$$\Delta u_t \in \left[\Delta u_{it} + 2\pi \cdot \text{floor}\left(\frac{\Delta t_d}{T_l} - 1\right), \Delta u_{it} + 2\pi \cdot \text{ceil}\left(\frac{\Delta t_d}{T_u} + 1\right) \right] \tag{32}$$

while if $\Delta t_d > 0$

$$\Delta u_t \in \left[\Delta u_{it} + 2\pi \cdot \text{floor}\left(\frac{\Delta t_d}{T_u} - 1\right), \Delta u_{it} + 2\pi \cdot \text{ceil}\left(\frac{\Delta t_d}{T_l} + 1\right) \right] \tag{33}$$

The *floor* function rounds its argument down to the nearest integer while the *ceil* function rounds its argument up to the nearest integer. The Δu_t limits specified by Eq. (32) and Eq. (33) should always be updated if necessary to ensure that they are within the absolute Δu_t limits. The greatest orbit lifetime and hence the maximum Δu_t occurs when $C_{b1} = C_{b2} = C_{b_{min}}$ and is calculated using Eq. (22) as

$$\Delta u_{t_{max}} = \frac{C_{b10}\Delta u_{10} + C_{b20}\Delta u_{20}}{C_{b_{min}}} \tag{34}$$

By similar reasoning, the minimum Δu_t occurs when $C_{b1} = C_{b2} = C_{b_{max}}$ and is calculated by

$$\Delta u_{t_{min}} = \frac{C_{b10}\Delta u_{10} + C_{b20}\Delta u_{20}}{C_{b_{max}}} \tag{35}$$

To fully explore the control space, all Δt_t values that are between the minimum and maximum orbit life and satisfy the equation

$$\Delta t_t = \Delta t_{it} + \left(\frac{2\pi}{\omega_e - \Omega_{avg}} \right) n \tag{36}$$

where n is an integer should be tested with each Δu_t value in the range given by Eq. (32) and Eq. (33). Note that the maximum and minimum Δt_t can be calculated via a similar method using Eq. (23) as

$$\Delta t_{t_{max}} = \frac{C_{b10}\Delta t_{10} + C_{b20}\Delta t_{20}}{C_{b_{min}}} \tag{37}$$

$$\Delta t_{t_{min}} = \frac{C_{b10}\Delta t_{10} + C_{b20}\Delta t_{20}}{C_{b_{max}}} \tag{38}$$

For each tested combination of Δt_t and Δu_t , the time controllability (t_c) should be recorded. t_c characterizes the available orbit lifetime control margin and is defined as follows where $\Delta t_{t_{min}}$ and $\Delta t_{t_{max}}$ are the minimum and maximum terminal times for the desired Δu_t , and $\Delta t_{t_{des}}$ is the desired Δt_t :

- If $\Delta t_{t_{des}}$ cannot be achieved for the desired Δu_t

$$t_c = -\min(|\Delta t_{t_{min}} - \Delta t_{t_{des}}|, |\Delta t_{t_{max}} - \Delta t_{t_{des}}|) \tag{39}$$

- If $\Delta t_{t_{des}}$ can be achieved for desired Δu_t

$$t_c = \min(|\Delta t_{t_{min}} - \Delta t_{t_{des}}|, |\Delta t_{t_{max}} - \Delta t_{t_{des}}|) \tag{40}$$

As mentioned previously, a limitation of this analytical theory is that it requires the swap point to occur at the same semi major axis in the new and initial numerically propagated trajectories. To circumvent this limitation, analytical control parameter solutions for initial t_{swap} values between the minimum and maximum feasible t_{swap} in increments of t_i seconds can be tested. The minimum t_{swap} is 0 and the maximum t_{swap} occurs if C_{b1} is maintained all the way to the terminal point and is given by

$$t_{s_{max}} = t_{10} + \frac{C_{b20}\Delta t_{20}}{C_{b10}} \tag{41}$$

The Δt_1 , Δt_2 , Δu_1 , and Δu_2 values corresponding to a trajectory with this new t_{swap} can be analytically calculated as described previously by dividing this new trajectory and the initial numerically propagated trajectory into three phases before the terminal point as shown in Fig. 5 and calculating the time and change in argument of latitude in each phase using Eq. (5) and Eq. (6). These newly calculated Δt and Δu values can be used directly as the Δt_{10} , Δt_{20} , Δu_{10} , and Δu_{20} in the aforementioned analytical control solution. The testing of the full range of t_{swap} values implicitly allows the swap point to occur at all semi-major axes below the initial semi-major axis, facilitating a full exploration of the control space. Among all tested scenarios, the combination of parameters that yields the largest t_c should be chosen and the C_{b1} , C_{b2} , and t_{swap} corresponding to these parameters should be returned.

3.3. Drag-work enforcement method

The drag-work enforcement method minimizes the discrepancies between the analytical and numerical solutions by ensuring that the work done by drag in the numerical solution at time t_g matches the expected work done by drag at that time in the analytical solution. The

rate at which work (per unit mass) is done by drag (\dot{W}_d) is defined as

$$\dot{W}_d = \mathbf{a}_d \cdot \mathbf{v} = -C_b \rho v_{\infty} \mathbf{v}_{\infty} \cdot \mathbf{v} \quad (42)$$

To use this method, the total work done by aerodynamic drag is treated as a state variable (in addition to ECI position and velocity) in the orbit propagation and is found by numerically integrating Eq. (42) along the orbit. To analytically calculate the expected \dot{W}_d at some point in time t_w for a new trajectory with the same initial conditions as an initial numerically propagated trajectory but a different C_b profile, the trajectories can be divided into phases defined by initial and final semi major axes where the C_b does not change in either trajectory as shown in Fig. 5. The times required for each phase of the new and old trajectories (Δt_i and Δt_{i0} values) can be determined using Eq. (5). From these times, one can determine the phase of the new trajectory that the satellite would be located in at t_w . If t_w is in phase i , the time between the beginning of phase i and t_w in the new trajectory can be calculated by

$$\Delta t_{i_w} = t_w - \sum_{k=1}^{i-1} \Delta t_k \quad (43)$$

From Δt_{i_w} , the time since the start of phase i in the initial trajectory (Δt_{i0_w}) at which the satellite is at the same energy state (same semi major axis) as the satellite in the initial trajectory at t_w can be calculated by

$$\Delta t_{i0_w} = \frac{C_{bi} \Delta t_{i_w}}{C_{bi0}} \quad (44)$$

The time since t_0 at which the old trajectory is at the same energy state as the new trajectory at t_w can then be calculated by

$$t_{w_{eq}} = \Delta t_{i0_w} + \sum_{k=1}^{i-1} \Delta t_{k0} \quad (45)$$

Because the analytical solution assumes that drag force is the only effect that can reduce a satellite's orbital energy, \dot{W}_d at $t_{w_{eq}}$ in the initial numerically propagated will be equal to the expected \dot{W}_d at t_w in the new trajectory.

Once, the desired work done by drag at t_g is determined, C_b is iteratively adjusted to ensure that this desired \dot{W}_d is achieved. This is done by first numerically propagating the satellite's orbit for t_g seconds with the prescribed C_b profile and recording the \dot{W}_d at the end of the propagation. A drag scaling factor, s_d , is defined based on the difference between the desired and actual \dot{W}_d at t_g as

$$s_d = \frac{\dot{W}_{d_{des}}}{\dot{W}_{d_{act}}} \quad (46)$$

All C_b values between t_0 and t_g are multiplied by s_d and that portion of the trajectory is re-propagated with the updated C_b profile. The process continues until the actual \dot{W}_d is within a specified tolerance of the desired \dot{W}_d .

Note that sometimes, if the nominal C_b is already at $C_{b_{min}}$ or $C_{b_{max}}$, multiplication by s_d may result in $C_b < C_{b_{min}}$ or $C_b > C_{b_{max}}$. However, because the developed algorithms assume that some C_b margin will remain for guidance tracking, minor violations of the C_b constraints can be allowed and will only serve to slightly reduce this margin. These minor C_b control margin reductions did not adversely affect the ability to track guidances in the tested scenarios.

3.4. Back-stepping method

With the shrinking horizon strategy, a percentage (t_g seconds) of the beginning of each trajectory propagated with the analytically calculated set of control parameters is retained and used for the guidance. The remainder of the trajectory is utilized to analytically calculate a new set of control parameters which are then propagated to create the next portion of the guidance. If t_c is a small positive number when using

the full numerically propagated trajectory, using this smaller portion of the trajectory may result in insufficient controllability remaining to target the desired de-orbit point using aerodynamic drag (negative t_c). In such a case, t_g may have been too large. Such cases can be handled by reducing t_g by a certain factor and continuing to do this until there is sufficient controllability in the remainder of the numerical trajectory or a maximum number of such “back-steps” is reached. The maximum number of back-steps and the reduction in t_g per back-step are up to the user based on the required guidance accuracy and available computational power. In this study, sufficient accuracy was achieved with the drag-work enforcement method in the tested cases and back-stepping was not needed.

3.5. Latitude targeting for error reduction and terminal orbit characterization

In the analytical solution, it is assumed that the new trajectory to be analyzed experiences the same changes in orbital elements as the initial trajectory after the terminal point since both trajectories maintain the same C_b after this point. However, the oblateness of Earth, the rotating atmosphere, and the temporal and spatial variations in density (even at the same altitude) can render this assumption invalid and result in divergences between analytical and numerical solutions. To remedy this, after each numerically propagated trajectory, the spacecraft's C_b just a few hours before the terminal point (time t_{mod}) is modified to ensure proper latitude targeting. This ensures that the spacecraft is flying through the correct region of the atmosphere at the end of the trajectory and provides a more accurate characterization of the terminal behavior of the satellite that can be utilized in future analytical solutions. If the modified C_b is within the range of feasible spacecraft C_b values and the resulting total guidance error is less than some predefined threshold, the guidance generation algorithm is considered complete and this final modification of C_b and the change in the final trajectory that results from it is included in the guidance.

Once the increase in argument of latitude (Δu_d) required to achieve proper latitude targeting has been calculated, the change in C_b needed to achieve this (without regard to the total orbit lifetime) can be calculated based on the initial Δu_{init} that occurs between t_{mod} and the terminal point. The ratio of the initial to the required Δu during this period is given by

$$r_{\phi} = \frac{\Delta u_{init}}{\Delta u_{init} + \Delta u_d} \quad (47)$$

To achieve the desired Δu_d , all C_b values after t_{mod} must be multiplied by r_{ϕ} and t_{swap} must be adjusted using Eq. (26) to ensure that the C_b swap point (if applicable) occurs at the same semi major axis as in the initial trajectory. In this case, it is permissible if some of the resulting C_b values exceed the feasible range because the goal is to better characterize the behavior of the satellite after the terminal point, not before. Note that the satellite always maintains the same $C_{b_{term}}$ after the terminal point.

In the final shrinking horizon guidance generation step, making a few final attempts to correct only the latitude error through variations in C_b reduces the overall guidance error in a number of cases. In such a scenario, it is desirable to maintain C_b values within the acceptable range while attempting to achieve a desired Δu_d . To do this, r_{ϕ} can be calculated as before using Eq. (47) and multiplied by all C_b values after t_{mod} to get the new desired C_b values. No matter what, the trajectory after t_{mod} will have at most one swap in C_b and can be decomposed into the familiar C_{b10} , C_{b20} , Δu_{10} , Δu_{20} , Δt_{10} , and Δt_{20} . For the final latitude targeting attempt, t_{mod} can be set to zero to take advantage of the full C_b controllability of the trajectory. Note that if only one C_b is maintained between t_{mod} and t_{term} , then $\Delta t_{10} = \Delta u_{10} = 0$ and $C_{b10} = C_{b20}$. If the desired C_{b1} is greater than $C_{b_{max}}$, C_{b1} can be reduced to $C_{b_{max}}$ and the resulting change in orbit lifetime can be calculated by Eq. (3) assuming the swap point occurs at the same semi major axis. C_{b2} can then be

modified according to Eq. (3) to ensure that the desired Δu_d is maintained. t_{swap} can then also be modified according to Eq. (26) to ensure that the swap point occurs at the same semi major axis. If both C_{b1} and C_{b2} exceed $C_{b_{\text{max}}}$, then both C_b values should be reduced to $C_{b_{\text{max}}}$ to achieve as close to the desired Δu_d as possible. If $C_{b2} > C_{b_{\text{max}}}$, then C_{b2} must be reduced to $C_{b_{\text{max}}}$ and C_{b1} increased to ensure that the desired Δu_d is achieved. A similar procedure applies if either C_b value is below $C_{b_{\text{min}}}$.

4. Guidance tracking algorithm

Due to uncertainties in the drag force, the spacecraft will eventually drift from the guidance if the desired ballistic coefficient profile is applied open loop. While it is possible to re-generate the guidance once the drift exceeds a given threshold, guidance generation is computationally expensive and there is no guarantee that a new guidance with equally low error will exist from the new spacecraft initial conditions. For this reason, feedback control techniques must be utilized to vary the commanded spacecraft C_b based on the difference between the actual and desired position and velocity to ensure that the computed guidance is followed. While the spacecraft is tracking an initial guidance, new guidances can be periodically generated and tracked to take into account updated density forecasts.

4.1. Schweighart Sedwick relative motion dynamics for feedback control

The Schweighart Sedwick (SS) equations of relative motion [25] can be utilized to specify the evolution of the position and velocity of the spacecraft relative to the guidance at any given time when the separation between the spacecraft and the guidance is small compared to the radius of Earth. The relative position and velocity are specified in the non-inertial Local-Vertical-Local-Horizontal (LVLH) frame centered on a fictitious satellite that is following the guidance trajectory with the x-axis pointing along the zenith vector (up), the z-axis aligned with the angular momentum vector, and the y-axis completing the right-handed coordinate system [27] as shown in Fig. 6. Note that the LVLH frame can be specified entirely based on the guidance position and velocity (\mathbf{r}_g and \mathbf{v}_g) at the relevant point in time. The basis vectors of the LVLH frame expressed in the Earth Centered Inertial (ECI) frame are

$${}^E \hat{\mathbf{i}} = \frac{{}^E \mathbf{r}_g}{r_g} \tag{48}$$

$${}^E \hat{\mathbf{k}} = \frac{{}^E \mathbf{r}_g \times {}^E \mathbf{v}_g}{{}^E \mathbf{r}_g \times {}^E \mathbf{v}_g} \tag{49}$$

$${}^E \hat{\mathbf{j}} = {}^E \hat{\mathbf{k}} \times {}^E \hat{\mathbf{i}} \tag{50}$$

The direction cosine matrix that transforms vectors from the ECI frame to the LVLH frame can be written in terms of the LVLH basis vectors expressed in the ECI frame as

$$R_{E2L} = \begin{bmatrix} {}^E \hat{\mathbf{i}}^T \\ {}^E \hat{\mathbf{j}}^T \\ {}^E \hat{\mathbf{k}}^T \end{bmatrix} \tag{51}$$

The position and velocity of the spacecraft relative to the guidance as seen by an observer in the LVLH frame are given by Ref. [27].

$$\delta \mathbf{r} = \mathbf{r}_{\text{sc}} - \mathbf{r}_g \tag{52}$$

$$\delta \mathbf{v} = \mathbf{v}_{\text{sc}} - \mathbf{v}_g - \left(\frac{\mathbf{r}_g \times \mathbf{v}_g}{r_g^2} \right) \times \delta \mathbf{r} \tag{53}$$

Note that the subscript “sc” denotes the spacecraft while “g” denotes the guidance. If the vectors used to calculate Eqs. (52) and (53) are

expressed in the ECI frame, the relative position and velocity will also be expressed in the ECI frame and can be converted to the LVLH frame through a pre-multiplication by R_{E2L} .

If it is assumed that J_2 and two-body gravity are the only perturbations, $\delta \mathbf{r} \ll R_e$, and $\mathbf{r}_g \cdot \mathbf{v}_g \approx 0$ at all points (approximately circular orbit), the equations of relative motion can be linearized in a form known as the Schweighart Sedwick equations. The SS dynamics are a good approximation for low Earth orbit satellites because such orbits are nearly always close to circular. Differential drag can be incorporated into the SS dynamics as a control input that induces a relative acceleration. The SS approach can be utilized to incorporate additional perturbations into the linearized dynamics, but such a level of accuracy is unnecessary for this application. Considering only the in-plane relative position and velocity ($\delta x, \delta y, \delta \dot{x}, \delta \dot{y}$) because aerodynamic drag cannot be used for out-of-plane control and considering a relative $\delta \dot{y}$ due to a difference in the C_b between the spacecraft and the guidance, the SS linearization can be written as [25].

$$\begin{bmatrix} \delta \dot{x} \\ \delta \dot{y} \\ \delta \ddot{x} \\ \delta \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ b & 0 & 0 & d \\ 0 & 0 & -d & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta \dot{x} \\ \delta \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\rho v_g^2 \end{bmatrix} \Delta C_b \tag{54}$$

where

$$\Delta C_b = (C_{b_{\text{sc}}} - C_{b_g}) \tag{55}$$

$$n = \sqrt{\frac{\mu}{a^3}}, \quad c = \sqrt{1 + \frac{3J_2 R_e^2}{8a^2} [1 + 3\cos(2i)]}, \quad d = 2nc, \quad b = (5c^2 - 2)n^2 \tag{56}$$

4.2. LQR control for guidance tracking

With the dynamics of the relative motion between the spacecraft and the guidance given by Eq. (54) in the classic state-space form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{57}$$

it is possible to use a linear quadratic regulator (LQR) [24] control approach to drive the relative position and velocity to zero. An LQR controller derives the gain \mathbf{K} to yield the feedback control law

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \tag{58}$$

that drives the state to zero and minimizes the cost functional

$$J = \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \tag{59}$$

where \mathbf{Q} and \mathbf{R} are square weighting matrices of appropriate dimension. Because the state is four-dimensional and the control is one-dimensional, \mathbf{Q} and \mathbf{R} will be 4 by 4 and 1 by 1 matrices respectively and

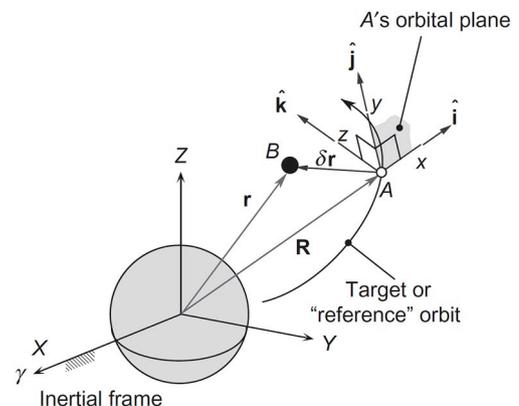


Fig. 6. Local vertical local horizontal (LVLH) frame [27].

\mathbf{K} will be a 1 by 4 gain matrix with the control given by

$$\Delta C_b = -[K_1 \quad K_2 \quad K_3 \quad K_4] \begin{bmatrix} \delta x \\ \delta y \\ \delta \dot{x} \\ \delta \dot{y} \end{bmatrix} \quad (60)$$

The LQR gain is optimal in the sense that no linear feedback control law can be derived that yields a lower value of J as $t \rightarrow \infty$. However, the practical performance of the controller is heavily dependent on \mathbf{Q} and \mathbf{R} which weight the relative importance of driving the state to zero as fast as possible and executing minimal control effort. In many cases, \mathbf{Q} and \mathbf{R} are selected through trial and error, but for this problem there is a rigorous way to define these matrices. The method for determining \mathbf{Q} and \mathbf{R} is a key novelty in this work and is not found in existing literature analyzing this problem. Because along-track error is far greater than radial error in general, radial error is considered only in terms of its contribution to along-track error. For this reason, setting

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (61)$$

yields superior guidance tracking performance for a given value of \mathbf{R} than any other value of \mathbf{Q} with a comparable matrix 2-norm (as defined in Ref. [28]). This value of \mathbf{Q} implies that the controller should only be concerned with errors in the along-track (δy) direction. However, if there is any radial (δx) error or relative velocity ($\delta \dot{x}$, $\delta \dot{y}$) error, that will automatically result in a δy error over time. Thus the controller drives to zero errors in δx , $\delta \dot{x}$, and $\delta \dot{y}$ in its attempt to drive δy to zero, but permits radial and velocity errors to increase temporarily if doing so is necessary to drive δy to zero as quickly and efficiently as possible.

With \mathbf{Q} fixed, \mathbf{R} must be set based on the desired magnitude of the spacecraft response to deviations from the guidance. This can be done by first defining Δr_{sat} as the desired δy at which the commanded change in ballistic coefficient will be equal to $C_{bmax} - C_{bmin}$. That is, the controller will be guaranteed to saturate at $\delta y = \Delta r_{sat}$. The LQR gain \mathbf{K} can first be computed using an arbitrary initial \mathbf{R} ($\mathbf{R} = 10,000$ used in this work) and then recomputed after updating \mathbf{R} based on the initially obtained \mathbf{K} to enforce controller saturation at $\delta y = \Delta r_{sat}$. The equation to update \mathbf{R} is

$$\mathbf{R}_{new} = \mathbf{R}_0 \left(\frac{C_{bmax} - C_{bmin}}{\mathbf{K}_0 [0 \quad \Delta r_{sat} \quad 0 \quad 0]} \right)^2 \quad (62)$$

Note that there is no benefit to changing the magnitude of \mathbf{Q} as all performance variations that could result from a change in the magnitude of \mathbf{Q} can be achieved through a manipulation of \mathbf{R} for this problem.

Finally, the \mathbf{A} and \mathbf{B} matrices in the SS dynamics will change as the spacecraft decays to a lower orbit and experiences a different ambient density and semi major axis. To account for this, \mathbf{K} can be recomputed using the LQR strategy with new \mathbf{A} and \mathbf{B} matrices whenever the current atmospheric density (ρ_{new}) differs from the density used to compute the previous gain (ρ_{old}) by a factor of p or more where

$$p = \frac{\rho_{new}}{\rho_{old}} \quad (63)$$

In the current work, $p = 1.2$ is used. Plots of the performance of this controller in specific scenarios are included in Section 6.

4.3. Controller saturation

Controller saturation occurs when the commanded control is beyond what the actuator is physically capable of providing [29]. While some systems become unstable when in a saturated state, the guidance tracking algorithm remains stable and returns the spacecraft to the guidance under saturation conditions. If the desired C_b is below C_{bmin} ,

the desired C_b is simply set to C_{bmin} . Similarly, if the desired C_b is greater than C_{bmax} , then C_b is set to C_{bmax} . The simulations conducted to verify the performance of the controller under saturation are discussed in Section 6.3.2.

4.4. Actuator deadband and performance limitations

To prevent the feedback law given in Eq. (60) from changing the spacecraft C_b for any infinitesimal change in the state vector, an actuator deadband is utilized. With the deadband approach, the ballistic coefficient of the spacecraft is not modulated until the difference between the current and desired ballistic coefficient is greater than a certain percentage (5% in this work) of the current ballistic coefficient.

The finite times required to achieve desired C_b changes (through attitude variations or drag device actuation) are also considered in this work. The assumption that 4 min would be required to go from C_{bmin} to C_{bmax} is made since this is an upper bound for the deployment times of current retractable drag devices [17,18]. In all tested cases, the controller was robust to non-instantaneous actuation and remained functional with the C_b deadband as evidenced by the simulation results in Section 6.

4.5. Extended Kalman Filter for relative orbit determination and noise filtering

Spacecraft utilizing the guidance tracking algorithm would likely receive position and velocity measurements from a GPS unit, such as the piNAV-L1 [30], which can be converted to relative position and velocity using the procedure in Section 4.1. Because aerodynamic drag is such a weak force, the controller must react vigorously to any perceived orbit errors, making the system very sensitive to the noise present in GPS measurements. Fortunately, the structure of the relative position and velocity needed for the guidance tracking algorithm lends itself to the implementation of an Extended Kalman Filter (EKF) [26] that provides a more accurate estimate of the spacecraft state relative to the guidance than a direct filtering of the ECI position and velocity.

Since the linearized dynamics are already known and the control signal has the form $\mathbf{u} = -\mathbf{K}\mathbf{x}$, the matrix exponential of the $(\mathbf{A} - \mathbf{BK})$ matrix from Eq. (54) and the computed LQR gain can be utilized to calculate the state transition matrix (Φ) as

$$\Phi = e^{(\mathbf{A} - \mathbf{BK})\Delta t} \quad (64)$$

where Δt is the time since the last state estimate. In a linear Kalman filter (popular if computing power is limited), the state estimate \mathbf{x}_i^- can be calculated by multiplying the previous state estimate by Φ . With the EKF, \mathbf{x}_i^- can be calculated more accurately (though at a higher computational cost) by converting the previous state estimate \mathbf{x}_{i-1}^+ to ECI position and velocity, numerically propagating the orbit for time Δt , and converting the final result back to relative position and velocity based on the guidance state at that time. Let this conversion and propagation process be denoted by the function $f(t_{i-1}, t_i, \mathbf{x}_{i-1}^+)$ which will be utilized in the predict stage of the EKF. The new covariance matrix estimate P_i^- can be calculated by a similarity transform using Φ . The state and covariance estimates for the predict stage are thus

$$\begin{aligned} \mathbf{x}_i^- &= f(t_{i-1}, t_i, \mathbf{x}_{i-1}^+) \\ P_i^- &= \Phi_i P_{i-1}^+ \Phi_i^T + Q \end{aligned} \quad (65)$$

where Q is the process noise covariance matrix.

The update stage of the EKF involves updating the a-priori state and covariance estimates based on some measurement \mathbf{z}_i . The update stage is described in Ref. [26] as follows

$$\begin{aligned} K_i &= P_i G^T (G P_i G^T + W)^{-1} \\ \mathbf{x}_i^+ &= \mathbf{x}_i^- + K_i (\mathbf{z}_i - G \mathbf{x}_i^-) \\ P_i^+ &= (I - K_i G) P_i^- \end{aligned} \quad (66)$$

Where W is the measurement noise covariance matrix, Λ is a term greater than 1 ($\Lambda=1.02$ used in this work) utilized to ensure that P does not become too small (filter smugness [31]), and G specifies the linear mapping between the measurement and the state as

$$\mathbf{z} = G\mathbf{x} \tag{67}$$

Note that in this scenario, \mathbf{z} is the raw GPS measurement converted to in-plane relative position and velocity so G is a 4×4 identity matrix.

5. Algorithm simulations

One thousand Monte Carlo simulations of the guidance generation algorithm were conducted to verify the ability to calculate an achievable drag profile and corresponding trajectory that if followed, will allow the spacecraft to de-orbit in a desired location. In all cases, guidance generation was set to stop if at any point a trajectory was generated with a guidance error of less than 25 km. For each of the resulting guidances, a guidance tracking simulation was conducted with realistic models of GPS sensor noise and density uncertainty to validate the ability to follow a guidance in a realistic environment. Simulations of specific cases were also conducted to assess particular aspects of the algorithms' performance.

5.1. Environmental force and uncertainty modeling

A high fidelity orbit propagator was created in MATLAB including gravitational perturbations modeled by geopotential coefficients through degree and order 10 using the procedure from Montenbruck's book [32] and the EGM2008 gravity model [33]. Atmospheric density was given by the NRLMSISE-00 model [34] with historic F10.7 and Ap indices [35]. Because Ap is updated every 3h and F10.7 is updated daily, a cubic spline was utilized to interpolate between the historic F10.7 and Ap values as suggest by Vallado [36]. This ensured continuously varying atmospheric indices and a continuously varying density. Use of a continuous rather than discrete density profile minimized the discrepancies between the numerical and analytical solutions and significantly improved the performance and convergence rate of the guidance generation algorithm. In each simulation, the ballistic coefficient (defined in Eq. (1)) used at any point in time was required to lie in a specified C_b range. Solar radiation pressure, solar gravity, and lunar gravity were found to be insignificant in low Earth orbits and were neglected. Optionally, a user could use the guidance generation algorithm with a more computationally efficient but less accurate orbit propagator such as SGP4 [34] in applications where computing power is limited, but this will result in less realistic guidance trajectories that are more difficult to track.

By far the greatest source of uncertainty is in the aerodynamic drag force due to the difficulties in modeling the density and drag coefficient. For the purposes of guidance generation, NRLMSISE-00 density was used directly with the interpolated F10.7 and Ap indices. To characterize the effects of drag estimation errors when trying to track the guidance, the nominal drag force was multiplied by an error coefficient when simulating the guidance tracking algorithm. This error coefficient was the combination of a bias term and three sinusoidal terms and was calculated by

$$k_{err} = k_0 + \sum_{i=1}^3 k_i \sin\left(\frac{2\pi}{T_i}t - \phi_i\right) \tag{68}$$

The T values were set to $T_1=26$ days, $T_2=1$ day, $T_3=5400$ s based on observed density variations on real satellite missions [36–38]. These corresponded to the synodic period (sun rotation), Earth day, and approximate orbital period. ϕ values were randomly selected from a uniform distribution between 0 and 2π . k_0 values were randomly selected from a uniform distribution between 0.77 and 1.3 and the other k values were set to $k_1=.25$, $k_2=.1$, $k_3=.1$ based on historically observed

drag estimation errors [38]. All guidance tracking algorithms were run assuming that the maximum C_b achievable by the spacecraft was a factor of two greater than the maximum allowable guidance C_b and the minimum achievable C_b was a factor of two less than the smallest allowable guidance C_b . This ensured that there would always be a sufficient C_b margin to correct for the simulated drag uncertainty errors, and any tracking errors would be a result of suboptimal controller performance rather than a complete saturation of the actuator. Ballistic coefficient modifications resulting from the drag-work enforcement methods were allowed to exceed the minimum and maximum guidance C_b values by a factor of up to 1.05. In the rare cases where this happened, the margins between the ranges of achievable tracker C_b values and the range of guidance C_b values were slightly reduced, but this did not have an adverse effect on tracker performance.

5.2. Sensor noise model

Because drag is weaker and takes far longer to achieve a desired orbit change than conventional space propulsion systems, the control system must respond vigorously to any small difference between the current and desired spacecraft state. Unfortunately, this makes the controller much more sensitive to navigation noise than a propulsive orbit control algorithm would be, making it crucial to verify that the controller is still functional in a noisy environment. The noise model used in this work is based on the error in position and velocity measurements applicable to the piNAV-L1 CubeSat GPS unit [30]. The manufacturer claims that the piNAV's position and velocity errors have a standard deviation of not more than 5 m and 5 cm/s respectively with simulated position estimation errors shown in Fig. 7 [30].

As Fig. 7 shows, the errors are not always zero mean. To simulate this, Gaussian noise terms with the specified standard deviations were added to the true ECI position and velocity along with sinusoidally varying position and velocity bias errors given by

$$\Delta \mathbf{r}_{bias} = \begin{bmatrix} .001 \\ -.005 \\ .002 \end{bmatrix} \sin\left(\frac{2\pi t}{5400}\right) m, \Delta \mathbf{v}_{bias} = \begin{bmatrix} .00005 \\ .00005 \\ .000025 \end{bmatrix} \sin\left(\frac{2\pi t}{5400}\right) m/s \tag{69}$$

5.3. Case-specific simulation results

Simulations were conducted with different effects included to assess the performance of the guidance generation, guidance tracking, and state estimation algorithms under a variety of different circumstances. The results of these simulations are discussed below.

5.3.1. Guidance generation and noise-free tracking with drag bias error

A sample guidance that lasted 11.7 days and resulted in 12.3 km total targeting error was generated for the following scenario:

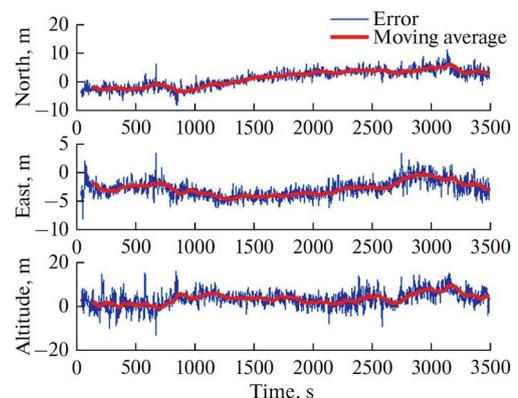


Fig. 7. piNAV-L1 simulated position errors for ISS orbit [30].

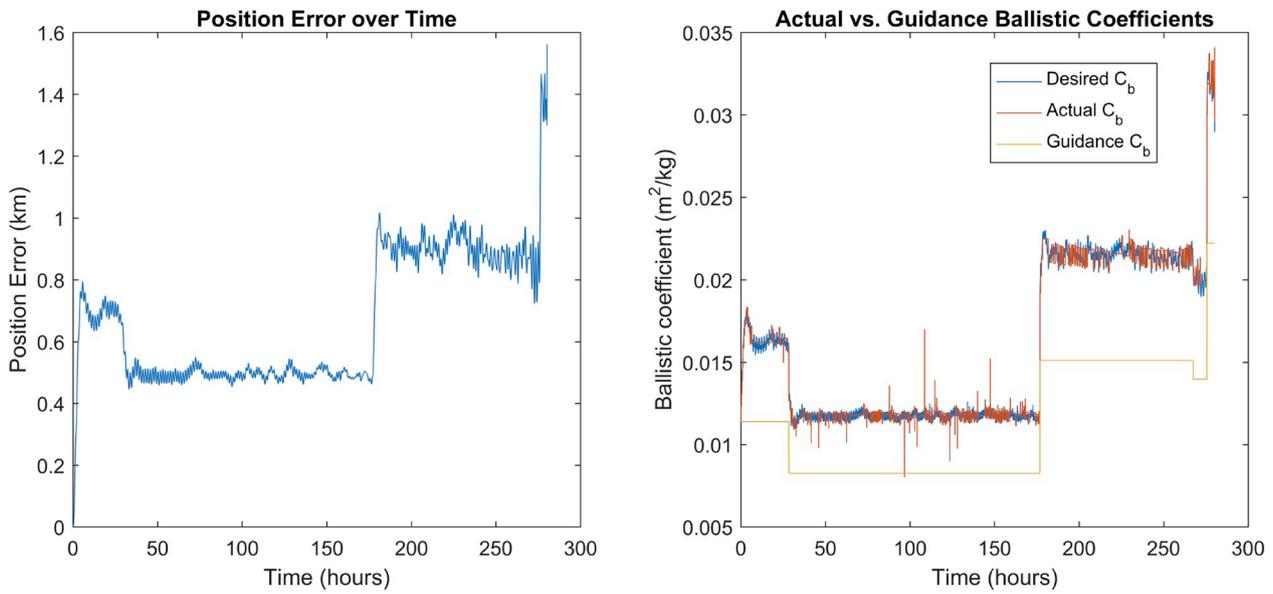


Fig. 8. Tracking position error and C_b over time with drag bias error of 0.7.

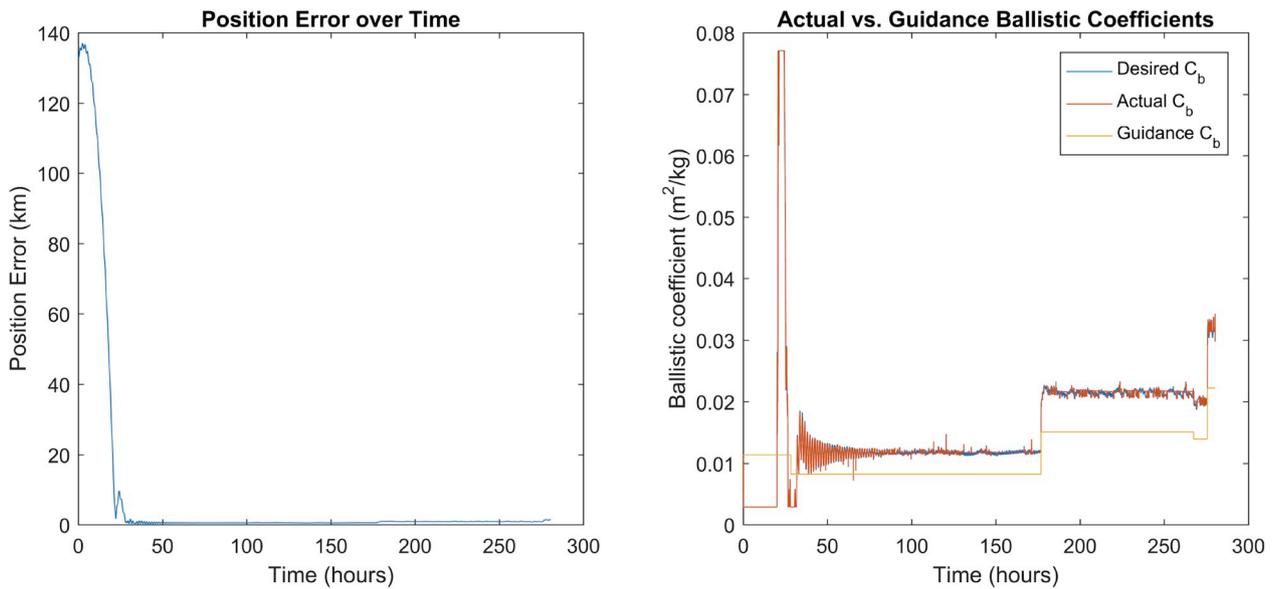


Fig. 9. Tracking position error and C_b over time with 132 km initial error and actuator saturation.

- epoch = January 24, 2004, 6:48:29.48 UTC
- initial osculating orbital elements: ($a = 6715.97\text{km}$, $e = .000471$, $\Omega = 214.24^\circ$, $\omega = .1790^\circ$, $\theta = 359.77^\circ$, $i = 70.67^\circ$)
- $C_{b\text{min}} = .0059\text{m}^2/\text{kg}$, $C_{b\text{max}} = .0386\text{m}^2/\text{kg}$, $C_{b\text{term}} = .0222\text{m}^2/\text{kg}$
- target de-orbit location: -54.54° latitude, 160.85° longitude at a 70 km geocentric altitude
- Targeting set to begin at 6500 s of longitude controllability as defined Ref. [22].

The guidance tracking algorithm was run with perfect knowledge of the state, a 5% actuator dead-band, the assumption that the drag device takes 4 min to fully deploy, and a constant bias error of 0.7 (nominal drag force values all multiplied by .7). Tracking was continued down to a geodetic altitude of 90 km. As shown in Fig. 8, the tracking algorithm was able to maintain the spacecraft on the guidance with an error of less than 2 km by automatically adjusting the C_b to compensate for the

difference in the drag properties between the guidance and the tracking simulations. The drag actuator needed to run for only 0.13% of the orbit lifetime to produce the indicated C_b fluctuations.

5.3.2. Tracker performance under actuator saturation

If the along-track separation between the satellite and the guidance is greater than Δr_{sat} , the controller will saturate, meaning that the commanded C_b will be greater than that achievable by the satellite. Fortunately, this controller performs well under saturation and will return the satellite back to the guidance as long as the separation is not so great that the SS relative motion equations become invalid. Fig. 9 illustrates the ability of the controller to return the satellite to the guidance used in Section 6.3.1 given an initial true anomaly error of 0.02 radians (about 132 km) and a density bias of 0.7 as before.

To further verify the operation of the tracker under saturation, one hundred tracking simulations were conducted for different guidances

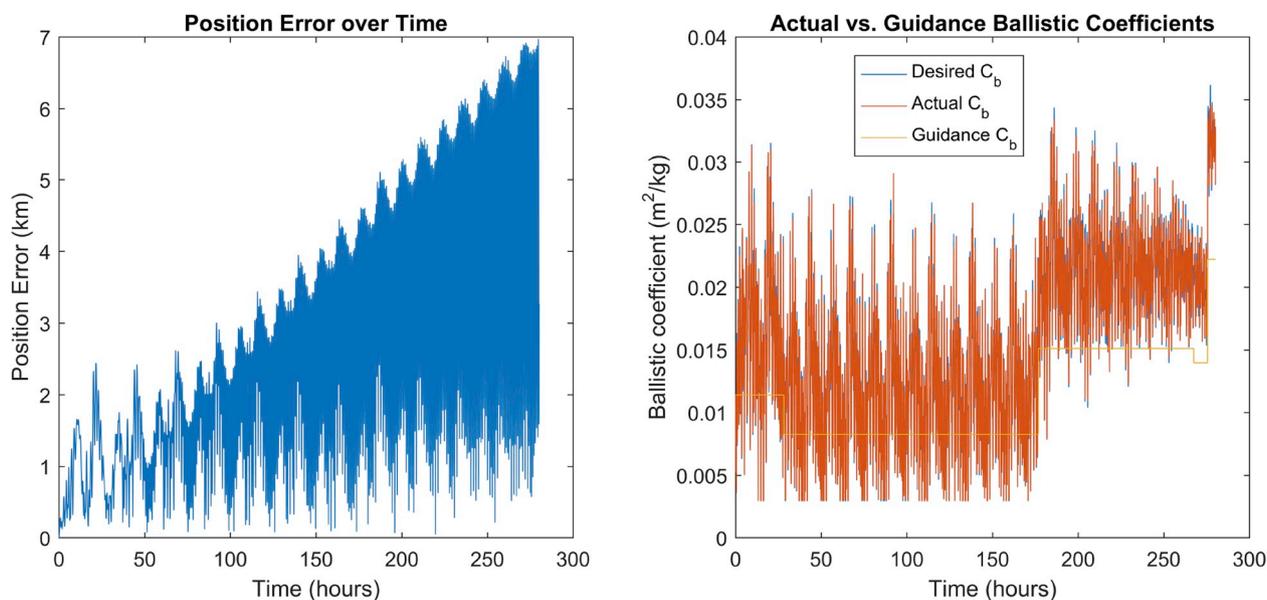


Fig. 10. Tracking position error and C_b over time with low order gravity model.

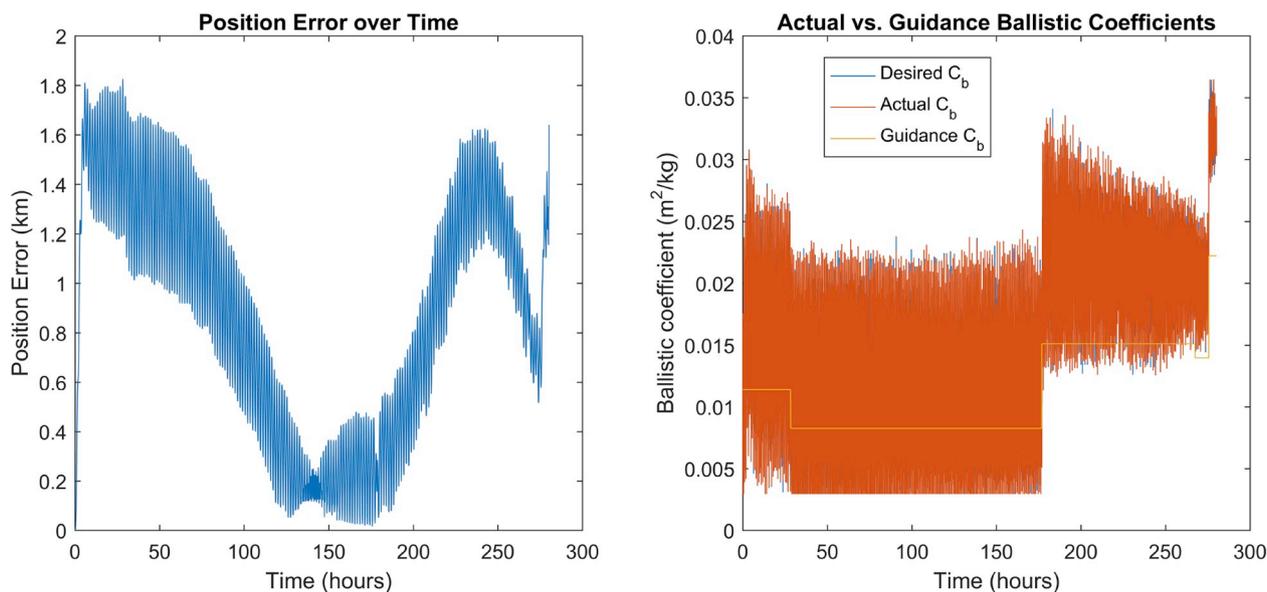


Fig. 11. Tracking position error and C_b over time with noisy GPS position and velocity measurements.

with an initial tracking true anomaly error of 0.02 radians with $\Delta r_{sat} = 5\text{ km}$ and in all cases, the tracker was able to return to the guidance.

5.3.3. Tracking performance with lower order gravitational models

As described in Section 6.2, the tracking algorithm is very sensitive to noise or un-modeled perturbations. To illustrate this, the scenario from Section 6.3.1 was re-run with only the J_2 perturbation considered in the tracking simulation. The result is similar to a case where the tracking algorithm is used on a real spacecraft with a guidance created considering only J_2 gravitational perturbations. As shown in Fig. 10, the perturbations resulting from the higher order gravity terms resulted in greater control effort (actuator running 2.8% of the time) in an attempt to track the perceived state errors. Cases where the guidance was generated using a point-mass gravity model were un-trackable in a realistic environment. This demonstrates why a high fidelity orbit propagator is necessary for guidance generation and why guidance generation algorithms that do not incorporate the full nonlinear

dynamics such as the analytical solution in Ref. [19] will result in guidances that are not easily trackable. Gravitational perturbations beyond degree and order ten were found to be insignificant and not worth the additional computational cost of including in guidance generation. To prove this and to further validate the orbit propagator, the High Fidelity Orbit Propagator (HPOP) in AGI's Systems ToolKit (STK) software was utilized to generate a guidance with historic NRLMS-ISE-00 density and gravitational perturbations through degree and order 21. The tracking of this guidance was simulated using the in-house MATLAB orbit propagator and the satellite was able to effectively track the guidance with minimal error and minimal actuator effort.

5.3.4. Noisy guidance tracking without filter

As shown in Fig. 11, if unfiltered GPS position and velocity measurements are used, the controller will respond erratically in its attempt to track the noise. While the tracker was able to keep the satellite on the guidance, the actuator was running almost constantly (73% of the time) to achieve the ballistic coefficient profile shown in Fig. 11.

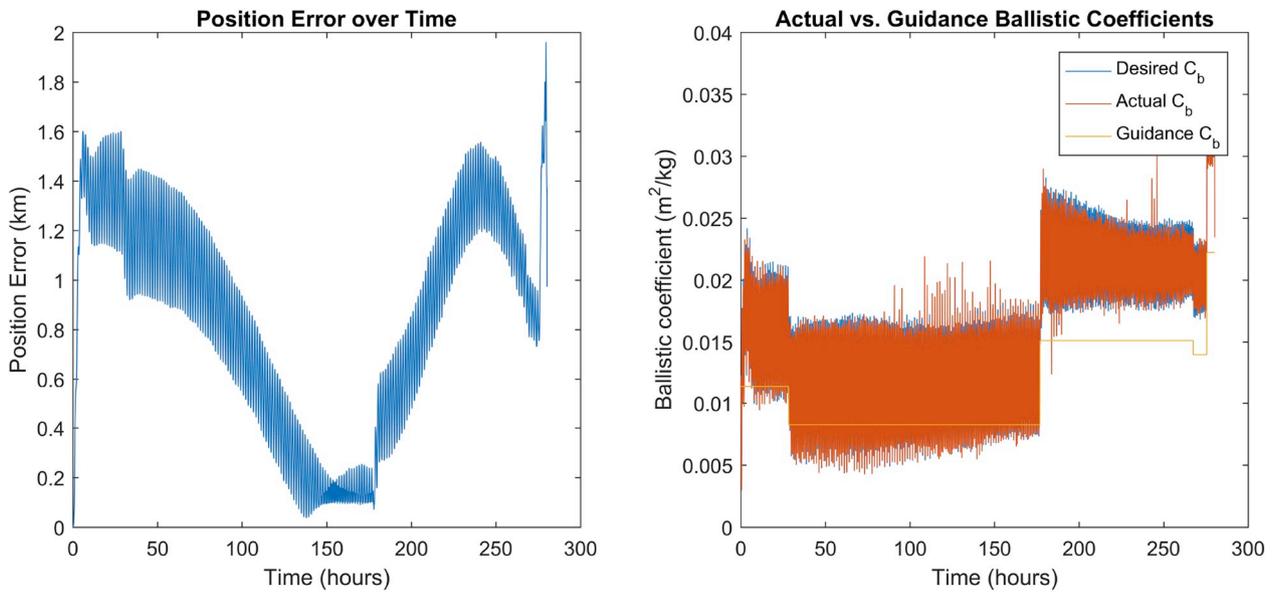


Fig. 12. Tracking position error and C_b over time with noisy state estimates and EKF

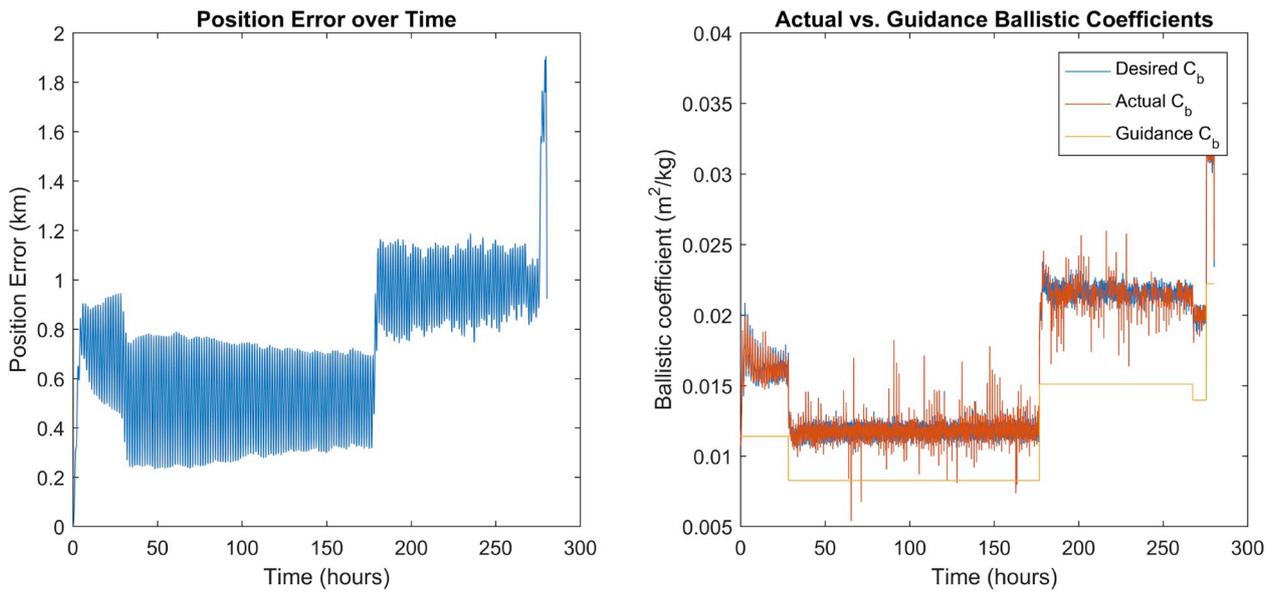


Fig. 13. Tracking position error and C_b over time with purely Gaussian noise and EKF

5.3.5. Noisy guidance tracking with filter

The scenario from Section 6.3.4 was re-run with an EKF as detailed in Section 5. The controller was still able to maintain tracking within 2 km as shown in Fig. 12, but the actuator was only running 2.5% of the time. The majority of the actuator run time was due to tracking the sinusoidally varying bias errors on the GPS position and velocity, because a noise filter cannot remove bias errors.

The aforementioned scenario was run with zero mean GPS measurement error with only Gaussian noise, and the results are plotted in Fig. 13. Fig. 13 shows that the EKF can very effectively remove Gaussian noise and simulation results were very similar to those found in the scenario from Section 6.3.1 with the actuator running 0.68% of the time. Additionally, by filtering on position and velocity relative to the guidance, more accurate state estimates are made than by filtering directly on the inertial position and velocity of the satellite.

5.3.6. Noisy guidance tracking with complete drag error and EKF

Fig. 14 shows the position error and the desired, actual, and

guidance ballistic coefficients over time for the most realistic simulation case including sensor noise and drag estimation errors with an EKF utilized to filter the noise. Despite the sensor noise and model uncertainties, the system still maintained tracking within 2 km but more actuator run time (2.66% of total time) was required to correct for the drag force prediction errors.

5.3.7. Tracking simulation with errors due to F10.7 and A_p forecasts

In addition to the errors inherent in the NRLMSISE-00 density model itself, the F10.7 and A_p indices upon which the model depends must be forecast and these forecasts introduce additional uncertainty. To characterize the impacts of these forecasting uncertainties on the guidance tracking, a guidance was generated with an Epoch of June 17, 2016 with the same target location, C_b range, and initial conditions as in Section 6.3.1. In the guidance generation simulations, a set of F10.7 and A_p data were used that were generated on June 16, 2016² and included

² <https://celestrak.com/SpaceData/SW-Last5Years.txt>.

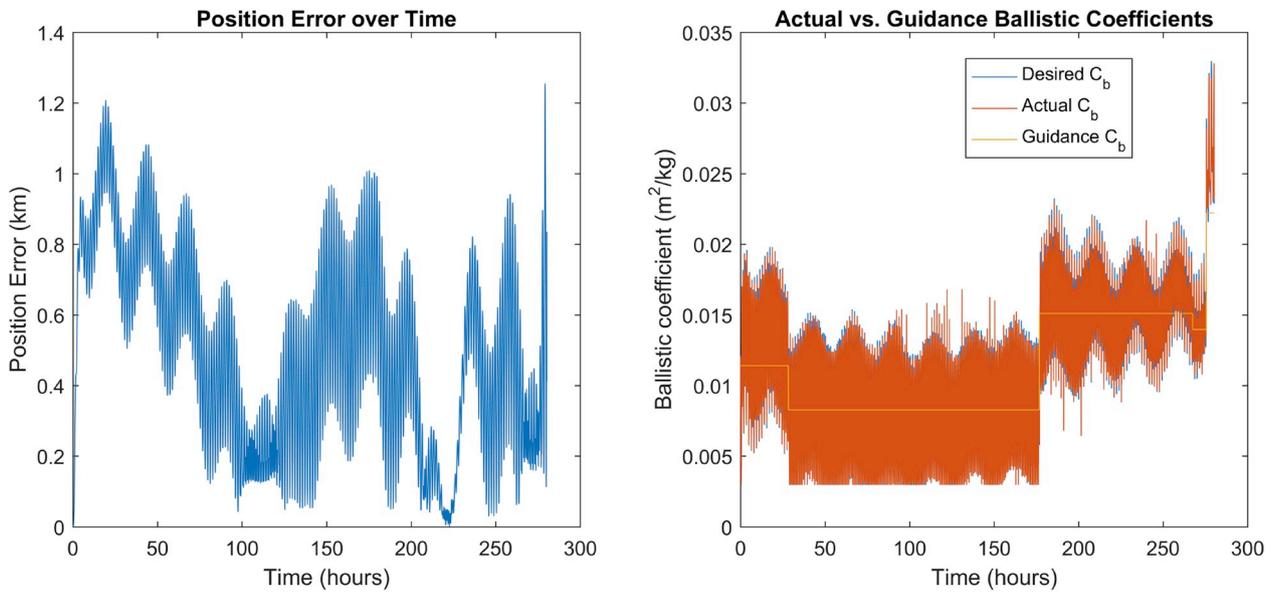


Fig. 14. Tracking position error and C_b over time with drag uncertainties and GPS measurement noise.

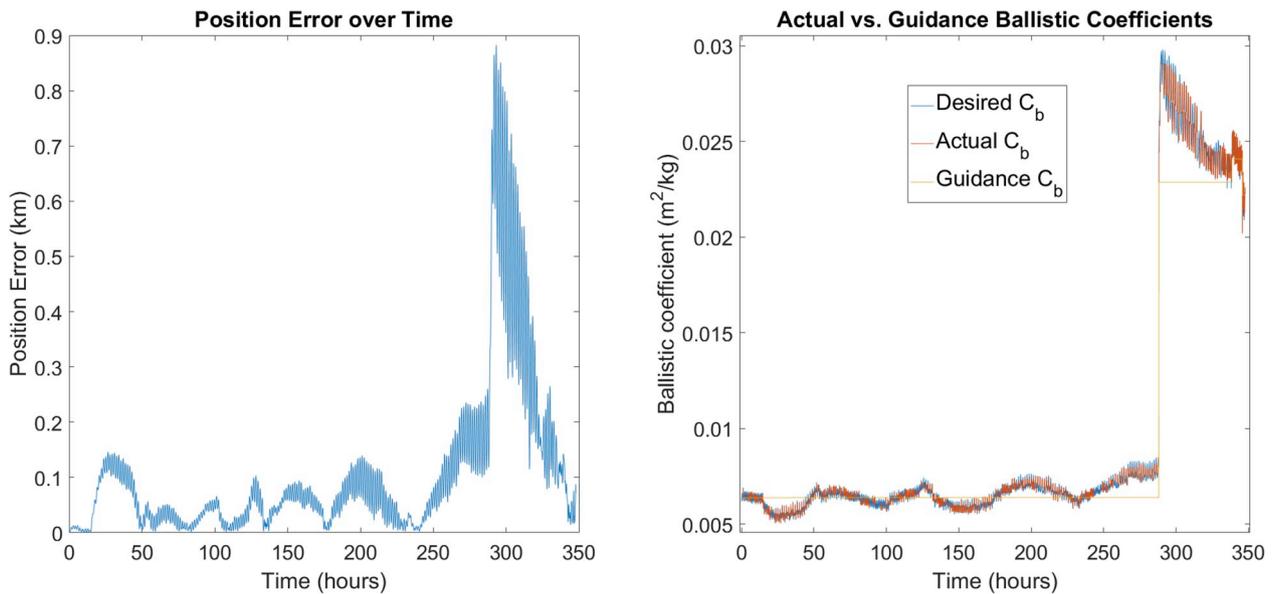


Fig. 15. Tracking position error and C_b over time with guidance generated using forecast density.

Table 1
Monte Carlo simulation parameters.

Variable	Range	Distribution
Semi Major Axis	[6698, 6718] km	Uniform
True Anomaly	[0, 360] degrees	Uniform
Eccentricity	[0, .004]	Uniform
Right Ascension	[0, 360] degrees	Uniform
Argument of the Perigee	[0, 360] degrees	Uniform
Inclination	[1, 97] degrees	Uniform
Impact Latitude	[min reachable lat +.1, max reachable lat -.1]	Uniform
Impact Longitude	[-180, 180] degrees	Uniform
$C_{b_{max}}$	[.033, .067]	Uniform
$C_{b_{min}}$	[.0053, .027]	Uniform
Epoch	[11/1/2003, 11/1/2014]	Uniform

historic data through June 16 and forecast F10.7 and A_p values for dates after June 16, 2016. Because this forecast was made on June 16, 2016, this data accurately depicts the atmospheric data that would be available to generate a guidance for a satellite that was in orbit on June 16, 2016. The tracking of this guidance was then simulated using historic F10.7 and A_p data throughout the entire simulation time frame. Sensor noise and additional density errors were not included in this simulation to isolate the effects of F10.7 and A_p prediction errors. The effort required by the actuator to follow the guidance and the corresponding tracking error provide a good indication of the tracking performance errors that would result from guidances generated with forecast (instead of observed) F10.7 and A_p indices. As shown in Fig. 15, the satellite is able to follow the guidance with a lower tracker error and less actuator effort than in the case shown in Fig. 14. The density errors resulting from the inaccurate F10.7 and A_p forecasts are also roughly sinusoidal in nature. This shows that the sinusoidal density

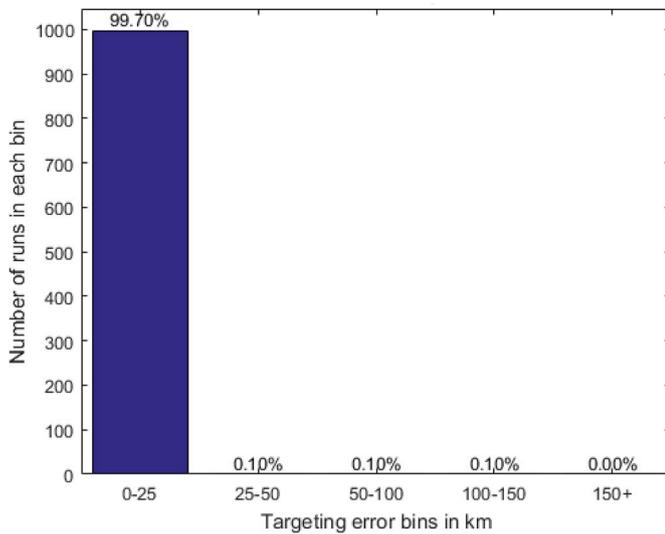


Fig. 16. Guidance errors from Monte Carlo simulations.

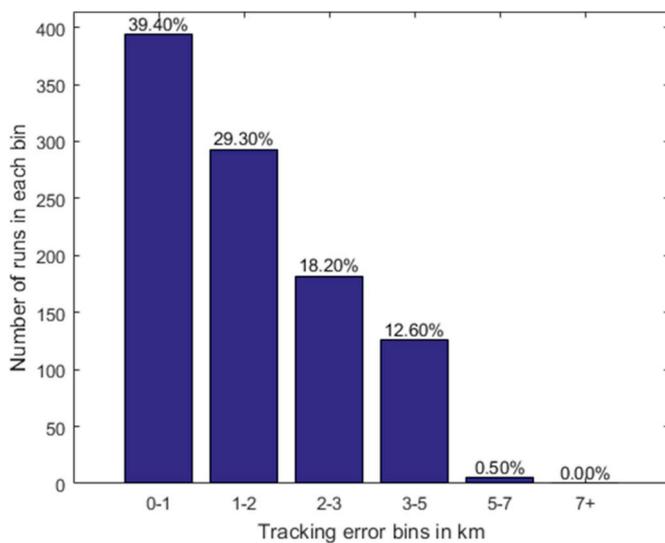


Fig. 17. Final guidance tracking errors from Monte Carlo simulations.

errors utilized in previous simulations (Eq. (68)) accurately characterize the nature of the expected density uncertainties and in general, represent a greater uncertainty than would normally be associated with the F10.7 and A_p forecasting.

5.4. Monte Carlo simulation results

One thousand simulations of the guidance generation algorithm to a geodetic altitude of 120 km were conducted with randomized initial conditions as shown in Table 1. The mean guidance error was 12.5 km with a standard deviation of 7.5 km leading to a 99% confidence interval for the expected average guidance error of 11.9 km to 13.2 km. All guidance errors were below 106 km and are shown in Fig. 16. All but three guidances converged to below the 25 km threshold at which no further attempts to improve the guidance were made. After running guidance tracking algorithms on all generated guidances down to a geodetic altitude of 120 km, all final tracking errors were less than 6 km with an average error of 1.6 km. Tracking algorithms were run for each guidance with density uncertainties and sensor noise simulated as was done in Section 6.3.6. Results of the tracking simulations are shown in Fig. 17.

6. Conclusions

This paper presents novel guidance generation, guidance tracking, and state estimation algorithms capable of guiding a spacecraft to a desired re-entry location solely by modulating the spacecraft's aerodynamic drag. These algorithms offer significant improvements over the state of the art and are able to operate effectively despite model uncertainties, sensor noise, and actuator delays. Monte Carlo campaigns and case-specific simulations were conducted to validate the effectiveness and robustness of the presented algorithms.

The guidance and tracking errors meet NASA's debris mitigation guidelines [15] which state that the probability of human casualty from re-entering debris must be less than 1 in 10,000. Because the desired re-entry location would likely be over the South Pacific Ocean Uninhabited Area (SPOUA) where there is no inhabited land for thousands of kilometers, it is extremely unlikely for the targeting error to be so large that some spacecraft debris re-enters over land and poses a threat to persons or property. As such, this re-entry point targeting algorithm could be utilized for re-entering higher-stakes items like rocket upper stages as long as they have a means of modulating their aerodynamic drag, resulting in significant fuel savings. The guidance tracking algorithm could also be used for spacecraft rendezvous using aerodynamic drag or in any orbital maneuvering scenario where a satellite must track a guidance using aerodynamic drag.

Acknowledgments

The authors wish to thank a.i. solutions for sponsoring this investigation under a NASA Kennedy Space Center subcontract (project LSP 15-025: A Drag Device for Controlled De-Orbiting of LEO Spacecraft). This work was also supported by funds from a NASA Space Technology Research Fellowship.

References

- [1] R.P. Patera, W.H. Ailor, The realities of reentry disposal, Proceedings of the AAS/AIAA Space Flight Mechanics Meeting, American Astronautical Society/AIAA, Monterey, California, 1998, pp. 9–11 https://www.researchgate.net/profile/Russell_Patera/publication/252732286_The_Realities_of_Reentry_Disposal/links/55a3cf6a08ae72349bbcf971.pdf.
- [2] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, R. Twiggs, CubeSat: a new generation of picosatellite for education and industry low-cost space experimentation, Proceedings of the 14th Annual AIAA/USU Conference on Small Satellites, Logan, UT, 2000 <http://digitalcommons.usu.edu/smallsat/2000/AII2000/32>.
- [3] C.L. Leonard, W.M. Hollister, E.V. Bergmann, Orbital formation keeping with differential drag, J. Guid. Contr. Dynam. 12 (1) (1989) 108–113, <https://doi.org/10.2514/3.20374> <http://arc.aiaa.org/doi/10.2514/3.20374>.
- [4] B.S. Kumar, A. Ng, K. Yoshihara, A.D. Ruitter, Differential drag as a means of spacecraft formation control, IEEE Trans. Aero. Electron. Syst. 47 (2) (2011) 1125–1135, <https://doi.org/10.1109/TAES.2011.5751247>.
- [5] L. Mazal, D. Pérez, R. Bevilacqua, F. Curti, Spacecraft rendezvous by differential drag under uncertainties, J. Guid. Contr. Dynam. 39 (8) (2016) 1721–1733, <https://doi.org/10.2514/1.G001785> <http://arc.aiaa.org/doi/10.2514/1.G001785>.
- [6] D. Pérez, R. Bevilacqua, Lyapunov-based spacecraft rendezvous maneuvers using differential drag, AIAA Guidance, Navigation, and Control Conference, 2011, p. 6630, <https://doi.org/10.2514/6.2011-6630> <http://arc.aiaa.org/doi/pdf/10.2514/6.2011-6630>.
- [7] D. Pérez, R. Bevilacqua, Differential drag spacecraft rendezvous using an adaptive lyapunov control strategy, Acta Astronaut. 83 (2013) 196–207, <https://doi.org/10.1016/j.actaastro.2012.09.005> <http://www.sciencedirect.com/science/article/pii/S0094576512003621>.
- [8] D. Pérez, R. Bevilacqua, Differential drag-based reference trajectories for spacecraft relative maneuvering using density forecast, J. Spacecraft Rockets 53 (1) (2016) 234–239, <https://doi.org/10.2514/1.A33332> <http://arc.aiaa.org/doi/10.2514/1.A33332>.
- [9] S.R. Omar, J.M. Wersinger, Satellite formation control using differential drag, Proceedings of 53rd AIAA Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics, Kissimmee, FL, 2014, <https://doi.org/10.2514/6.2015-0002> <http://arc.aiaa.org/doi/abs/10.2514/6.2015-0002>.
- [10] M. Pastorelli, R. Bevilacqua, S. Pastorelli, Differential-Drag-based roto-translational control for propellant-less spacecraft, Acta Astronaut. 114 (2015) 6–21, <https://doi.org/10.1016/j.actaastro.2015.04.014> <http://linkinghub.elsevier.com/retrieve/pii/S0094576515001708>.
- [11] C. Foster, H. Hallam, J. Mason, Orbit determination and differential-drag control of Planet Labs cubesat constellations, Advances in the Astronautical Sciences

- Astrodynamics, Volume 156 ArXiv: 1509.03270. URL <http://arxiv.org/abs/1509.03270>.
- [12] E. Lokcu, R.L. Ash, T.A. Force, A de-orbit system design for CubeSat payloads, 2011 5th International Conference on Recent Advances in Space Technologies (RAST), 9–11 June 2011, 2011, pp. 470–474 <http://www.jaqarssoftware.com/downloads/partners/edu/RAST2011%20deorbit%20paper%20PID1720489.pdf>.
- [13] B. Cotton, On-orbit results from CanX-7 drag sail de-orbit mission, Proceedings of the 31st Annual AIAA/USU Conference on Small Satellites, Logan, Utah, 2017 <http://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=3672&context=smallsat>.
- [14] P. Harkness, M. McRobb, P. Lützkendorf, R. Milligan, A. Feeney, C. Clark, Development status of AEOLDOS – a deorbit module for small satellites, *Adv. Space Res.* 54 (1) (2014) 82–91, <https://doi.org/10.1016/j.asr.2014.03.022> <http://www.sciencedirect.com/science/article/pii/S0273117714001999>.
- [15] NASA, Process for limiting orbital debris, Tech. Rep. NASA-std-8719, (May 2012), p. 14A.
- [16] Z. Wu, R. Hu, X. Qu, X. Wang, Z. Wu, Space debris reentry analysis methods and tools, *Chin. J. Aeronaut.* 24 (4) (2011) 387–395, [https://doi.org/10.1016/S1000-9361\(11\)60046-0](https://doi.org/10.1016/S1000-9361(11)60046-0) <http://linkinghub.elsevier.com/retrieve/pii/S1000936111600460>.
- [17] C. Mason, G. Tilton, N. Vazirani, J. Spinazola, D. Guglielmo, S. Robinson, R. Bevilacqua, J. Samuel, Origami-based drag sail for CubeSat propellant-free maneuvering, Proceedings of the 5th Nano-satellite Symposium, Tokyo, Japan, 2013.
- [18] D. Guglielmo, S. Omar, R. Bevilacqua, L. Fineberg, J. Treptow, B. Poffenberger, Y. Johnson, Drag deorbit device: a new standard reentry actuator for CubeSats, First IAA Conference on Space Situational Awareness, to Appear in the AIAA Journal of Spacecraft and Rockets, Orlando, FL, 2017.
- [19] J. Virgili, P. Roberts, Atmospheric interface reentry point targeting using aerodynamic drag control, *J. Guid. Contr. Dynam.* 38 (3) (2015) 1–11, <https://doi.org/10.2514/1.G000884>.
- [20] S. Dutta, A. Bowes, A.M. Dwyer Cianciolo, C. Glass, R.W. Powell, Guidance scheme for modulation of drag devices to enable return from low Earth orbit, AIAA Atmospheric Flight Mechanics Conference, 2017, <https://doi.org/10.2514/6.2017-0467> <http://arc.aiaa.org/doi/pdf/10.2514/6.2017-0467>.
- [21] S.R. Omar, R. Bevilacqua, D. Guglielmo, L. Fineberg, J. Treptow, S. Clark, Y. Johnson, Spacecraft deorbit point targeting using aerodynamic drag, *J. Guid. Contr. Dynam.* (2017) 1–7, <https://doi.org/10.2514/1.G002612> <https://arc.aiaa.org/doi/10.2514/1.G002612>.
- [22] S.R. Omar, R. Bevilacqua, Spacecraft de-orbit point targeting using aerodynamic drag, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2017, , <https://doi.org/10.2514/6.2017-1268> <http://arc.aiaa.org/doi/10.2514/6.2017-1268>.
- [23] MathWorks, Linear-quadratic Regulator (LQR) Design - MATLAB LQR, (2017) <https://www.mathworks.com/help/control/ref/lqr.html>.
- [24] G. Franklin, J. Powell, A. Emami-Naeini, *Feedback Control of Dynamic Systems*, fourth ed., Prentice Hall, Upper Saddle River, NJ, 2002.
- [25] S.A. Schweighart, R.J. Sedwick, High-fidelity linearized J model for satellite formation flight, *J. Guid. Contr. Dynam.* 25 (6) (2002) 1073–1080, <https://doi.org/10.2514/2.4986> <https://doi.org/10.2514/2.4986>.
- [26] A. Kelly, A 3d State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles, Tech. Rep. The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1994 <http://www.dtic.mil/dtic/tr/fulltext/u2/a282853.pdf>.
- [27] H. Curtis, *Orbital Mechanics for Engineering Students*, second ed., Elsevier, Burlington, MA, 2009.
- [28] D. Bau III, L. Trefethen, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.
- [29] N. Nise, *Control Systems Engineering*, fifth ed., John Wiley and Sons, 2008.
- [30] P. Kovář, piNAV L1—GPS receiver for small satellites, *Gyroscopy and Navigation* 8 (2) (2017) 159–164, <https://doi.org/10.1134/S2075108717020079> <http://link.springer.com/10.1134/S2075108717020079>.
- [31] S. Słojkowski, J. Lowe, J. Woodburn, Orbit determination for the lunar reconnaissance orbiter using an extended kalman filter, 25th International Symposium on Space Flight Dynamics (ISSFD) 2015, Oct. 2015, pp. 19–23 Munich; Germany <https://ntrs.nasa.gov/search.jsp?R=20150019754>.
- [32] O. Montenbruck, E. Gill, *Satellite Orbits*, first ed., Springer Berlin Heidelberg, 2005.
- [33] N. Pavlis, S. Holmes, S. Kenyon, J. Factor, *An Earth Gravitation Model to Degree 2160*, (Apr. 2008), p. EGM2008.
- [34] D. Vallado, *Fundamentals of Astrodynamics and Applications*, fourth ed., Microcosm Press, Hawthorne, CA, 2013.
- [35] National Oceanic and Atmospheric Administration, USAF 45-Day Ap and F10.7cm Flux Forecast, (2017) <http://www.swpc.noaa.gov/products/usaf-45-day-ap-and-f107cm-flux-forecast>.
- [36] D.A. Vallado, D. Finkleman, A critical assessment of satellite drag and atmospheric density modeling, *Acta Astronaut.* 95 (2014) 141–165, <https://doi.org/10.1016/j.actaastro.2013.10.005> <http://www.sciencedirect.com/science/article/pii/S0094576513003755>.
- [37] F. Marcos, B. Bowman, R. Sheehan, Accuracy of Earth's thermospheric neutral density models, Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Keystone, CO, 2006 <http://arc.aiaa.org/doi/pdf/10.2514/6.2006-6167>.
- [38] J.L. Lean, J.M. Picone, J.T. Emmert, G. Moore, Thermospheric densities derived from spacecraft orbits: application to the starshine satellites, *J. Geophys. Res.: Space Physics* 111 (A4) (2006) A04301, <https://doi.org/10.1029/2005JA011399> <http://onlinelibrary.wiley.com/doi/10.1029/2005JA011399/abstract>.



Sanny Omar is a PhD Candidate at the University of Florida. He graduated Summa Cum Laude with Honors from Auburn University in 2015 with a bachelor's degree in Aerospace Engineering and minors in business, Spanish, and computer science. He has conducted internships at GE Aviation, SpaceX, and NASA Ames and has been awarded the National Science Foundation Graduate Research Fellowship and NASA Space Technology Research Fellowship for his PhD studies. Sanny is also an instrument rated private pilot and a certified scuba diver. Sanny specializes in spacecraft dynamics and control and has extensive experience in controlling spacecraft orbits using aerodynamic drag.



Dr. Riccardo Bevilacqua is an Associate Professor of the Mechanical and Aerospace Engineering Department, at the University of Florida. He holds a M.Sc. in Aerospace Engineering (2002), and a Ph.D. in Applied Mathematics (2007), both from the University of Rome, “Sapienza”, Italy. Dr. Bevilacqua is the recipient of two Young Investigator Awards, from the Air Force Office of Scientific Research (2012) and the Office of Naval Research (2013), of the 2014 Dave Ward Memorial Lecture Award from the Aerospace Controls and Guidance Systems Committee, and of two Air Force Summer Fellowships (2012 and 2015). His research interests focus on spacecraft formation flight and space robotics. He has authored and co-authored more than 60 journal and conference publications on the topic. Dr. Bevilacqua is a corresponding member of the IAA and an associate fellow of the AIAA.