1<sup>st</sup> IAA Conference on Space Situational Awareness (ICSSA)

## Orlando, FL, USA

IAA-ICSSA-17-01-02

# A HYBRID ADAPTIVE CONTROL ALGORITHM FOR SPACECRAFT GUIDANCE TRACKING USING AERODYNAMIC DRAG

Sanny R. Omar<sup>(1)</sup>, Riccardo Bevilacqua<sup>(2)</sup>

 <sup>(1)</sup>University of Florida, ADAMUS Laboratory, Gainesville, Florida 32611, USA 502-216-2168, sanny.omar@ufl.edu
 <sup>(2)</sup>University of Florida, ADAMUS Laboratory, Gainesville, Florida 32611, USA, bevilr@ufl.edu

Keywords: Spacecraft, Drag, Guidance, Tracking, Re-Entry Point Targeting

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 a ccuracy, 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 24.3 km and a maximum error below 750 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 90 km with a final position error under 1 0 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.

### Nomenclature

a	=	Semi-major axis (km)	
$C_b$	=	Ballistic coefficient $\left(\frac{m^2}{kg}\right)$	
е	=	Eccentricity	
ECI	=	Earth centered inertial	
EKF	=	Extended Kalman Filter	
i	=	Inclination (radians)	
$J_2$	=	Constant that describes Earth's oblateness	
K	=	LQR Gain	
LEO	=	Low Earth Orbit	
LQR	=	Linear quadratic regulator	
Q	=	LQR weighting matrix for state error	
R	=	LQR weighting matrix for actuator effort	
$R_e$	=	Radius of Earth	
r	=	Spacecraft position vector (km)	
Т	=	Period of orbit or sinusoidal noise term (s)	
t	=	Time (s)	
и	=	Argument of latitude (radians)	
V	=	Spacecraft velocity vector (km)	
$\theta$	=	True anomaly ( <i>radians</i> )	
$\mu$	=	Earth's gravitational parameter $(km^3/s^2)$	
Ω	=	Right ascension of ascending node (radians)	
ω	=	Argument of periapsis (radians)	
$\omega_a$	=	Average orbital angular velocity (radians/s)	
$\omega_e$	=	Earth rotation rate (radians/s)	

### 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, 5, 6, 7, 8, 9, 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 bangbang 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, 13, 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]. 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] or by changes in the spacecraft's cross-sectional area. Prior works by Virgili [18] and Dutta [19] discuss algorithms for aerodynamically-based re-entry control, but these algorithms have some limitations (further elaborated on in Ref [20]). 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, 20] 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 [22, 23] 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 [24], resulting in improved performance over prior algorithms which only account for two states (generally mean anomaly and semi major axis) and do not consider J2 perturbations in the dynamics. A means of analytically calculating the Q and 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 [25] 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 1,000 guidance generation and guidance tracking algorithms is 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, 20] but offers significant improvements. The algorithm in Ref. [20] 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  $C_{b_{term}}$  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 Ref. [20] and [21] and in this paper,  $C_b$  is defined as

$$C_b = \frac{C_d A}{2m} \tag{1}$$

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. [21], 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. This solution is based on the relation [21] that the time and true anomaly required for a spacecraft to decay from an initial to final semi major axis due to aerodynamic drag increase linearly with decreasing ballistic coefficient. Assuming a satellite with ballistic coefficient  $C_{b1}$  takes time  $t_1$  to achieve some change in semi major axis  $\Delta a$  and undergoes true anomaly change  $\Delta \theta_1$  during this drop, the time and true anomaly change a satellite with the same initial conditions and some different  $C_{b2}$  will undergo to achieve the same  $\Delta a$  is given by

$$\Delta t_2 = \frac{C_{b1} \Delta t_1}{C_{b2}} \tag{2}$$

$$\Delta \theta_2 = \frac{C_{b1} \Delta \theta_1}{C_{b2}} \tag{3}$$

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

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

As shown in Fig. 1, 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. In the last region (below the terminal point) both trajectories have the same  $C_b$  so they can be assumed to experience the same change in orbital elements between the terminal point and the de-orbit point. For the three phases before the terminal point, Eqs. (2-4) can be utilized to calculate the changes in time and orbital elements can be added to calculate the new trajectory.



Figure 1: Dividing Orbits into Phases for Analytical De-Orbit Point Calculation

final time and orbital elements, and hence the latitude and longitude, at the de-orbit point.

The argument of latitude (u) required to de-orbit at the desired latitude is calculated by recognizing that the *z*-component of the ECI position vector is equal to the last row of the perifocal to the ECI frame direction cosine matrix given by eq. 4.49 in [26] multiplied by the perifocal position vector.

$$r\sin(lat) = r \left[ \sin(\omega)\sin(i) \quad \cos(\omega)\sin(i) \quad \cos(i) \right] \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix}$$
(5)

Recognizing the trigonometric identity

$$\sin(u) = \sin(\omega + \theta) = (\sin(\omega)\cos(\theta) + \cos(\omega)\sin(\theta))$$
(6)

Eq. 5 can be simplified to

$$u = \sin^{-1} \left( \frac{\sin(lat)}{\sin(i)} \right) \tag{7}$$

Because the  $\sin^{-1}$  function only returns values between  $-\pi/2$  and  $\pi/2$ , the other *u* values that yields proper latitude targeting can be calculated by

$$u_2 = \pi - u_1 \tag{8}$$

For each calculated u, the increase in true anomaly  $(\Delta \theta_d)$  required for latitude targeting can be calculated based on the initial  $u_i$  as

$$\Delta \theta_d = \mod (u_d - u_i, 2\pi) \tag{9}$$



Figure 2: Prior Targeting Algorithm Schematic [20]

Using this mapping from the control parameters to the de-orbit point, the change in  $t_{swap}$  necessary for a desired  $\Delta \theta_d$  is calculated by [20]

$$\Delta t_{swap} = \frac{\Delta \theta_d C_{b2}}{\omega_{a2} \left( C_{b2} - C_{b1} \right)} \tag{10}$$

where  $\omega_{a2}$  is the average angular velocity during phase 2 of the orbit shown in Figure 1. The algorithm in Ref. [20] proceeds using a numerical optimization approach as shown in Fig. 2. The analytical solution is used to calculate the control parameters needed for targeting by decoupling the targeting of latitude and longitude. First, the value of  $t_{swap}$  that yields perfect latitude targeting but the lowest correctable longitude error is calculated using Eq. 10. The variations in the control parameters needed to correct that longitude error to the extent possible without upsetting the latitude targeting are then calculated using Eqs. (11-13) derived in Ref [20] with  $\Delta \theta_t$  as required for latitude targeting and  $\Delta t_t$  as required to correct for the remaining longitude error.

#### 2.2. Limitations of Prior Algorithm

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. This may make it impossible to perform latitude targeting through a variation of only  $t_{swap}$ , resulting in the algorithm failing to converge. This issue also hinders the ability to periodically regenerate the guidance (desirable if atmospheric forecasts change or the drag coefficient model is updated). 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  $t_{swap}$ . These control parameters cannot be used as an initial guess in a new guidance generation process because latitude targeting using  $t_{swap}$  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 guarantee that the newly generated guidance will have the same accuracy as the original guidance since the initial conditions are very different.

Another limitation of the prior algorithm is due to the trade-off between controllability and sensitivity. Because aerodynamic drag force is weak and only acts in the along-track direction, significant time is required to maneuver the satellite into a desired orbit. Thus, the controllability (set of reachable target locations) is drastically reduced when limited orbit lifetime is remaining. To ensure that every point with latitude below the orbital inclination is reachable using aerodynamic drag, maneuvering must begin on the order of two weeks before the expected end of the satellite's orbital life. As maneuvering begins increasingly earlier, however, even small perturbations become significant because their effects are propagated for several days. In particular, the analytical solution for the control parameters hinges on the assumption that a spacecraft with the new control parameters will experience approximately the same density vs. semi major axis profile as the spacecraft in the initial numerically propagated trajectory. This is a reasonable assumption if the changes in the control parameters are small but will never be perfectly true when a high fidelity drag model is used because different regions of space, even at the same altitude and time, experience different densities and atmospheric winds. With early maneuvering, orbital perturbations caused by these differences become significant and result in large deviations between the analytical estimates of de-orbit locations and the more accurate solutions derived from numerical orbit propagation. This often leads to either a failure of the algorithm to converge or unacceptably long simulations times. In Ref. [20], 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 in a reasonable amount of time (around 1 hour). The resulting longitude error of up to 1250 km due to the limited controllability was accepted as reasonable assuming that the algorithm would likely be utilized to steer the satellite to the South Pacific Ocean Uninhabited Area (SPOUA) for debris mitigation purposes.

### 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. 2.2 using an improved analytical solution and a receding horizon strategy [27]. As in the prior algorithm, the analytical solution calculates the control parameters needed for de-orbit point targeting, but both latitude and longitude targeting are handled in the same calculation and there are no restrictions on the initial  $C_b$  values. 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 receding horizon strategy is used in place of the numerical optimization approach and capitalizes on the benefits of high controllability at high initial altitudes and the reduced sensitivity to drag force perturbations at low altitudes. The general form of the receding 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  $t_{o}$ ) 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 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. This algorithm is depicted graphically in Fig. 3.



Figure 3: Basic Form of the Improved Guidance Generation Algorithm

### 3.2. Analytical Solution for Control Parameters

As shown in Ref. [21], assuming that  $t_{swap}$  occurs at the same semi-major axis in the new and initial trajectories, the  $C_{b1}$  and  $C_{b2}$  values needed to achieve both a desired total orbit lifetime  $\Delta t_t$  and desired total change in true anomaly  $\Delta \theta_t$  to the terminal point ( $t_{term}$  in Figure 1) can be calculated in terms of the initial numerically propagated trajectory as

$$C_{b2} = \frac{C_{b20}(\Delta t_{20}\Delta\theta_{10} - \Delta t_{10}\Delta\theta_{20})}{\Delta t_t \Delta\theta_{10} - \Delta t_{10}\Delta\theta_t}$$
(11)

$$C_{b1} = \frac{\Delta \theta_{10} C_{b10} C_{b2}}{\Delta \theta_t C_{b2} - \Delta \theta_{20} C_{b20}}$$
(12)

 $t_{swap}$  must also 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

$$t_{s_{new}} = \frac{t_{s_{old}}C_{b10}}{C_{b1}}$$
(13)

Note that variables with subscript 0 correspond to the initial numerically propagated trajectory. If a trajectory is numerically propagated with some initial set of control parameters, Eqs. (7-8) can be utilized to calculate the argument of latitude (*u*) required for the spacecraft to impact the correct latitude. Unless the target latitude is exactly equal to the orbit's inclination, there will always be two feasible *u* values; one corresponding to the ascending portion of the orbit (increasing latitude) and one corresponding to the descending portion of the orbit (decreasing latitude). If  $-pi/2 \le u < pi/2$  then the spacecraft is ascending along its orbit. For any other value of *u*, the spacecraft is descending or descending part of the orbit, the correct *u* and corresponding  $\Delta \theta_d$  can be chosen. The longitude error that would result if  $\Delta \theta_d$  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, adding  $\Delta \theta_d$  to the true anomaly, and calculating the impact latitude and longitude using the original de-orbit time. For

a given longitude error between this impact location and the desired impact location denoted by

$$\lambda_e = \lambda_{imp} - \lambda_{des} \tag{14}$$

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

$$\Delta t_d = \frac{\lambda_e}{\omega_e} \tag{15}$$

where  $\omega_e$  is the rotation rate of Earth. The total time and change in true anomaly to the terminal point required for latitude and longitude targeting can now be calculated as

$$\Delta t_t = \Delta t_{t0} + \Delta t_d \tag{16}$$

$$\Delta \theta_t = \Delta \theta_{t0} + \Delta \theta_d \tag{17}$$

These values of  $\Delta \theta_t$  and  $\Delta t_t$  can now be substituted into Eqs. 11-13 to calculate the set of control parameters needed for proper targeting.

Applying the aforementioned method directly causes issues when there is insufficient ballistic coefficient controllability available to achieve the desired  $\Delta \theta_t$  and  $\Delta t_t$ . 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 achieve the desired  $\Delta \theta_t$  and a  $\Delta t_t$  as close as possible to the desired value. With  $\Delta \theta_t$ calculated using Eq. 17, the minimum and maximum achievable  $\Delta t_t$  can be calculated by first solving Eq. 11 for  $\Delta t_t$  to get

$$\Delta t_t = \frac{C_{b20}(\Delta t_{20}\Delta\theta_{10} - \Delta t_{10}\Delta\theta_{20}) + C_{b2}\Delta t_{10}\Delta\theta_t}{C_{b2}\Delta\theta_{10}}$$
(18)

For a maximum  $\Delta t_t$ , the largest  $C_{b2}$  and smallest  $C_{b1}$  that yield the correct  $\Delta \theta_t$  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  $\Delta \theta$  is also longer. Substituting  $C_{bmax}$  for  $C_{b2}$  in Eq. 12 gives the  $C_{b1}$  needed to achieve the desired  $\Delta \theta_t$  if  $C_{b2} = C_{bmax}$ . If the required  $C_{b1}$  is greater than  $C_{bmin}$ , 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 solving Eq. 12 for  $C_{b2}$  and substituting  $C_{bmin}$  for  $C_{b1}$  to get

$$C_{b2} = \frac{C_{b_{min}} \Delta \theta_{20} C_{b20}}{C_{b_{min}} \Delta \theta_t - \Delta \theta_{10} C_{b10}}$$
(19)

The maximum  $\Delta t_t$  can be found by substituting the calculated values of  $C_{b1}$  and  $C_{b2}$  into Eq. 18. Similarly, the minimum  $\Delta t_t$  requires the maximum valid  $C_{b1}$  and the minimum valid  $C_{b2}$  that yield the correct  $\Delta \theta_t$  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. 12 and increasing  $C_{b2}$  as necessary to ensure  $C_{b_{min}} < C_{b1} < C_{b_{max}}$ .

If the desired  $\Delta t_t$  is within the feasible range for the given  $\Delta \theta_t$ , then the control parameters required for proper targeting can be calculated using Eqs. 12-13. If not,  $\Delta t_t$  should be set to either the minimum or maximum feasible value to minimize the magnitude of the difference between the desired  $\Delta t_t$  and the best achievable  $\Delta t_t$ .

It is worth noting that any total true anomaly change given by

$$\Delta \theta_t = \Delta \theta_{t_i} + 2n\pi \tag{20}$$

where *n* is an integer and  $\Delta \theta_{t_i}$  is the  $\Delta \theta_t$  initially calculated for proper latitude targeting will also provide proper latitude targeting. In each analytical targeting iteration, a range of  $\Delta \theta_t$  values to test should be calculated. Based on the value of  $\Delta t_d$ , a range of  $\Delta \theta_t$ values that give proper latitude targeting can be calculated that will be guaranteed to contain the  $\Delta \theta_t$  that minimizes the longitude targeting error. If  $\Delta t_d$  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 ( $a_i$ ).

$$T_l = 2\pi \sqrt{\frac{R_e^3}{\mu}} \tag{21}$$

$$T_u = 2\pi \sqrt{\frac{a_i^3}{\mu}}$$
(22)

From Eq. 21 and Eq. 22, the range of  $\Delta \theta_t$  values to test can be calculated as follows where each  $\Delta \theta_t$  must satisfy Eq. 20. If  $\Delta t_d \leq 0$ 

$$\Delta \theta_t \epsilon [\Delta \theta_{t_i} + 2\pi \cdot floor\left(\frac{\Delta t_d}{T_l} - 1\right), \Delta \theta_{t_i} + 2\pi \cdot ceil\left(\frac{\Delta t_d}{T_u} + 1\right)]$$
(23)

while if  $\Delta t_d > 0$ 

$$\Delta \theta_t \epsilon [\Delta \theta_{t_i} + 2\pi \cdot floor\left(\frac{\Delta t_d}{T_u} - 1\right), \Delta \theta_{t_i} + 2\pi \cdot ceil\left(\frac{\Delta t_d}{T_l} + 1\right)]$$
(24)

The *floor* function rounds its argument down to the nearest integer while the *ceil* function rounds its argument up to the nearest integer. The  $\Delta \theta_t$  limits specified by Eq. 23 and Eq. 24 should always be updated if necessary to ensure that they are within the absolute  $\Delta \theta_t$  limits. The greatest orbit lifetime and hence the maximum  $\Delta \theta_t$  occurs when  $C_{b1} = C_{b2} = C_{b_{min}}$  and is calculated using Eq. 3 as

$$\Delta \theta_{max} = \frac{C_{b10} \Delta \theta_{10} + C_{b20} \Delta \theta_{20}}{C_{b_{min}}}$$
(25)

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

$$\Delta\theta_{min} = \frac{C_{b10}\Delta\theta_{10} + C_{b20}\Delta\theta_{20}}{C_{b_{max}}}$$
(26)

To fully explore the control space, all  $\Delta t_t$  values that are between the minimum and maximum orbit life and satisfy the equation

$$\Delta t_t = \Delta t_{t_i} + \frac{2\pi}{\omega_e} n \tag{27}$$

should be tested with each  $\Delta \theta_t$  value in the range given by Eq. 23 and Eq. 24. Note that the maximum and minimum orbit life can be calculated by substituting  $\Delta t_{10}$  for  $\Delta \theta_{10}$  and  $\Delta t_{20}$  for  $\Delta \theta_{20}$  in Eqs. 25 and 26.

For each tested combination of  $\Delta t_t$  and  $\Delta \theta_t$ , 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 achievable orbit lifetimes for the desired  $\Delta \theta_t$ , and  $\Delta t_{t_{des}}$  is the desired  $\Delta t_t$ :

• If  $\Delta t_{t_{des}}$  cannot be achieved for the desired  $\Delta \theta_t$ 

$$t_c = -min\left(\left|\Delta t_{t_{min}} - \Delta t_{t_{des}}\right|, \left|\Delta t_{t_{max}} - \Delta t_{t_{des}}\right|\right)$$
(28)

• If  $\Delta t_{t_{des}}$  can be achieved for desired  $\Delta \theta_t$ 

$$t_c = \min\left(\left|\Delta t_{t_{min}} - \Delta t_{t_{des}}\right|, \left|\Delta t_{t_{max}} - \Delta t_{t_{des}}\right|\right)$$
(29)

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

$$t_{s_{max}} = t_{10} + \frac{C_{b20}\Delta t_{20}}{C_{b10}}$$
(30)

The  $\Delta t_1$ ,  $\Delta t_2$ ,  $\Delta \theta_1$ , and  $\Delta \theta_2$  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. 1 and calculating the time and change in true anomaly in each phase using Eq. 2 and Eq. 3. These newly calculated  $\Delta t$  and  $\Delta \theta$  values can be used directly 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. Back-Stepping Method

With the receding horizon strategy, a percentage of the beginning of the trajectory propagated with the analytically calculated set of control parameters is retained and used for the guidance. This makes up the first  $t_g$  seconds of 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 results 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. In this implementation, such cases were handled by reducing  $t_g$  by a factor of two and continuing to do this until there was sufficient controllability in the remainder of the numerical trajectory or a maximum number of such "back-steps" was 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 work,  $t_g$  is initially set to 10% of the remaining orbit life, each back-step reduces  $t_g$  by a factor of two, and a maximum of two back-steps are allowed.

#### 3.4. 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 its life 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.

If precise control over orbit lifetime is not required, once the  $\Delta\theta_d$  required to achieve proper latitude targeting has been calculated, the change in  $C_b$  needed to achieve this can be calculated based on the initial  $\Delta\theta_i$  that occurs between  $t_{mod}$  and the terminal point. The ratio of the initial to the required  $\Delta\theta$  during this period is given by

$$r_{\theta} = \frac{\Delta \theta_i}{\Delta \theta_i + \Delta \theta_d} \tag{31}$$

To achieve the desired  $\Delta \theta_d$ , all  $C_b$  values after  $t_{mod}$  must be multiplied by  $r_{\theta}$  and  $t_{swap}$  must be adjusted using Eq. 13 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 receding 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  $\Delta \theta_d$ . To do this,  $r_{\theta}$ can be calculated as before using Eq. 31 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  $C_{b10}$ ,  $C_{b20}$ ,  $\Delta\theta_{10}$ ,  $\Delta\theta_{20}$ ,  $\Delta t_{10}$ ,  $\Delta t_{20}$ after multiplication by  $r_{\theta}$ . Note that if only one  $C_b$  is maintained between  $t_{mod}$  and  $t_{term}$ , then  $\Delta t_{10} = \Delta \theta_{10} = 0$  and  $C_{b10} = C_{b20}$ . If  $C_{b10} > C_{b_{max}}$ ,  $C_{b1}$  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  $\Delta \theta_d$  is maintained.  $t_{swap}$  can then also be modified according to Eq. 13 to ensure that the swap point occurs at the same semi major axis. If both  $C_{b1}$  and  $C_{b2}$  exceed  $C_{b_{max}}$ , then both  $C_b$  values should be reduced to  $C_{b_{max}}$  to achieve as close to the desired  $\Delta \theta_d$  as possible. If  $C_{b20} > C_{b_{max}}$ , then  $C_{b2}$  must be reduced to  $C_{b_{max}}$  and  $C_{b1}$  increased to ensure that the desired  $\Delta \theta_d$  is achieved. A similar procedure applies if either  $C_b$  value is below  $C_{b_{min}}$ .

### 4. Guidance Tracking Algorithm

Due to uncertainties in the drag force acting on the spacecraft, 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.

#### 4.1. Schweighart Sedwick Relative Motion Dynamics for Feedback Control

The Schweighart Sedwick (SS) equations of relative motion [24] 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 [26] as shown in Fig. 4. 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 ECI frame are

$${}^{E}\mathbf{\hat{i}} = \frac{{}^{E}\mathbf{r}_{g}}{r_{g}}$$
(32)

$${}^{E}\mathbf{\hat{k}} = \frac{{}^{E}\mathbf{r_{g}} \times {}^{E}\mathbf{v_{g}}}{\left|{}^{E}\mathbf{r_{g}} \times {}^{E}\mathbf{v_{g}}\right|}$$
(33)

$${}^{E}\hat{\mathbf{j}} = {}^{E}\hat{\mathbf{k}} \times {}^{E}\hat{\mathbf{i}}$$
(34)

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

$$R_{E2L} = \begin{bmatrix} E_{\mathbf{i}}^T \\ E_{\mathbf{j}}^T \\ E_{\mathbf{k}}^T \end{bmatrix}$$
(35)

The position and velocity of the spacecraft relative to the guidance as seen by an observer in the LVLH frame are given by [26]

$$\delta \mathbf{r} = \mathbf{r}_{sc} - \mathbf{r}_{g} \tag{36}$$

$$\delta \mathbf{v} = \mathbf{v}_{sc} - \mathbf{v}_{g} - \left(\frac{\mathbf{r}_{g} \times \mathbf{v}_{g}}{r_{g}^{2}}\right) \times \delta \mathbf{r}$$
(37)

Note that the subscript "sc" denotes the spacecraft while "g" denotes the guidance. If the vectors used to calculate Eqs. 36-37 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, the equations of relative motion can be linearized in a form known as the Schweighart Sedwick equations. 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



Figure 4: Local Vertical Local Horizontal (LVLH) Frame [26]

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  $\delta \ddot{y}$  due to a difference in the  $C_b$  between the spacecraft and the guidance, the SS linearization can be written as [24]

$$\begin{bmatrix} \delta \dot{x} \\ \delta \dot{y} \\ \delta \ddot{y} \\ \delta \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ b & 0 & 0 & d \\ 0 & 0 & -d & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta \dot{x} \\ \delta \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\rho v_g^2 \end{bmatrix} \Delta C_b$$
(38)

where

$$\Delta C_b = \left( C_{b_{sc}} - C_{b_g} \right) \tag{39}$$

$$n = \sqrt{\frac{\mu}{a^3}}, c = \sqrt{1 + \frac{3J_2R_e^2}{8a^2}} \left[1 + 3\cos(2i)\right], d = 2nc, b = \left(5c^2 - 2\right)n^2$$
(40)

## 4.2. LQR Control for Guidance Tracking

With the dynamics of the relative motion between the spacecraft and the guidance given by Eq. 38 in the classic state-space form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{41}$$

it is possible to use a LQR [23] control approach to drive the relative position and velocity to zero. An LQR controller derives the gain K to yield the feedback control law

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \tag{42}$$

that drives the state to zero and minimizes the cost functional

$$J = \int_0^\infty \left( \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right) dt$$
(43)

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

$$\Delta C_b = -\begin{bmatrix} K_1 & K_2 & K_3 & K_4 \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \\ \delta \dot{x} \\ \delta \dot{y} \end{bmatrix}$$
(44)

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  $t \to \infty$ . However, the practical performance of the controller is heavily dependent on **Q** and **R** which weight the relative importance of driving the state to zero as fast as possible and executing minimal control effort. In many cases, **Q** and **R** are selected through trial and error, but for this problem there is a rigorous way to define these matrices. Because along-track error is far greater than radial error in general, radial error is considered only in terms of its contribution  $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ 

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

With **Q** fixed, **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  $\Delta r_{sat}$  as the desired  $\delta y$  at which the commanded change in ballistic coefficient will be equal to  $C_{b_{max}} - C_{b_{min}}$ . That is, the controller will be guaranteed to saturate at  $\delta y = \Delta r_{sat}$ . The LQR gain **K** can first be computed using an arbitrary initial **R** (**R** = 10,000 used in this work) and then recomputed after updating **R** based on the initially obtained **K** to enforce controller saturation at  $\delta y = \Delta r_{sat}$ . The equation to update **R** is

$$\mathbf{R}_{new} = \mathbf{R}_0 \left( \frac{C_{b_{max}} - C_{b_{min}}}{\mathbf{K}_0 \begin{bmatrix} 0 & \Delta r_{sat} & 0 & 0 \end{bmatrix}^T} \right)^2$$
(45)

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

Finally, the **A** and **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, **K** can be recomputed using the LQR strategy with new **A** and **B** matrices whenever the current atmospheric density ( $\rho_{new}$ ) differs from the density used to compute the previous gain ( $\rho_{old}$ ) by a factor of p or more where

$$p = \frac{\rho_{new}}{\rho_{old}} \tag{46}$$

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  $C_{b_{min}}$ , the desired  $C_b$  is simply set to  $C_{b_{min}}$ . Similarly, if the desired  $C_b$  is greater than  $C_{b_{max}}$ , then  $C_b$  is set to  $C_{b_{max}}$ . 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. 44 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 four minutes would be required to go from  $C_{b_{min}}$  to  $C_{b_{max}}$  is made since this is an upper bound for the deployment times of current retractable drag devices [17, 30]. 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.

#### 5. Extended Kalman Filter for Relative Orbit Determination and Noise Filtering

Spacecraft utilizing the guidance tracking algorithm would likely receive ECI position and velocity measurements from a GPS unit which can be converted to relative position and velocity using the procedure in Section 4.1. The piNAV-L1 [31] is a popular GPS unit for small satellites and the manufacturer-specified position and velocity errors are approximately Gaussian with a standard deviation of 5 m and 5 cm/s respectively with bias errors that can be up to 5 m in position and 5 cm/s in velocity. The piNAV-L1 has a measurement frequency of 1 Hz. Other GPS units such as the NovAtel OEM615 have similar performance. While this level of error would likely be insignificant for maneuvers performed with large chemical engines, drag-based orbit control algorithms are much more sensitive. The low magnitude of aerodynamic drag force in space means that any orbit changes take a long time to achieve and any errors take a long time to correct. This means that the controller must react vigorously to any tracking error in order to prevent excessive drift, making the controller much more sensitive

to noise (a fact ignored by much existing literature on aerodynamic orbit control). When tracking simulations were run with unfiltered noisy simulated GPS measurements, the  $C_b$  commands became erratic with the actuator running over 70% of the time, as opposed to around 1-3% of the time during the noise free case. Fortunately, the structure of the relative position and velocity needed for the guidance tracking algorithm lends itself to the implementation of an EKF [25] that provides a more accurate estimate of the spacecraft state relative to the guidance than a direct filtering of the ECI position and velocity.

A Kalman filter has two stages, a predict stage and an update stage. In the predict stage, an a-priori estimate of the state and estimation error covariance matrix is first made based on the state and covariance estimate at the previous time point and knowledge of the system dynamics. A crucial part of the predict phase is the state transition matrix  $\Phi$  which gives the new state when pre-multiplied by the previous state. Since the linearized dynamics are already known, the matrix exponential of the A matrix from Eq. 38 can be utilized to calculate  $\Phi$  as

$$\Phi = e^{\mathbf{A}\Delta t} \tag{47}$$

where  $\Delta t$  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  $\Phi$ . 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  $\Delta t$ , 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  $\Phi$ . The state and covariance estimates for the predict stage are thus

$$\mathbf{x}_{i}^{-} = f(t_{i-1}, t_{i}, \mathbf{x}_{i-1}^{+}) P_{i}^{-} = \Phi_{i} P_{i-1}^{+} \Phi_{i} + Q$$
(48)

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  $z_i$ . The update stage is described in Ref. [25] as follows

$$K_{i} = P_{i}G^{T} \left( GP_{i}^{-}G^{T} + W \right)^{-1}$$

$$\mathbf{x}_{i}^{+} = \mathbf{x}_{i}^{-} + K_{i} \left( \mathbf{z}_{i} - G\mathbf{x}_{i}^{-} \right)$$

$$P_{i}^{+} = (I - K_{i}G)P_{i}\Lambda$$
(49)

Where *W* is the measurement noise covariance matrix,  $\Lambda$  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 [32]), and *G* specifies the linear mapping between the measurement and the state as

$$\mathbf{z} = G\mathbf{x} \tag{50}$$

Note that in this scenario z is the raw GPS measurement converted to in-plane relative position and velocity and so *G* is a 4x4 identity matrix.

### 6. 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.

## 6.1. 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 4 using the procedure from Montenbruck's book [33] and the EGM2008 gravity model [34]. Atmospheric density was given by the NRLMSISE-00 model [35] with historic F10.7 and Ap indices [36]. 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.

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. 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

$$k_{err} = k_0 + \sum_{i=1}^{3} k_i \sin(\frac{2\pi}{T_i}t - \phi_i)$$
(51)

The *T* values were set to  $T_1 = 26$  days,  $T_2 = 1$  day,  $T_3 = 5400s$  based on observed density variations on real satellite missions [37, 38, 39]. These corresponded to the synodic period (sun rotation), Earth day, and approximate orbital period.  $\phi$  values were randomly selected from a uniform distribution between 0 and  $2\pi$ .  $k_0$  values were randomly selected from a uniform distribution between .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 [39]. 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 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.

### 6.2. 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,



Figure 5: piNAV-L1 Simulated Position Errors for ISS Orbit [31]

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 [31]. 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 Figure 5 [31]. As Figure 5 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

$$\Delta \mathbf{r}_{bias} = \begin{bmatrix} .001 \\ -.005 \\ .002 \end{bmatrix} \sin\left(\frac{2\pi t}{5400}\right) m, \Delta \mathbf{v}_{bias} = \begin{bmatrix} .00005 \\ .00005 \\ .000025 \end{bmatrix} \sin\left(\frac{2\pi t}{5400}\right) m/s$$
(52)

## 6.3. 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.

### 6.3.1. 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 with randomized initial conditions:

- epoch = January 24, 2004, 6:48:29.48 UTC
- initial osculating orbital elements: (a = 6715.97km, e = .000471, Ω = 214.24°, ω = .1790°, θ = 359.77°, i = 70.67°)
- $C_{b_{min}} = .0059m^2/kg$ ,  $C_{b_{max}} = .0386m^2/kg$ ,  $C_{b_{term}} = .0222m^2/kg$
- target de-orbit location: -54.54° latitude, 160.85° longitude at a 70 km geocentric altitude
- Targeting set to begin at 6500 seconds of longitude controllability as defined Ref. [21].

The guidance tracking algorithm was run with perfect knowledge of the state, a 5% actuator dead-band, the assumption that the drag device takes four minutes to fully deploy, and a constant bias error of .7 (nominal drag force values all multiplied by .7). Tracking was continued down to a geodetic altitude of 90 km. As shown in Figure 6, 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 .13% of the orbit lifetime to produce the indicated  $C_b$  fluctuations.



Figure 6: Tracking Position Error and C<sub>b</sub> over Time with Drag Bias Error of .7

#### 6.3.2. Tracker Performance under Actuator Saturation

If the along-track separation between the satellite and the guidance is greater than  $\Delta r_{sat}$ , 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. Figure 7 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 .02 radians (about 132 km) and a density bias of .7 as before. To further verify the operation of the tracker under saturation, one hundred



Figure 7: Tracking Position Error and C<sub>b</sub> over Time with 132 km Initial Error and Actuator Saturation

tracking simulations were conducted for different guidances with an initial tracking true anomaly error of .02 radians with  $\Delta r_{sat} = 5km$  and in all cases, the tracker was able to return to the guidance.

#### 6.3.3. 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 rerun with only the *J*<sup>2</sup> perturbation considered in the tracking simulation. The result is similar to if the tracking algorithm was used on a real spacecraft with a guidance created considering only *J*<sup>2</sup> gravitational perturbations. As shown in Figure 8, 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. [18] will result in guidances that are not easily trackable. Gravitational perturbations beyond degree and order four were found to be insignificant and not worth the additional computational cost of including in guidance generation.

### 6.3.4. Noisy Guidance Tracking without Filter

As shown in Figure 9, if unfiltered GPS position and velocity measurements are used, the controller will respond erratically in its attempt to track the noise. While the



Figure 8: Tracking Position Error and C<sub>b</sub> over Time with Low Order Gravity Model

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 Figure 9.



Figure 9: Tracking Position Error and C<sub>b</sub> over Time with Noisy GPS Position and Velocity Measurements

#### 6.3.5. 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 Figure 10, 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



Figure 10: Tracking Position Error and  $C_b$  over Time with Noisy State Estimates and EKF

and velocity, because a noise filter cannot remove bias error. The aforementioned scenario was run with zero mean GPS measurement error with only Gaussian noise, and the results are plotted in Figure 11. Figure 11 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.



Figure 11: Tracking Position Error and C<sub>b</sub> over Time with Purely Gaussian Noise and EKF

### 6.3.6. Noisy Guidance Tracking with Complete Drag Error and EKF

Figure 12 shows the position error and 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.



Figure 12: Tracking Position Error and C<sub>b</sub> over Time with Drag Uncertainties and GPS Measurement Noise

### 6.4. Monte Carlo Simulation Results

One thousand simulations of the guidance generation algorithm were conducted with randomized initial conditions as shown in Table 2. The mean guidance error was 24.3 *km* with a standard deviation of 49.0 *km* leading to a 99% confidence interval for the expected average guidance error of 20.3 *km* to 28.3 *km*. All guidance errors were below 750 *km* and are shown in Figure 13. After running guidance tracking algorithms on all generated guidances down to a geodetic altitude of 90 km, all tracking errors were less than 10 km with an average error of 3.4 *km*. Tracking algorithms were run for each guidance with density uncertainties and sensor noise simulated as was done in Section 6.3.6. After simulating guidance tracking down to a geodetic altitude of 90 km, all tracking errors were less than 10 km with an average error of 3.4 *km*. Tracking algorithms were run for each guidance with density uncertainties and sensor noise simulated as was done in Section 6.3.6. After simulating guidance tracking down to a geodetic altitude of 90 km, all tracking errors were less than 10 km with an average error of 3.4 *km*. Results of the tracking simulations are shown in Figure 14.

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 Periapsis	[0, 360] degrees	Uniform
Inclination	[1, 97] degrees	Uniform
Impact Latitude	[min reachable lat +.1,	Uniform
	max reachable lat1]	
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

Table 2: Monte Carlo simulation parameters



Figure 13: Guidance Errors from Monte Carlo Simulations



Figure 14: Maximum Guidance Tracking Errors from Monte Carlo Simulations

### 7. 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 casespecific simulations are presented to validate 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. 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).

### References

- R. P. Patera, W. H. Ailor, The realities of reentry disposal, in: Proceedings of the AAS/AIAA Space Flight Mechanics Meeting, American Astronautical Society/AIAA, Monterey, California, 1998, pp. 9–11.
- [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, in: Proceedings of the 14th Annual AIAA/USU Conference on Small Satellites, Logan, UT.
- [3] C. L. Leonard, W. M. Hollister, E. V. Bergmann, Orbital Formation Keeping with Differential Drag, Journal of Guidance, Control, and Dynamics 12 (1989) 108–113.
- [4] B. S. Kumar, A. Ng, K. Yoshihara, A. D. Ruiter, Differential Drag as a Means of Spacecraft Formation Control, IEEE Transactions on Aerospace and Electronic Systems 47 (2011) 1125–1135.
- [5] L. Mazal, D. Prez, R. Bevilacqua, F. Curti, Spacecraft Rendezvous by Differential Drag Under Uncertainties, Journal of Guidance, Control, and Dynamics 39 (2016) 1721–1733.
- [6] D. Pérez, R. Bevilacqua, Lyapunov-Based Spacecraft Rendezvous Maneuvers using Differential Drag, in: AIAA Guidance, Navigation, and Control Conference, p. 6630.
- [7] D. Pérez, R. Bevilacqua, Differential Drag Spacecraft Rendezvous using an Adaptive Lyapunov Control Strategy, Acta Astronautica 83 (2013) 196–207.
- [8] D. Pérez, R. Bevilacqua, Differential Drag-Based Reference Trajectories for Spacecraft Relative Maneuvering Using Density Forecast, Journal of Spacecraft and Rockets 53 (2016) 234–239.
- [9] S. R. Omar, J. M. Wersinger, Satellite Formation Control using Differential Drag, in: Proceedings of 53rd AIAA Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics, Kissimmee, FL, 2014.
- [10] M. Pastorelli, R. Bevilacqua, S. Pastorelli, Differential-Drag-based Roto-Translational Control for Propellant-less Spacecraft, Acta Astronautica 114 (2015) 6–21.

- [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 (2015). ArXiv: 1509.03270.
- [12] E. Lokcu, R. L. Ash, T. A. Force, A De-Orbit System Design for CubeSat Payloads, in: 2011 5th International Conference on Recent Advances in Space Technologies (RAST), 9-11 June 2011, pp. 470–4.
- [13] B. Cotton, On-Orbit Results from CanX-7 Drag Sail De-Orbit Mission, in: Proceedings of the 31st Annual AIAA/USU Conference on Small Satellites, Logan, Utah.
- P. Harkness, M. McRobb, P. Ltzkendorf, R. Milligan, A. Feeney, C. Clark, Development Status of AEOLDOS A Deorbit Module for Small Satellites, Advances in Space Research 54 (2014) 82–91.
   MACA Presses for Limiting Orbital Debria. Technical Parent NACA STD 9710 144, 2019.
- [15] NASA, Process for Limiting Orbital Debris, Technical Report NASA-STD-8719.14A, 2012.
- [16] Z. Wu, R. Hu, X. Qu, X. Wang, Z. Wu, Space Debris Reentry Analysis Methods and Tools, Chinese Journal of Aeronautics 24 (2011) 387–395.
- [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, in: Proceedings of the 5th Nano-Satellite Symposium, Tokyo, Japan.
- [18] J. Virgili, P. Roberts, Atmospheric Interface Reentry Point Targeting Using Aerodynamic Drag Control, Journal of Guidance, Control, and Dynamics 38 (2015) 1–11.
- [19] 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, in: AIAA Atmospheric Flight Mechanics Conference, p. 0467.
- [20] S. R. Omar, R. Bevilacqua, D. Guglielmo, L. Fineberg, J. Treptow, S. Clark, Y. Johnson, Spacecraft Deorbit Point Targeting Using Aerodynamic Drag, Journal of Guidance, Control, and Dynamics (2017) 1–7.
- [21] S. R. Omar, R. Bevilacqua, Spacecraft De-Orbit Point Targeting using Aerodynamic Drag, American Institute of Aeronautics and Astronautics, 2017.
- [22] MathWorks, Linear-Quadratic Regulator (LQR) Design MATLAB LQR, 2017.
- [23] G. Franklin, J. Powell, A. Emami-Naeini, Feedback Control of Dynamic Systems, Prentice Hall, Upper Saddle River, NJ, 4 edition, 2002.
- [24] S. A. Schweighart, R. J. Sedwick, High-Fidelity Linearized J Model for Satellite Formation Flight, Journal of Guidance, Control, and Dynamics 25 (2002) 1073–1080.
- [25] A. Kelly, A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles, Technical Report, The Robotics Institute, Carnegie Melon University, Pittsburgh, PA, 1994.
- [26] H. Curtis, Orbital Mechanics for Engineering Students, Elsevier, Burlington, MA, 2 edition, 2009.
- [27] W. H. Kwon, S. H. Han, Receding Horizon Control: Model Predictive Control for State Models, Springer Science & Business Media, 2006. Google-Books-ID: ITSKhlKy2u8C.
- [28] D. Bau III, L. Trefethen, Numerical Linear Algebra, Society for Industrial and Applied Mathematics, Philadelphia, 1997.
- [29] N. Nise, Control Systems Engineering, John Wiley and Sons, 5 edition, 2008.
- [30] D. Guglielmo, S. R. Omar, R. Bevilacqua, L. Fineberg, J. Treptow, B. Poffenberger, Y. Johnson, Drag De-Orbit Device (D3): A Retractable Device for CubeSat Attitude and Orbit Control using Aerodynamic Forces, Orlando, FL.
- [31] P. Kovár, piNAV L1GPS receiver for small satellites, Gyroscopy and Navigation 8 (2017) 159–164.
- [32] S. Slojkowski, J. Lowe, J. Woodburn, Orbit Determination for the Lunar Reconnaissance Orbiter Using an Extended Kalman Filter, Technical Report, NASA Goddard Spaceflight Center, 2015.
- [33] O. Montenbruck, E. Gill, Satellite Orbits, Springer Berlin Heidelberg, 1 edition, 2005.
- [34] N. Pavlis, S. Holmes, S. Kenyon, J. Factor, An Earth Gravitation Model to Degree 2160: EGM2008, 2008.
- [35] D. Vallado, Fundamentals of Astrodynamics and Applications, Microcosm Press, Hawthorne, CA, 4 edition, 2013.
- [36] N. O. a. A. Administration, USAF 45-Day Ap and F10.7cm Flux Forecast, 2017.
- [37] D. A. Vallado, D. Finkleman, A Critical Assessment of Satellite Drag and Atmospheric Density Modeling, Acta Astronautica 95 (2014) 141–165.
- [38] F. Marcos, B. Bowman, R. Sheehan, Accuracy of Earths Thermospheric Neutral Density Models, in: Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Keystone, CO.
- [39] J. L. Lean, J. M. Picone, J. T. Emmert, G. Moore, Thermospheric Densities Derived from Spacecraft Orbits: Application to the Starshine Satellites, Journal of Geophysical Research: Space Physics 111 (2006) A04301.