(Preprint) AAS 26-783

REMOTE GUIDANCE OF A MOTHERSHIP-DAUGHTERSHIP FORMATION FOR ASTEROID EXPLORATION

Logan Feld,* Shyam Bhaskaran,† Andrew French,‡ Joshua Lyzhoft,§ Koki Ho¶

This work presents an application of Model Predictive Control to maintain long-term stability for a three-satellite formation around the asteroid Apophis via remote guidance. The mothership and two daughterships perform tomographic characterization with a bistatic radar. The mothership estimates the states of all three spacecraft and determines the necessary control to maintain the desired antipodal daughtership states. An Extended Kalman Filter estimates the mothership states. A Batch Least Squares algorithm estimates the daughtership states and performs maneuver reconstruction for the mothership-administered thrust commands under thruster and gravitational uncertainty. The formulation maintains the antipodal angle with growing instability amidst these uncertainties.

INTRODUCTION

With the Near-Earth Asteroid (NEA) 99942 Apophis making its closest approach of Earth in April 2029,[1] work is necessary to determine the feasibility of long-term prior characterization of Apophis to enable researchers, scientists, and engineers to seize the rare opportunity to study the characteristics of these unique celestial bodies. Several missions have been proposed and are already in the development stages[2–4] with various mission goals, including sample collection and proximity operations. These goals can not be met if characterization of Apophis is not performed prior in order to construct accurate models for dynamics propagation and internal structure. One method to characterize an asteroid is with tomography, a noninvasive imaging technique utilized to create cross-sectional slices of an object. NASA Jet Propulsion Lab (JPL) proposed The Distributed Radar Observations of Interior Distributions (DROID) mission which employs a mothership for the deployment of two CubeSats on opposite sides of Apophis, denoted "antipodal" with respect to Apophis, equipped with a bistatic radar to perform the tomographic characterization.[5] Many past missions have flown formations of satellites for characterization around larger celestial bodies, like NASA's GRACE and GRAIL missions,[6,7] but no missions to date have flown that deploy controlled formations of satellites around a small body. While work has been done to verify that maintaining an orbit around Apophis during its closest flyby is feasible,[8] additional work is still necessary to determine if it is possible to maintain a formation of satellites with the tomographic imaging CubeSats on antipodal orbits, a problem this work aims to address.

Several challenges present themselves when trying to maintain a formation around Apophis, namely state estimation and orbit determination, gravity and thruster modeling, and which spacecraft administers the thrust

*Graduate Student Researcher, School of Aerospace Engineering, Georgia Institute of Technology, Ferst Dr., Atlanta, 30332, GA, USA.
†Principal Engineer, Outer Planet Navigation Group, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.
‡Mission Design & Navigation Engineer, Outer Planets Navigation Group, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.
§Aerospace Engineer, Navigation & Mission Design Branch, NASA Goddard Space Flight Center, Greenbelt Rd., Greenbelt, 20771, MD, USA.
¶Dutton-Ducoffe Professor; Associate Professor, School of Aerospace Engineering, Georgia Institute of Technology, Ferst Dr., Atlanta, 30332, GA, USA.

commands. For state estimation, all three spacecraft need to have minimal uncertainty on their states for control decisions to be made with confidence. Further, because of the microgravity environment, the gravity model and other smaller forces modeled have significant impact on the trajectory of the spacecraft. In practice, having a perfect dynamics model is impossible; the system must be able to handle uncertainties in the dynamics model. Similarly, because of the microgravity environment, ΔV's as low as millimeter-per-second scale are needed for precise control and station-keeping, a magnitude thrusters aren't suited to perfectly match. Thrust uncertainty is a critical component the system must be able to estimate and address. Lastly, the choice of having the CubeSats make their own decisions or have the mothership administer commands to the CubeSats is an important aspect to consider.

In order to maintain the satellite formation, a robust combination of an Extended Kalman Filter (EKF) and a Batch Least Squares (BLS) algorithm is implemented to fully estimate the three-spacecraft formation. The EKF and BLS algorithms are well-understood for nonlinear dynamics and are frequently utilized in the astrodynamics community; EKFs were flown on board OSIRIS-REx[9] and Artemis 1[10] and BLS is often used for orbit determination of satellites in LEO and other orbit regimes.[11,12] Because Apophis itself blocks direct line-of-sight (LoS) between the two CubeSats, henceforth denoted the "daughterships", the mothership is required to estimate the daughtership states and also know her own state prior to collecting daughtership measurements. As the motership states are estimated in a separate filter from the daughterships, the daughtership residuals The mothership can estimate her own state by taking simulated images of the surface of Apophis and processing the computed pixel coordinates of known feature locations in the image within the EKF.[13] If an adequate amount of predetermined Apophis surface features are mapped prior to the mission, a task typically completed with stereophotoclinometry (SPC),[14] the mothership estimated state can be updated each time step. With the mothership state estimated, the daughtership estimated states are updated using the BLS algorithm[12] that accumulates both range and pixel coordinate observations from the mothership to each daughtership every time step. The pixel coordinates assume use of the same camera the mothership used to estimate her own state. The range measurement can be obtained with a radiometric crosslink. Once a user-specified amount of new range and image coordinate observations are accumulated at equal time intervals, the BLS is run to update the trajectories of each daughtership by estimating their initial conditions along their trajectories within the current BLS measurement window and re-propagating from the new best initial estimate.

Once the entire system's estimated states are updated, an application of Model Predictive Control (MPC) is employed to determine if either or both daughterships must administer control to maintain the anitpodal states necessary for the bistatic radar to return usable data amidst gravitational and thruster uncertainties. MPC is a control algorithm that optimizes a control input across a future time horizon where the state of the system is predicted across said horizon based on a chosen set of control actions.[15] The cost function being minimized by MPC can be designed such that the estimated states deviating from specific mission goals across the prediction horizon are penalized. This work builds upon previously completed work that performed an extensive analysis of six different cost function designs for a time-of-flight free MPC formulation for single-spacecraft asteroid landings.[16] In this application, the cost function design is extended to consider multiple spacecraft simultaneously and incorporate considerations beyond fuel and individual reference trajectory proximity; it now additionally attempts to maintain both daughterships on comparable relative antipodal orbit geometries from their states estimated by the mothership, necessitating the online computation of relative geometries and osculating orbital elements within the cost function. The MPC minimizing the cost with respect to the estimated states means that the control outputs are near-optimal, not fully optimal, since there will never be perfect state knowledge. The MPC's prediction of future states from control input enables robustness to state and control uncertainties as the system can react to deviations in state or control in real time. This allows the implementation of gravitational and thruster uncertainties to more realistically simulate the Apophis environment. So, the MPC will determine the near-optimal control action for each daughtership across the predicition horizon based on the formation's estimated states. To simulate dynamics model uncertainty, the MPC uses a lower resolution gravity model while the true dynamics propagation outside the MPC is propagated with the most up-to-date polyhedral shape model available.[17] The near-optimal ΔV will be applied exactly to the estimated state whereas an additional erroneous ΔV will be applied to the true state in order to simulate thruster uncertainty. If there are any ΔV's, the next BLS run will also estimate the actual ΔV as well as the initial conditions of the daughterships within the current BLS measurement window.

This work chose to have the mothership remotely guide and administer control decisions to each daughtership rather than have each daughtership compute their individual control decisions to minimize the need for data transfer, to reduce mission cost, and to improve robustness to deployment failure. This decision necessitates only that the mothership be able to send commands one-way to each daughtership. The mothership computing control decisions for the entire system and sending them to each respective daughtership ensures that control decisions don't counteract each other, a concern if control decisions were made by individual daughterships. In addition, this presents a cost savings for the mission, as the computational hardware for determining the control decisions only needs to be present on the mothership rather than multiple daughterships. Further, should anything go wrong to one of the daughterships during deployment from the mothership where control of that daughtership is no longer possible, the MPC cost function optimized on the mothership can be easily adjusted to account for only one daughtership being controllable and some data can still be collected while the defunct daughtership remains in orbit around Apophis.

METHODS

This section presents the modeling and algorithmic frameworks necessary for each component in this model predictive control application for remote guided asteroid exploration. First, the dynamics model utilized for state propagation in the EKF, BLS, and reference trajectories will be outlined. Next, the measurement model and update equations for the EKF will be outlined for mothership state estimation. This will be followed by the formulation of the measurement model and full BLS algorithm for daughtership state estimation. Lastly, the MPC cost function will be formulated and the full MPC algorithm will be outlined. Figure 1 shows the full mission architecture with all the components listed above visualized as a flowchart.

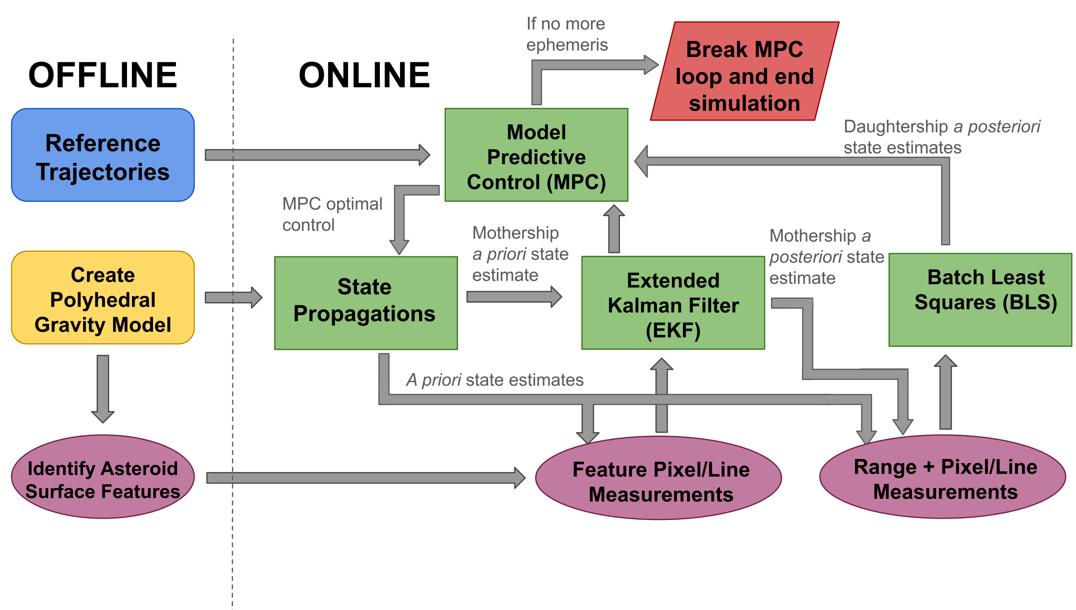


Figure 1. Full Apophis Mission Flowchart

The Dynamics Model

To support state propagation and estimation, the dynamics model dictating the spacecraft's motion incorporates several dominant accelerations on the spacecraft, including a high-fidelity gravity model, solar radiation pressure (SRP), and third body effects from the Sun, Jupiter, and the Earth-Moon barycenter. Each of these forces are incorporated using the equations outlined below following the definition of the state vector and the ordinary differential equation (ODE) dictating the dynamics.

The state vector $\boldsymbol{s}$ is the standard 7-D vector incorporating inertial x, y, z spacecraft position and velocity as well as the natural log of the spacecraft's mass, denoted q. The origin of the inertial frame is set to be the center of mass (CoM) of Apophis. Spacecraft mass is stored in natural log form so that mass flow can be computed linearly, like position and velocity, with the rocket equation.

$$\boldsymbol{s} = [x, y, z, v_x, v_y, v_z, q]^T = [\boldsymbol{r}, \boldsymbol{v}, q]^T \tag{1}$$

The dynamics are dictated by the classic first-order ODE approach, as seen in Equation 2.

$$\dot{\boldsymbol{s}} = [\boldsymbol{v}, \boldsymbol{a}(\boldsymbol{s}, t), \dot{q}]^T \tag{2}$$

where $\boldsymbol{a}(\boldsymbol{s})$ are the accelerations from gravity, SRP, and third body effects and $\dot{q}$ is the mass flow rate shown in Equation 3.

$$\dot{q} = -\|\boldsymbol{\Delta V}\|/v_{\text{exhaust}} \tag{3}$$

The first acceleration considered in the dynamics model is the gravitational acceleration which is modeled with the polyhedral gravity model for this work. The equation to compute the gravitational acceleration from the polyhedral shape model's facets and edges, denoted $\boldsymbol{a}(\boldsymbol{s})_{\text{poly}}$, is omitted from this work, but can be found in the paper by Werner and Scheeres.[18] Because the acceleration from polyhedral gravity is in the body-fixed frame, it is necessary to transform it into the inertial frame to match the state and other forces modeled. While Apophis is a non-principal axis rotator, this work assumes that Apophis rotates solely around its Z-axis to make the problem more tractable. Apophis rotates with an angular velocity of 0.00005713025 rad/sec or 30.55 hr/rev,[19] making the transformation a simple rotation matrix around the Z-axis, $\boldsymbol{R}_z(t)$.

The next acceleration considered in the dynamics model is the effects of SRP. Acceleration from SRP is a consequence of photons from the Sun impacting the spacecraft and exerting force. The acceleration from SRP in the inertial frame is as follows.[20]

$$\boldsymbol{a}(\boldsymbol{s}, t)_{\text{SRP}} = -P_{\odot} C_R \frac{A}{m} \frac{\boldsymbol{r}_{\odot}(t)}{r_{\odot}(t)^3} r^2_{\odot \rightarrow \oplus} \tag{4}$$

where $P_{\odot} \approx 4.56 * 10^{-6} N/m^2$ is the solar radiation pressure constant, $C_R \in [1, 2]$ is the coefficient of reflection, A is the spacecraft's surface area illuminated by the Sun, $m = e^q$ is the mass of the spacecraft, $\boldsymbol{r}_{\odot}$ is the position vector from the spacecraft to the Sun (with $r_{\odot}$ as its magnitude), and $r_{\odot \rightarrow \oplus}$ is the conversion from meters to astronomical units, au.

The last set of accelerations considered in the dynamics model are the third body accelerations. Third body effects from the Sun, Jupiter, and the Earth-Moon barycenter (EMB) are considered. The equation for the acceleration on the spacecraft from a third body in the inertial frame is as follows.[20]

$$\boldsymbol{a}(\boldsymbol{s}, t)_{\text{3rd}} = \mu_{\text{3rd}} \Big(\frac{\boldsymbol{r}_{\text{3rd}}(t) - \boldsymbol{r}(t)}{\|\boldsymbol{r}_{\text{3rd}}(t) - \boldsymbol{r}(t)\|^3} - \frac{\boldsymbol{r}_{\text{3rd}}(t)}{\|\boldsymbol{r}_{\text{3rd}}(t)\|^3} \Big) \tag{5}$$

where μ_{3rd} is the gravitational parameter of the third body, $\boldsymbol{r}_{\text{3rd}}(t)$ is the vector from Apophis to the third body, and $\boldsymbol{r}$ is the state vector position from Apophis to the spacecraft. The ephemerides for the Sun, Jupiter, and EMB contained within $\boldsymbol{r}_{\text{3rd}}(t)$ are obtained through JPL's Horizons web app within the Solar System Dynamics group.[21]

The ODE uses the summation of these acceleration components, $\boldsymbol{a}(\boldsymbol{s}, t)$, in the inertial frame.

$$\boldsymbol{a}(\boldsymbol{s},t) = \boldsymbol{R}_z(t)\boldsymbol{a}(\boldsymbol{s})_{\text{poly}} + \boldsymbol{a}(\boldsymbol{s},t)_{\text{SRP}} + \boldsymbol{a}(\boldsymbol{s},t)_{\text{3rd,Sun}} + \boldsymbol{a}(\boldsymbol{s},t)_{\text{3rd,Jupiter}} + \boldsymbol{a}(\boldsymbol{s},t)_{\text{3rd,EMB}} \tag{6}$$

With the acceleration outlined, the dynamics model f for time step k utilized in the simulation can be fully formulated as follows, ready to be employed within the EKF, BLS and MPC algorithms.

$$\boldsymbol{s}_{k+1} = f(\boldsymbol{s}_k, t) \tag{7}$$

The Extended Kalman Filter and its Measurement Model

Before control can be administered to either daughtership, the mothership's own state must be estimated with the EKF by taking simulated images of Apophis' surface. While the mothership does not thrust, it is still necessary to estimate the state to correct for any dynamics modeling errors. The EKF algorithm is commonly broken into two distinct steps, the "update" step that propagates the true and estimated states and the "predict" step that updates the state estimate from measurements taken of the system.[13] These steps are outlined below.

Predict

$$\hat{\boldsymbol{s}}_{k|k-1} = f(\hat{\boldsymbol{s}}_{k-1|k-1}, \boldsymbol{u}_{k-1}) \tag{8}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^T + \boldsymbol{Q}_{k-1} \tag{9}$$

Update

$$\boldsymbol{\Delta y} = \boldsymbol{y}_k - h(\hat{\boldsymbol{s}}_{k|k-1}) \tag{10}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{11}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1} (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T \tag{12}$$

$$\boldsymbol{F}_k = \left.\frac{\partial f}{\partial s}\right|_{\hat{\boldsymbol{s}}_{k-1|k-1}, \boldsymbol{u}_k} \tag{13}$$

$$\boldsymbol{H}_k = \left.\frac{\partial h}{\partial s}\right|_{\hat{\boldsymbol{s}}_{k|k-1}} \tag{14}$$

where $\boldsymbol{s}_k$ and $\boldsymbol{u}_k$ represent the state and control vectors at time step k and f is the dynamical model for state propagation from Equation 7. h is the measurement model and will be formulated in the subsequent paragraphs. The hat notation indicates a state estimate. The state covariance is captured in the $\boldsymbol{P}$ matrix, computed using the more stable and robust Joseph formulation.[13] $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the positive-definite state and measurement noise variances. $\boldsymbol{\Delta y}$ are the measurement residuals computed using the observations at the true state, $\boldsymbol{y}_k$. $\boldsymbol{K}_k$ is the Kalman gain. By choosing an observable measurement model, the EKF can improve state estimates under uncertainties.

The mothership leverages the EKF update and predict equations to update her state by utilizing the pixel coordinates of seen features within a simulated image of Apophis' surface with a nadir-pointing pinhole camera model. This necessitates defining what a feature is within the scope of this work and what features within an image are actually seen by the camera model. Figure 2 visualizes the necessary vectors for defining a seen feature in the inertial frame.

Features, in this work, are simply randomly-selected vertices in the polyhedral shape model that are flagged to be features. In practice, features are more than simply vertices; they are arrays of terrain information that can be fit to taken images, meaning they can take many forms.[9] Whether or not a feature is "seen" within a simulated image is determined through the intersection of three vector inequalities presented below.

$$\overrightarrow{\text{meas}} \cdot \overrightarrow{s}_{\text{vis}} \geq \cos(\phi/2) \tag{15}$$

$$\overrightarrow{\text{feat}} \cdot \overrightarrow{s}_{\text{vis}} \leq 0 \tag{16}$$

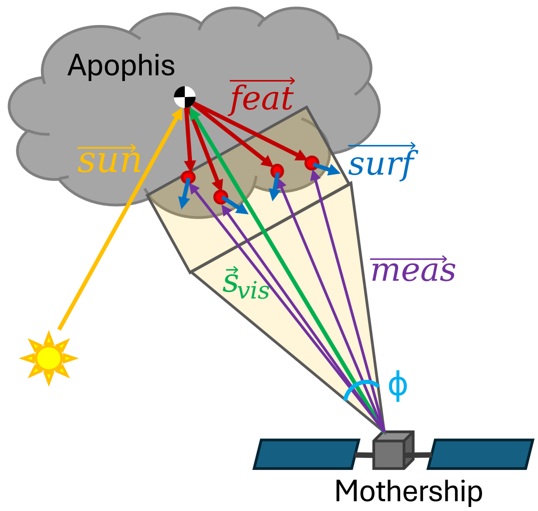


Figure 2. Mothership Feature Vectors

$$\overrightarrow{\mathrm{surf}} \cdot \overrightarrow{\mathrm{sun}} \leq 0 \tag{17}$$

For each epoch in the simulation, a simulated image is taken by the mothership and every pre-identified feature stored on board is processed with the above inequalities. Equation 15 determines whether the position of a feature lies within the camera field of view (FOV), ϕ. Equation 16 checks whether a feature is on the same hemisphere that the mothership is orbiting above. This prevents features on the opposite side of the asteroid, but within the FOV volume, from being flagged as seen. Equation 17 ensures that feature surface normals are illuminated by the Sun and will actually show up within a simulated image. Any feature that satisfies all three inequalities is denoted as "seen" and stored for use in the EKF update step.

As seen feature positions are naturally in the Apophis body-fixed frame, the positions of seen features must first be rotated into the inertial frame with the previously defined $\boldsymbol{R}_z(t)$ Z-axis rotation matrix and then further transformed into the camera frame in order to extract pixel coordinates for the EKF to use as measurements. This is done by constructing a radial, in-track, cross-track (RIC) frame centered at the mothership location for the nadir-pointing camera frame.

$$\vec{\boldsymbol{R}}_{\mathrm{RIC}} = -\frac{\boldsymbol{r}}{\|\boldsymbol{r}\|} \tag{18}$$

$$\vec{\boldsymbol{C}}_{\mathrm{RIC}} = -\frac{\boldsymbol{r} \times \boldsymbol{v}}{\|\boldsymbol{r} \times \boldsymbol{v}\|} \tag{19}$$

$$\vec{\boldsymbol{I}}_{\mathrm{RIC}} = \vec{\boldsymbol{C}}_{\mathrm{RIC}} \times \vec{\boldsymbol{R}}_{\mathrm{RIC}} \tag{20}$$

This allows the assembly of a rotation matrix to go from the inertial frame to the camera frame. Note that this construction of the RIC frame is left-handed; this is done so that the camera pointing direction (i.e., from the mothership to Apophis' COM) is in the positive Z-direction in the camera frame. The rotation matrix from the camera frame to the inertial frame is constructed as follows.

$$\boldsymbol{R}_{\mathrm{Camera}\rightarrow\mathrm{Inertial}} = \begin{bmatrix} \vec{\boldsymbol{I}}_{\mathrm{RIC}} \\ \vec{\boldsymbol{C}}_{\mathrm{RIC}} \\ \vec{\boldsymbol{R}}_{\mathrm{RIC}} \end{bmatrix} \tag{21}$$

As such, the vectors from the mothership to seen features, $\overrightarrow{\text{meas}}_{\text{Body}}$, can be expressed in the camera frame as follows.

$$\overrightarrow{\text{meas}}_{\text{Inertial}} = \boldsymbol{R}_z(t)\overrightarrow{\text{meas}}_{\text{Body}} \tag{22}$$

$$\overrightarrow{\text{meas}}_{\text{Camera}} = \boldsymbol{R}^T_{\text{Camera}\to\text{Inertial}}\overrightarrow{\text{meas}}_{\text{Inertial}} \tag{23}$$

The vectors in the camera frame are then normalized by their camera frame Z-components and converted into pixel coordinates p and l. Conversion to pixel coordinates is done using a simple pinhole camera model.[22] First, the camera calibration matrix, $\boldsymbol{K}_{\text{cam}}$ is constructed.

$$\boldsymbol{K}_{\text{cam}} = \begin{bmatrix} f_x & 0 & w/2 \\ 0 & f_y & h/2 \end{bmatrix} \tag{24}$$

where f_x and f_y are the camera focal lengths in the camera frame x and y directions while w and h are the width and height of the image in pixels. The pixel coordinates are therefore computed as follows.

$$\begin{bmatrix} p \\ l \end{bmatrix} = \boldsymbol{K}_{\text{cam}}\frac{\overrightarrow{\text{meas}}_{\text{Camera}}}{\text{meas}_{\text{Camera,Z}}} + \omega_{\text{pix}} = h_{\text{pix}}(\boldsymbol{s}, t_i) \tag{25}$$

The measurement model h_{pix} takes in the state of the spacecraft and outputs the pixel coordinates of all seen features in the image using the above Equations 18-25. When computing the residuals $\boldsymbol{\Delta y}$ for use in the EKF, observed and computed measurements are necessary. Computed measurements are simply found using the above equations with the estimated state inputted with no ω_{pix} value. Observations $\boldsymbol{y}_k$ are functions of the true state, but have an additional Gaussian zero-mean white noise ω_{pix} added to p and l with covariance $\boldsymbol{R}_k$.

The Batch Least Squares Algorithm and its Measurement Model

With the mothership state estimated, the daughtership states can now be estimated using the BLS algorithm to process pixel coordinate and range measurements from the mothership to each daughtership. The BLS algorithm estimates initial states by accumulating measurements taken across N_{BLS} epochs, including and after the initial state, to increase the observability of the initial state by utilizing backpropagation of the current measurement Jacobians to the initial epoch as to map those measurements to the initial state.

As was done with the EKF, a measurement matrix $\boldsymbol{H}_k$ needs to be computed. Much of the work done for the EKF can be reutilized for finding the measurement matrix for the BLS pixel coordinates. The RIC frame used as the camera frame must be adjusted to now handle the mothership-to-daughtership pointing direction and to account for multiple epochs needing to be processed for a single state update.

$$\overrightarrow{\boldsymbol{R}}_{\text{RIC,BLS}} = \frac{\boldsymbol{r}_\alpha - \boldsymbol{r}_{m,\alpha}}{\|\boldsymbol{r}_\alpha - \boldsymbol{r}_{m,\alpha}\|} \tag{26}$$

$$\overrightarrow{\boldsymbol{C}}_{\text{RIC,BLS}} = \overrightarrow{\boldsymbol{R}}_{\text{RIC,BLS}} \times \left(\frac{\boldsymbol{v}_\alpha - \boldsymbol{v}_{m,\alpha}}{\|\boldsymbol{v}_\alpha - \boldsymbol{v}_{m,\alpha}\|}\right) \tag{27}$$

$$\overrightarrow{\boldsymbol{I}}_{\text{RIC,BLS}} = \overrightarrow{\boldsymbol{C}}_{\text{RIC,BLS}} \times \overrightarrow{\boldsymbol{R}}_{\text{RIC,BLS}} \tag{28}$$

where $\alpha = \lfloor N_{\text{BLS}}/2 \rfloor$ represents the epoch at which the daughtership being viewed is at the center of the image and subscript m indicates the state of the mothership specifically while no subscript m indicates the daughtership state. The pixel coordinates of the daughtership can be computed for the BLS using the exact same method outlined in Equations 18-25, but using the RIC frame rotation for the BLS outlined in Equations 26-28 instead. The additional Gaussian zero-mean white noise term also instead has covariance $\boldsymbol{R}_{\text{pix,BLS}}$.

For full observability of the daughtership states in the BLS, a range measurement $\rho_{m,i}$ from the mothership to each daughtership at epoch i is also necessary. This can be computed simply with a vector subtraction.

$$\rho_{m,i} = \|\boldsymbol{r}_i - \boldsymbol{r}_{m,i}\| + \omega_\rho = h_\rho(\boldsymbol{s}, t_i) \tag{29}$$

As was done in the EKF, the additional Gaussian zero-mean white noise term ω_ρ is only added when simulating the observations and is not added when computing the measurements from estimated states. It has covariance $\boldsymbol{R}_{\rho,\mathrm{BLS}}$.

Within a single BLS window of observation epochs, there can be anywhere from 0 to N_{BLS} number of ΔVs performed that must each be estimated along with the state to account for the thruster uncertainties. Because the number of ΔVs changes each BLS window, the shape of $\boldsymbol{H}_k$ is a function of the number of ΔVs and it must be accounted for that a ΔV at epoch i does not affect any of the states before that epoch. Specifically, $\boldsymbol{H}_k \in \mathbb{R}^{3\times(7+3n)}$, where n is the number of ΔVs in the current BLS observation window.

To construct $\boldsymbol{H}_k$, $\tilde{\boldsymbol{H}}_k$ is first constructed, which maps the pixel coordinates and range measurement to the current state. Then, that $\tilde{\boldsymbol{H}}_k$ matrix is backpropagated to the initial epoch with a vector of state transition matrices (STMs).

$$\tilde{\boldsymbol{H}}_k = \begin{bmatrix} \frac{\partial h_{\mathrm{pix}}}{\partial s} \\ \frac{\partial h_\rho}{\partial s} \end{bmatrix} \in \mathbb{R}^{3\times 7} \tag{30}$$

$$\boldsymbol{H}_k = \tilde{\boldsymbol{H}}_k[\Phi(t_i, t_0), \quad \Psi(t_i, t_{j-n}), \dots \quad \Psi(t_i, t_j)] \in \mathbb{R}^{3\times(7+3n)} \tag{31}$$

where $\Phi(t_i, t_0)$ is the STM from the inital epoch of the sim t_0 to the current epoch of the sim t_i. For every ΔV, there is an additional $\Psi \in \mathbb{R}^{3\times 7}$ STM. The value of Ψ is computed below.

$$\begin{aligned} \text{If} \quad t_i < t_j: \quad & \Psi(t_i, t_j) = \mathbf{0}_{7\times 3} \\ \text{Else}: \quad & \Psi(t_i, t_j) = \Phi(t_i, t_j) \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{I}_{3\times 3} \\ \mathbf{0}_{1\times 3} \end{bmatrix} \end{aligned} \tag{32}$$

Note that the STM $\Phi(t_i, t_j)$ in the $\Psi(t_i, t_j)$ equation is a separate STM from $\Phi(t_i, t_0)$ that is seen in Equation 31 as it begins initialized at epoch j as $\boldsymbol{I}_{7\times 7}$, but it is propagated with the same dynamics model as $\Phi(t_i, t_0)$. This ensures that any applied ΔV only affects the states that come after it.

All of these steps are summarized in the full BLS algorithm presented in Algorithm 1.

As seen in Algorithm 1, $\boldsymbol{s^*}_0$ is the initial nominal trajectory state for the daughtership that is being estimated for in the BLS, $\bar{P}_0$ is the *a priori* daughtership state covariance, $\boldsymbol{\Delta\bar{s}}_0$ is the *a priori* daughtership state shift, $\boldsymbol{Y}_i = h(\boldsymbol{s}_i(t_i), t_i)$ are the observations evaluated at the true state with noise $\omega = [\omega_{\mathrm{pix}}, \omega_\rho]^T \sim N(0, R_{\mathrm{BLS}})$, with $R_{\mathrm{BLS}} = \mathrm{diag}(\sigma^2_{\mathrm{pix}}, \sigma^2_{\mathrm{pix}}, \sigma^2_\rho)$. $F(\boldsymbol{s}^*, t)$ is the same dynamics Jacobian from the EKF, but evaluated on the nominal trajectory. As such, the residuals $\boldsymbol{y}_i = [\Delta p, \Delta l, \Delta\rho]^T$ are constructed to match the order of the covariance and observation matrix structure.

The Model Predictive Control Algorithm and Cost Function

With all the states of the spacecraft estimated through the EKF and BLS algorithms, a cost function for use within a model predictive control algorithm can be formulated to determine the near-optimal $\boldsymbol{\Delta V}$ each daughtership should administer to maintain the antipodal angle and orbit geometry admist gravitational and thruster uncertainties. In order to maintain long-term orbit stability for the two daughterships, the cost function must incorporate several orbit geometry terms in addition to $\boldsymbol{\Delta V}$ and antipodal angle penalties. For the

Algorithm 1 Batch Least Squares

Initialize: Set $i = 1$. Set $t_{i-1} = t_0, \boldsymbol{s}^*(t_{i-1}) = \boldsymbol{s}^*{}_0, \Phi(t_{i-1}, t_0) = \Phi(t_0, t_0) = I, \Lambda = \bar{P_0}^{-1}, N = \bar{P_0}^{-1} \boldsymbol{\Delta} \bar{\boldsymbol{s}}_0$

for $i = 1 : N_{\text{BLS}}$ **do**

  Read in observation: $t_i, \boldsymbol{Y}_i, R_i$

  Integrate ref traj and STM from t_{i-1} to t_i:

    $\boldsymbol{s_i^*} = f(\boldsymbol{s_{i-1}^*}(t), t)$ with ICs $\boldsymbol{s}^*(t_{i-1})$

    $\boldsymbol{F} = \frac{\partial f(\boldsymbol{s},t)}{\partial \boldsymbol{s}}|_{\boldsymbol{s}^*}$, i.e. evaluated on the nominal trajectory

    $\dot{\Phi}(t, t_0) = \boldsymbol{F}\Phi(t, t_0)$ with ICs $\Phi(t_{i-1}, t_0)$

  Accumulate current observation:

    $\tilde{H}_i = \frac{\partial h(\boldsymbol{s},t_i)}{\partial \boldsymbol{s}}|_{\boldsymbol{s}^*}$

    $\boldsymbol{y}_i = \boldsymbol{Y}_i - h(\boldsymbol{s}^*{}_i(t), t_i)$

    $H_i = \tilde{H}_i[\Phi(t_i, t_0), \quad \Psi(t_i, t_{j-n}), ... \quad \Psi(t_i, t_j)]$

    $\Lambda = \Lambda + H_i^T R_i^{-1} H_i$

    $N = N + H_i^T R_i^{-1} \boldsymbol{y}_i$

  if $t_i < t_{final}$ **then**

    $i = i + 1$. Thus,

    $t_{i-1} \leftarrow t_i$

    $\boldsymbol{s}^*(t_{i-1}) \leftarrow \boldsymbol{s}^*(t_i)$

    $\Phi(t_{i-1}, t_0) \leftarrow \Phi(t_i, t_0)$

    Continue

  else if $t_i \geq t_{final}$ **then**

    Solve normal equations: $\boldsymbol{\Delta \hat{s}_0} = \Lambda^{-1} N$

    if $||\Delta \boldsymbol{\hat{s}_0}|| \geq \epsilon$ **then**

      $\boldsymbol{s}^*{}_0 \leftarrow \boldsymbol{s}^*{}_0 + \boldsymbol{\Delta \hat{s}_0}$

      $\boldsymbol{\Delta} \bar{\boldsymbol{s}}_0 \leftarrow \boldsymbol{\Delta} \bar{\boldsymbol{s}}_0 - \boldsymbol{\Delta \hat{s}_0}$

      Continue

    else if $||\Delta \boldsymbol{\hat{s}_0}|| < \epsilon$ **then**

      Break

    end if

  end if

end for

orbit geometry, antipodal angle, and $\boldsymbol{\Delta V}$ terms to be equally considered, they must additionally be normalized to be around the same order of magnitude, ~ 0 to 1. The full scalar cost function J_{MPC} that is being minimized is outlined below in Equation 33 followed by a detailed breakdown of each term.

$$
\begin{aligned}
J_{\text{MPC}} = \frac{1}{\mathcal{H}} \sum_{j=1}^{\mathcal{H}} \Bigg[& \frac{d_{\min,1,j}}{\beta_{\Delta t,\text{MPC}} l_{\max,1}} + \frac{d_{\min,2,j}}{\beta_{\Delta t,\text{MPC}} l_{\max,2}} + \frac{\alpha_{\text{antipodal}}}{\sigma_{\text{antipodal}}} + \left(\frac{a_{1,j} - a_{2,j}}{0.1 r_0}\right)^2 + 10 e_{1,j}^2 + ... \\
& ... + 10 e_{2,j}^2 + \frac{1}{2\sigma_{\text{inc}}}(i_{1,j} + i_{2,j}) + \left(\frac{\Delta V_{1,j}}{\Delta V_{\min}}\right)^2 \left(\left(\frac{\Delta V_{1,j}}{\Delta V_{\min}} - 1\right)^2 + 1\right) + ... \\
& ... + \left(\frac{\Delta V_{2,j}}{\Delta V_{\min}}\right)^2 \left(\left(\frac{\Delta V_{2,j}}{\Delta V_{\min}} - 1\right)^2 + 1\right) \Bigg]
\end{aligned} \tag{33}
$$

Where $\mathcal{H}$ is the number of steps in the MPC's prediction horizon. The MPC cost is accrued by computing the individual costs at epoch j in the prediction horizon and then propagating forward to step $j+1$ using the current epoch's $\boldsymbol{\Delta V_j}$ and a similar dynamics model to Equations 1-7 that is modified for gravitational uncertainties. This simple modification will be outlined following the breakdown of the cost components. Within the MPC's prediction horizon, the amount of time propagated between epochs is scaled by a factor $\beta_{\Delta t,\text{MPC}}$. If $\beta_{\Delta t,\text{MPC}} = 8$, between epochs j and $j+1$, $8\Delta t$ seconds will be propagated using the dynamics. This is to allow the MPC to see further ahead in time without having to increase the computational expense by adding more decision variables. Also note that, while the MPC optimizes the 3-D $\boldsymbol{\Delta V_j}$ for each daughtership at every epoch j, only the first epoch $\boldsymbol{\Delta V_j}$'s are used in propagation outside the MPC.

The first two of the nine cost function terms are denoted the "position costs". Daughtership 1's minimum distance from their nearest reference segment at epoch j in the prediction horizon is denoted $d_{\min,1,j}$. By checking distance to the nearest reference segment, this enables the MPC to be time-of-flight agnostic, a methodology explored in previously completed work for a simulated single-spacecraft asteroid landing on 6489 Golevka.[16] This term is normalized by $l_{\max,1}$, which represents the maximum segment length in the daughtership's reference trajectory. The second term performs the same computation for daughtership 2.

The third term in the cost function, denoted the "antipodal cost" is the penalty for deviating from the perfect 180° antipodal angle. The angle the daughterships are apart from the perfect 180° antipodal angle is $\alpha_{\text{antipodal}}$. Per the DROID mission proposal,[5] the maximum allowed $\alpha_{\text{antipodal}}$ is 15°, so we normalize with $3\sigma_{\text{antipodal}} = 15° \rightarrow \sigma_{\text{antipodal}} = 5°$.

The fourth term, denoted the "semimajor axis cost" penalizes the difference in the semimajor axes between each daughtership at epoch j in the prediction horizon. This difference is normalized by dividing by 10% of the initial circular orbit radius. The term is squared to ensure it is always positive.

Terms five and six, denoted the "eccentricity costs", penalize deviation from perfect circular orbits for each daughtership. Circular orbits are used over other orbit regimes for tractability. Frozen orbits could be considered here, but would require modification of the eccentricity cost to not penalize their non-zero eccentricities.

Term seven is the "inclination cost", which penalizes the average inclination of the daughterships. This term is normalized with $\sigma_{\text{inc}} = 10°$. Inclinations are computed with respect to the local terminator plane defined by the inertial Sun to Apophis unit vector direction.

Terms eight and nine are the "ΔV costs". The ΔV costs are quartic in nature to ensure that there is zero cost when the $\|\boldsymbol{\Delta V}\| = 0$, a cost of 1 when $\|\boldsymbol{\Delta V}\| = \Delta V_{\min}$, and a rapidly growing cost above $\Delta V_{\min}$. Because the MPC must handle thruster uncertainties, it is not desired to burn below $\Delta V_{\min}$ or a large portion of the true administered ΔV will be erroneous. However, if it is more optimal for the daughtership not to burn, it should accrue no penalty for doing so. The magnitude of $\Delta V_{\min}$ will be broken down in the subsequent paragraph.

As mentioned previously, the full algorithmic loop containing the EKF, BLS, and MPC must be able to handle gravitational and thruster uncertainty. The gravitational uncertainties, the aformentioned dynamics model modification for use inside the MPC prediction horizon, is done by replacing $\boldsymbol{a}(\boldsymbol{s})_{\text{poly}}$ with simple two-body gravitational acceleration $\boldsymbol{a}(\boldsymbol{s}) = \mu\boldsymbol{r}/r^3$, eliminating the higher-order gravity terms from the dynamics model. All other terms in the dynamics are left untouched. Thruster uncertainty is implemented outside the MPC. Once the MPC outputs its near-optimal ΔV for each daughtership, the ΔV values are clipped. Any $\|\boldsymbol{\Delta V}\| < \Delta V_{\text{min}}$ is zero-ed out and not performed. Then, if any ΔV remains, it is given to the Gates model for ΔV uncertainty.[23] The Gates model works by computing fixed and proportional magnitude and pointing errors in ΔV to add to the commanded ΔV vector. The Gates model is formulated as follows.[24]

$$\boldsymbol{P}_{\text{Gates}} = \begin{bmatrix} \sigma^2_{1,\text{Gates}} + \gamma^2\sigma^2_{2,\text{Gates}} & 0 & 0 \\ 0 & \sigma^2_{3,\text{Gates}} + \gamma^2\sigma^2_{4,\text{Gates}} & 0 \\ 0 & 0 & \sigma^2_{3,\text{Gates}} + \gamma^2\sigma^2_{4,\text{Gates}} \end{bmatrix} \tag{34}$$

The additional erroneous ΔV applied when propagating the true daughtership states is computed by sampling a normal distribution with zero-mean and covariance $\boldsymbol{P}_{\text{Gates}}$. $\sigma_{1,\text{Gates}}$ is the fixed magnitude error, $\sigma_{2,\text{Gates}}$ is the proportional magnitude error, $\sigma_{3,\text{Gates}}$ is the fixed pointing error, and $\sigma_{4,\text{Gates}}$ is the proportional pointing error with $\gamma = \|\boldsymbol{\Delta V}_{\text{MPC}}\|$. Note that the resulting erroneous $\boldsymbol{\Delta V}_{\text{Gates}}$ vector is added to the commanded $\boldsymbol{\Delta V}_{\text{MPC}}$ vector rotated into a frame where all commanded ΔV is in the x-direction. Following the vector addition in this ΔV frame, the actual velocity is rotated back into the inertial frame. As such, the actual ΔV vector is $\boldsymbol{\Delta V}_{\text{actual}} = \boldsymbol{\Delta V}_{\text{MPC}} + \boldsymbol{\Delta V}_{\text{Gates}}$. It is important to note that the erroneous ΔV is only used when propagating the true state; the estimated state uses the commanded ΔV straight out of the MPC. The erroneous component of the ΔV is then estimated in the next BLS. The full algorithm with the EKF, BLS, MPC, and post-processing is outlined below in Algorithm 2. Note that "s/c" denotes spacecraft and "d-ship" denotes daughtership in the algorithm pseudocode.

Algorithm 2 MPC Algorithm

$t \Leftarrow 0$
OBTAIN Apophis, EMB, Jupiter, Sun ephemeris data
GENERATE ref. trajs for all three s/c
while $t < t_{\text{duration}}$ **do**
    SET $\boldsymbol{\Delta V}_{\text{guess}} \Leftarrow \boldsymbol{0}_{1\times(2*3*\mathcal{H})}$
    COMPUTE $\boldsymbol{\Delta V}_{\text{MPC}}$ to minimize COST(J_{MPC}) with two-body gravity dynamics model
    if Daughterhsip k, first epoch of $\boldsymbol{\Delta V}_{\text{MPC},k} < \Delta V_{\text{min}}$ **then**
        $\boldsymbol{\Delta V}_{\text{MPC},k} \leftarrow \boldsymbol{0}$
    end if
    COMPUTE erroneous $\boldsymbol{\Delta V}_{\text{actual}}$ with Gates model
    PROPAGATE true states with first epoch of $\boldsymbol{\Delta V}_{\text{actual}}$ and full dynamics model
    PROPAGATE estimated states with first epoch of $\boldsymbol{\Delta V}_{\text{MPC}}$ and full dynamics model
    COMPUTE new observations: Apophis surface image, d-ship images, d-ship ranges
    FILTER mothership estimated state with the EKF and Apophis surface image
    if number of new d-ship observations since last BLS = N_{BLS} **then**
        FILTER d-ship estimated states, $\boldsymbol{\Delta V}_{\text{MPC}}$ with BLS, d-ship images, d-ship ranges
    end if
    t += 1
end while

RESULTS

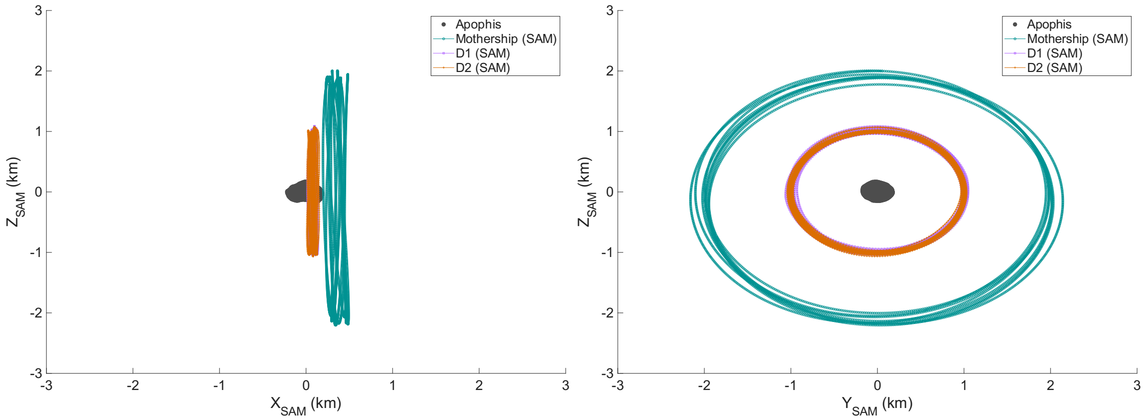
To simulate the long-term characterization of Apophis with the MPC for remote guidance, several parameters and initial conditions must be defined. Firstly, all three spacecraft are put into locally-stable offset terminator orbits around Apophis. Because Apophis is not in a circular heliocentric orbit, the offset from the

Table 1. Simulation Parameters and Initial Conditions

$\boldsymbol{s}_{0,\mathrm{M}}$ (m, mm/s)	$[-142.0, 470.6, -1962.1, 33.65, -16.68, -6.435]$
$\boldsymbol{s}_{0,\mathrm{D}_1}$ (m, mm/s)	$[853.1, -490.4, -189.2, 0.0, -19.82, 51.37]$
$\boldsymbol{s}_{0,\mathrm{D}_2}$ (m, mm/s)	$[-917.5, 377.3, 145.6, 0.0, 19.63, -50.88]$
$\boldsymbol{\Delta}\hat{\boldsymbol{s}}_{0,\mathrm{M}}$ (m, mm/s)	$[0.0, 0.0, 0.0, 0.0, 0.0, 0.0]$
$\boldsymbol{\Delta}\hat{\boldsymbol{s}}_{0,\mathrm{D}_1}$ (m, mm/s)	$[2.0, 2.0, 2.0, 1.0, 1.0, 1.0]$
$\boldsymbol{\Delta}\hat{\boldsymbol{s}}_{0,\mathrm{D}_2}$ (m, mm/s)	$[-2.0, -2.0, -2.0, -1.0, -1.0, -1.0]$
$\mathcal{H}$	10
$\beta_{\Delta t,\mathrm{MPC}}$	8
N_{BLS}	30
N_{Features}	600
$C_{\mathrm{R,M}}$, $C_{\mathrm{R,D}_1}$, $C_{\mathrm{R,D}_2}$	$1.2, 1.2, 1.2$
$m_{\mathrm{wet,M}}$, $m_{\mathrm{wet,D}_1}$, $m_{\mathrm{wet,D}_2}$ (kg)	$200, 12, 12$
$m_{\mathrm{dry,M}}$, $m_{\mathrm{dry,D}_1}$, $m_{\mathrm{dry,D}_2}$ (kg)	$180, 11, 11$
Surface area: S_{M}, S_{D_1}, S_{D_2} (m^2)	$3.14, 0.36, 0.36$
Gates: σ_1, σ_2, σ_3, σ_4 (mm/s, %, mm/s, mrad)[23]	$0.85, 0.0, 0.0, 4.3$
$\boldsymbol{Q}$ (mm^2, mm^2/s^2, kg^2)	diag(100, 100, 100, 0.001, 0.001, 0.001, 1e-20)
$\boldsymbol{P}_0$ (m^2, mm^2/s^2, kg^2)	diag(4, 4, 4, 1, 1, 1, 1e-16)
Image width, height (pix, pix)	1024, 1024
Image focal length, f (mm)	12
Image pixel pitch, μ_{pix} (mm/pix)	0.00325
σ^2_{pix}, σ^2_{ρ} (pix^2, mm^2)	0.01, 200
ΔV_{min} (mm/s)	$3\sigma_{1,\mathrm{Gates}} = 2.55$
Δt (s)	600
Simulation duration, t_{duration} (months, #Δt)	1, 4320
Simulation starting epoch (ET)	01-JUL-2028 00:00:00.0000

terminator plane is constantly shifting, hence locally-stable. The initial conditions of these orbits and other parameters like N_{BLS} and $\mathcal{H}$ can be found below in Table 1.

Several figures are generated to demonstrate the effectiveness of the EKF, BLS, and MPC algorithms for the remote guidance asteroid exploration application. The 3-D trajectories of the mothership and daughterships in the Sun Anti-Momentum (SAM) frame[25] can be found in Figure 3 and the per-axis breakdown in Figure 4. The SAM frame defines the +X-axis direction as the direction from the Sun to Apophis, the +Z-axis direction as Apophis' negative heliocentric angular momentum direction, and the +Y-axis direction defined to complete the right-handed frame. The direct MPC-commanded ΔV and the actual ΔV experienced by the spacecraft due to additional thruster error can be found in Figure 5. The EKF and BLS performance can be quickly assessed by plotting the state and ΔV residuals for the mothership and daughterships. The state



(a) SAM XZ-Plane (b) SAM YZ-Plane

Figure 3. 1-month Simulation 3D Trajectories in the SAM Frame

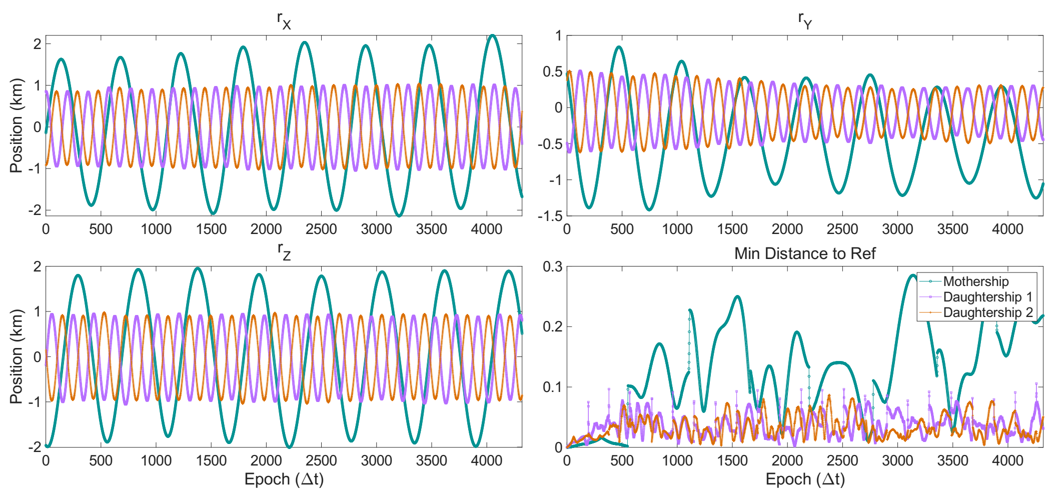


Figure 4. 1-Month Position State Results

residuals can be found in Figure 6 and the ΔV residuals can be found in Figure 7. With the primary mission goal being maintaining the daughterships on antipodal orbits, plotting the true and estimated antipodal angle between the daughterships over time will allow analysis of the MPC's ability to accomplish this goal. These results are found in Figure 8. The cost has many other terms outside the antipodal angle cost. The history of all the individual terms for the entire mission is plotted in Figure 9. Similarly, the orbital elements of each daugthership's orbit with respect to its local terminator plane are plotted in Figure 10.

DISCUSSION

Overall, this application of MPC demonstrated viability to the remote guidance asteroid exploration problem by administering ΔV that maintains the antipodal angle within the $3\sigma_{\text{antipodal}}$ bound of $15°$ maximum deviation from the perfect $180°$ antipodal angle, as seen in Figure 8. Daughterships 1 and 2 burned 123.55 and 80.92 mm/s ΔV over the course of the 1-month simulation, including erroneous ΔV contributions. Excluding erroneous ΔV, daughterships 1 and 2 commanded 118.87 and 82.47 mm/s, meaning erroneous ΔV added ΔV magnitude to daughership 1 and removed ΔV magnitude from daughtership 2.

When analyzing the position histories found in Figure 4, the peaks between daughtership 1 and daughter-

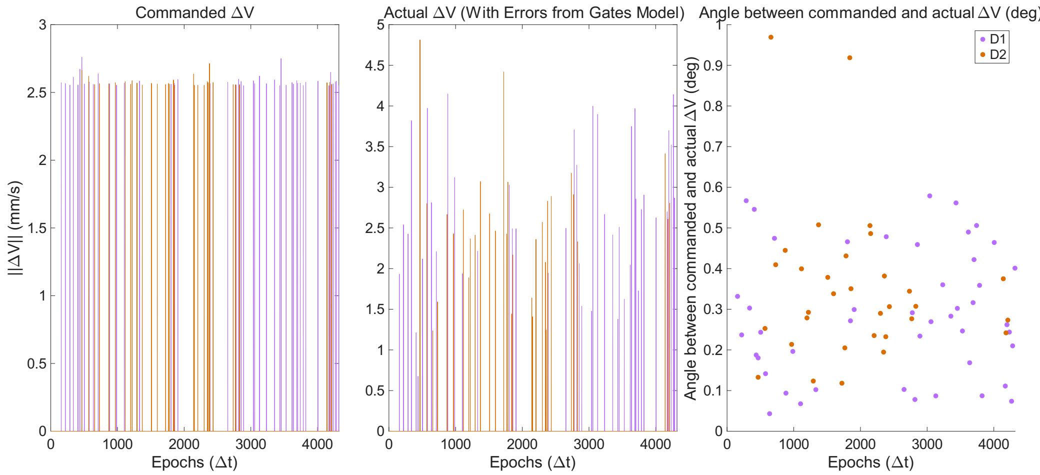


Figure 5. ΔV Commanded vs Actual History

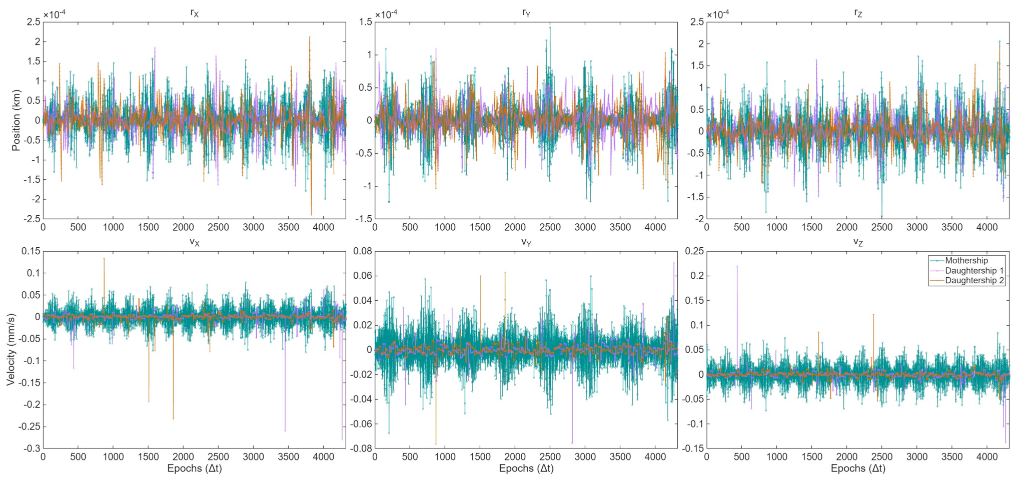


Figure 6. True - Estimated State History

ship 2 (henceforth denoted "D1" and "D2") are similar in magnitude and opposite in phase. The spikes seen in the minimum distance to the reference are a consequence of the reference trajectory being generated with respect to the initial terminator plane offset. The reference trajectories are only stored for a single orbit period and the spikes occur when the spacecraft begins a new orbit period. Because the offset is different than it was at the beginning of the simulation, there is a large jump in minimum distance as the spacecraft is still operating in the locally stable offset plane while the reference is at a different offset.

The commanded ΔV, as seen in Figure 5, all correctly is above the $3\sigma_{1,\text{Gates}}$ threshold. The actual ΔV magnitude varies significantly from the commanded ΔV magnitude. The direction between commanded and actual ΔV is similar, however, implying that most of the ΔV error is simply in the magnitude. The threshold, being only $3\sigma_{1,\text{Gates}}$, means numerous epochs are over $\pm 1\sigma_{1,\text{Gates}}$ away from the commanded ΔV, the D2 ΔV around epoch 450 being almost twice as much as it was commanded to be. The BLS must be able to properly estimate these significant ΔV errors.

The performance of the EKF and BLS can be assessed in Figures 6-7. Overall, the EKF and BLS reconstructed the actual spacecraft states with high accuracy, the position states being roughly centimeter-level accuracy and the velocity states being roughly micrometer-per-second-level accuracy, notably with the moth-

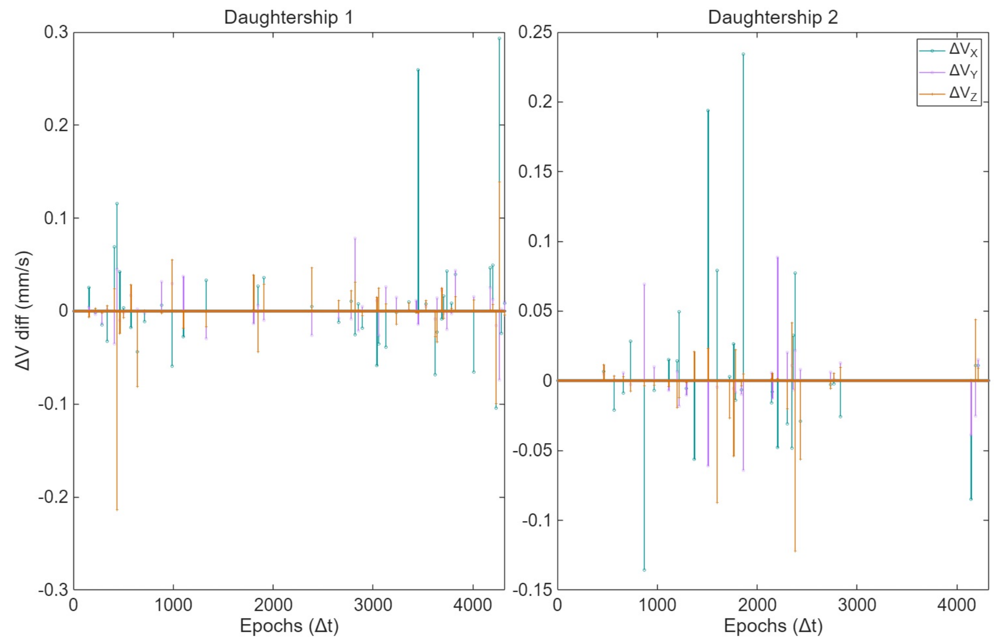


Figure 7. True - Estimated ΔV History

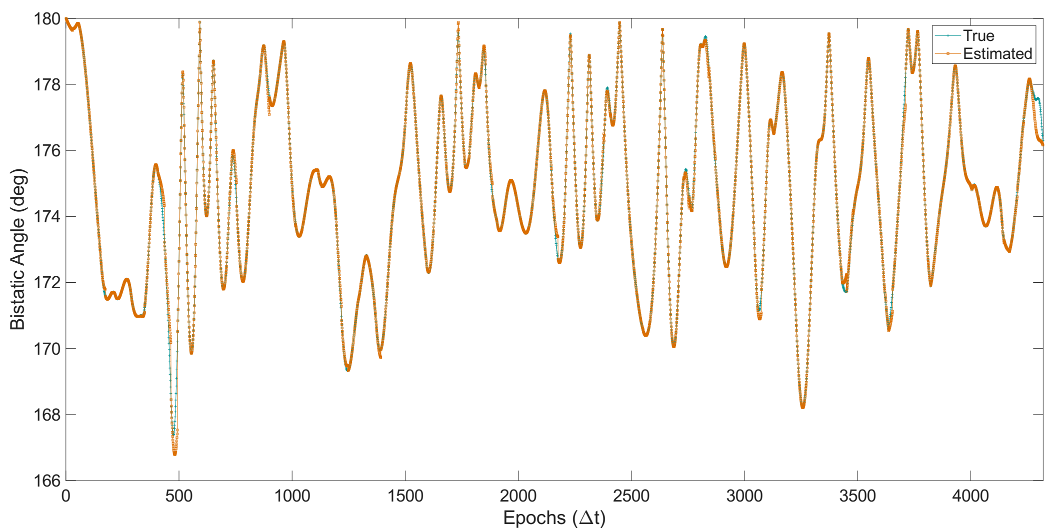


Figure 8. Antipodal Angle History

ership velocity estimated being an order of magnitude worse. This is due to the mothership estimating her full state from a single image of surface features. Much of the velocity information fed into the EKF is from the *a priori* covariance. The daughtership measurements, being both an image and a range for each epoch in the BLS, enables full state reconstruction while also providing significantly stronger velocity information as the pixel coordinates change in the image plane over time, but still subject to the estimation error in the mothership state since both the image and range measurements are relative to the mothership state. The ΔV estimation in the BLS significantly improved residuals between the commanded and actual ΔV vectors, the worst residual components being around $0.35\sigma_{1,\text{Gates}}$.

When analyzing the individual cost terms in Figure 9, the antipodal angle cost dominates, with the position

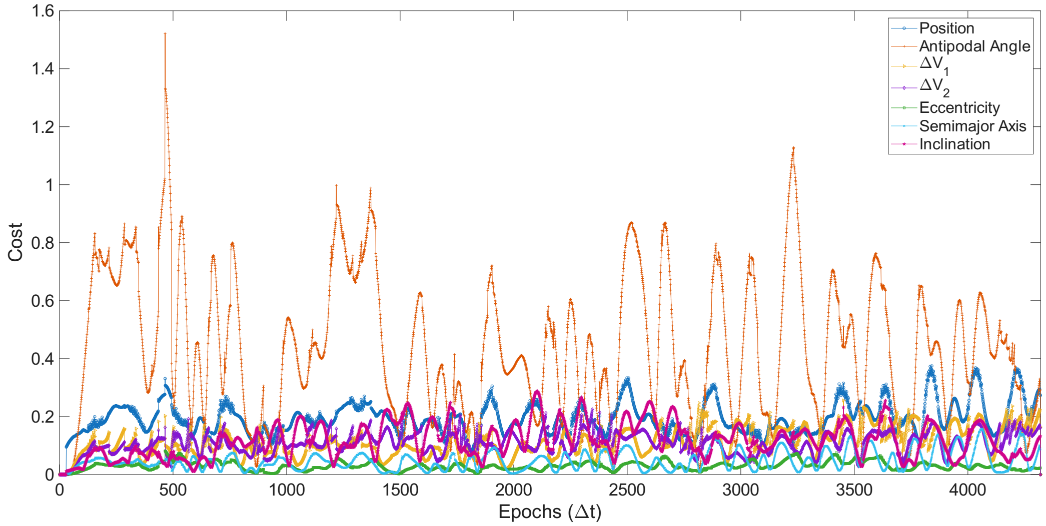


Figure 9. All Cost Term Histories

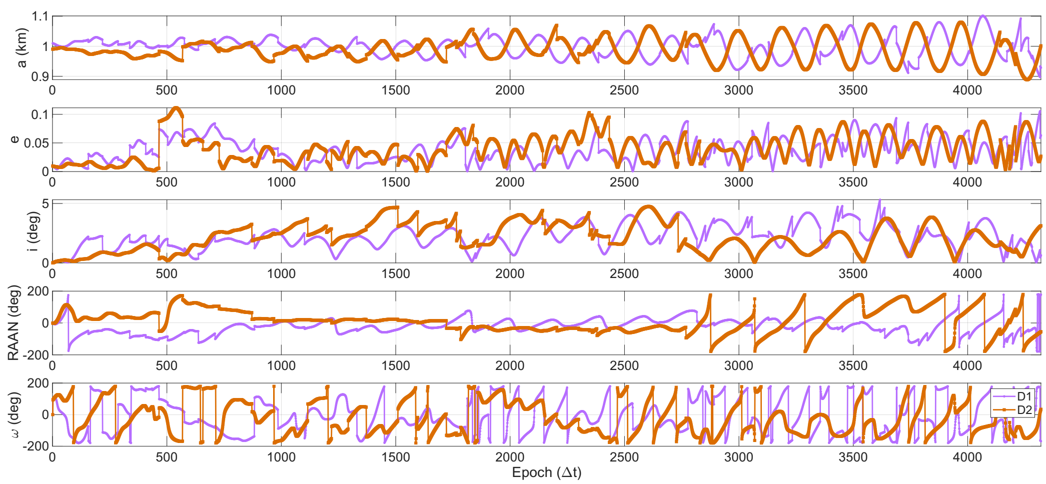


Figure 10. Orbital Element Histories in Terminator Plane

cost being the next highest average contribution. The antipodal cost grows the fastest as it is the least stable of the various geometric quantities assessed in this work. Recall that, by definition of the antipodal angle cost, the magnitude of that cost at the current epoch is the average antipodal cost $\beta_{\Delta t,\mathrm{MPC}}\mathcal{H} = 80$ time steps Δt in the future. This is why, at around epochs 480-500, the antipodal cost is extremely steep. The local antipodal minima at around epoch 350 predicts the antipodal angle will rise in the near future, hence low antipodal cost around this epoch. However, after rising to the local maxima around epoch 400, the antipodal angle is predicted to drop fast, so a spike in the antipodal cost is seen. ΔV action is taken to course-correct and the antipodal cost spikes back downwards. A similar analysis can be done on any of the other peaks and valleys of the antipodal angle history and the antipodal angle cost.

The ΔV profile seen in Figure 5 does not align with the ΔV cost magnitudes, however. While there are several epochs without burns, there are no epochs with zero ΔV cost after the first BLS window where no burns were allowed to be administered. This is a consequence of the ΔV thresholding. The choice to apply the $3\sigma_{1,\mathrm{Gates}}$ threshold on ΔV outside the optimizer was done as to not create a discontinuous feasible solution space where the ΔV can either be zero or above $3\sigma_{1,\mathrm{Gates}}$ magnitude. The optimizer has no

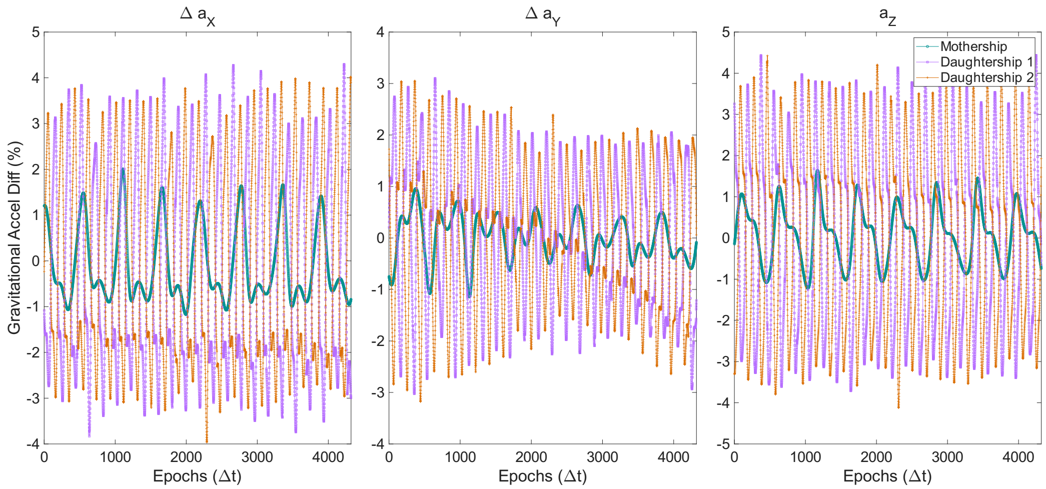


Figure 11. Polyhedral - Point Mass Gravity History

knowledge that ΔVs below $3\sigma_{1,\mathrm{Gates}}$ will not be performed, so it can command burns below the threshold. These ΔVs impact every other term in the cost function as the prediction horizon is propagated expecting the commanded ΔVs to take place. As such, when the burn the MPC commands does not occur, the costs are in a worse state than the previous run-through, so it finds a new set of ΔV commands. Eventually, the need to burn will be high enough that the commanded ΔV magnitude will be above threshold and the burn will actually occur. This is why the ΔV cost profiles are frequently sawtooth-shaped; the spike down in ΔV cost is once a commanded burn finally exceeds the threshold and is performed. However, this is a glaring issue in the formulation; the cost terms are likely not representative of any future trajectory that would be flown since any number of ΔVs in the prediction horizon can be below the ΔV threshold.

When analyzing the orbital element histories throughout the mission duration from Figure 10, there is a clear growing instability, particularly in semimajor axis and eccentricity. Inclination also grows over time, but the daughterships do seem to match one another better near the end of the simulation. Spikes seen in all orbital elements are caused by impulsive ΔVs. Despite the semimajor axis cost being a function of D1 and D2's semimajor axis difference squared, the two semimajor axes seem to instead prefer to be equal and opposite in their peaks. While eccentricity between the daughterships is not coupled, there is still similar oscillatory growth. There are multiple factors to account for this steady growth in orbital elements. One factor is the fact that the ΔVs have erroneous components to them that will almost certainly worsen orbital elements. Further, there isn't a guarantee that any error in a D1 ΔV will be equal and opposite a D2 ΔV error, so errors are not effectively "canceling out". Another significant factor is that the reference trajectories are only locally stable, as mentioned previously. While the rotation from inertial frame to terminator frame (Sun direction points to +Z) is local, the offset from the terminator X-Y plane is assumed constant for the reference trajectory. The orbital elements within the MPC cost in Figure 9 and as they are plotted in Figure 10 also assume this constant offset. For a single daughtership orbit period of about 32 hours, Apophis' distance from the Sun lowers by about 0.5% from its distance from the Sun at the start of the simulation, a non-negligible change in SRP magnitude and terminator orbit offset over the course of a 1-month simulation.

The use of the point-mass gravity model within the prediction horizon, when analyzing Figure 11, is unsurprisingly a fairly significant contribution of additional error. The acceleration difference is zero-mean, but the difference between the two gravity models can get as high as 4%. However, while this error could easily propagate over time, the cyclic nature of the gravity model differences likely end up negating much of the long-term effects, making it less of a contribution to instability than erroneous ΔV and the lack of local offset computation.

CONCLUSION

This work proposes an application of Model Predictive Control to address the remote guidance asteroid exploration problem, specifically attempting to maintain the antipodal angle between two daughterships equipped with a bistatic radar admist thrust and gravitational uncertainties. A 1-month simulation around Apophis demonstrated viability of the approach with the MPC's commanded ΔV profile maintaining the antipodal angle within the set $3\sigma_{\text{antipodal}}$ bound. However, in the current formulation, the ΔV threshold is applied only outside the MPC, so the cost function still penalizes sub-threshold burns that will ultimately be discarded and computes other costs assuming those same burns occurred, altering the converged near-optimal ΔV commands. This causes the MPC to optimize against a cost profile that does not correspond to the ΔV commands that are actually executed. Additionally, the assumption of a constant local terminator plane offset throughout the 1-month simulation results in unstable orbital elements between the daughterships, which artifically drives up MPC cost components, causing further instability.

FUTURE WORK

Significant progress has been made on implementing both the discontinuous, thresholded ΔV solution space with the cost terms being computed according to the actual commanded ΔV and incorporating local terminator plane offset estimation into the time of flight free MPC. It is expected to provide better orbital element stability while maintaining a better antipodal angle history and at a reduced computational expense.

ACKNOWLEDGMENTS

This material is based on work supported by the NASA Space Technology Graduate Research Opportunities Award No. 80NSSC23K1199. The authors gratefully acknowledge Malav Patel for his assistance in reviewing the code repository.

REFERENCES

[1] A. Barnett, "Apophis Facts," `https://science.nasa.gov/solar-system/asteroids/apophis-facts/`, 2025. Online; accessed 16-March-2026.

[2] D. N. DellaGiustina, M. C. Nolan, A. T. Polit, M. C. Moreau, D. R. Golish, A. A. Simon, C. D. Adam, P. G. Antreasian, R.-L. Ballouz, O. S. Barnouin, K. J. Becker, C. A. Bennett, R. P. Binzel, B. J. Bos, R. Burns, N. Castro, S. R. Chesley, P. R. Christensen, M. K. Crombie, M. G. Daly, R. T. Daly, H. L. Enos, D. Farnocchia, S. Freund Kasper, R. Garcia, K. M. Getzandanner, S. D. Guzewich, C. W. Haberle, T. Haltigin, V. E. Hamilton, K. Harshman, N. Hatten, K. M. Hughes, E. R. Jawin, H. H. Kaplan, D. S. Lauretta, J. M. Leonard, A. H. Levine, A. J. Liounis, C. W. May, L. C. Mayorga, L. Nguyen, L. C. Quick, D. C. Reuter, E. Rivera-Valentín, B. Rizk, H. L. Roper, A. J. Ryan, B. Sutter, M. M. Westermann, D. R. Wibben, B. G. Williams, K. Williams, and C. W. V. Wolner, "OSIRIS-APEX: An OSIRIS-REx Extended Mission to Asteroid Apophis," *The Planetary Science Journal*, Vol. 4, oct 2023, p. 198, 10.3847/PSJ/acf75e.

[3] J. Männel, T. Herbst, H. Kayal, J. Klesse, and T. Neumann, "Updates on the Apophis Interceptor Mission," *EPSC-DPS Joint Meeting, Helsinki, Finland*, 2025, https://doi.org/10.5194/epsc-dps2025-1299.

[4] V. H. Megia, T.-M. Ho, J. T. Grundmann, and M. Noeker, "Modeling and control techniques for landing on (99942) Apophis: An analysis of a mission scenario," *Acta Astronautica*, Vol. 235, 2025, pp. 185–194, https://doi.org/10.1016/j.actaastro.2025.04.031.

[5] C. A. Raymond, R. Amini, S. Bandyopadhyay, J. Bellerose, S. Bhaskaran, P. Adell, P. Bousquet, M. Calaprice, B. Davidsson, C. Elachi, L. Fesq, M. Haynes, A. Herique, R. Karimi, J. T. Keane, and P. Michel, "The Droid Mission Concept to Characterize Apophis Through its 2029 Earth Closest Approach.," *AGU Fall Meeting Abstracts*, Vol. 2023 of *AGU Fall Meeting Abstracts*, Dec. 2023, pp. P34A–04.

[6] B. D. Tapley and C. Reigber, "The GRACE mission: status and future plans," *EGS General Assembly Conference Abstracts*, 2002, p. 4819.

[7] R. Roncoli and K. Fujii, "Mission design overview for the Gravity Recovery and Interior Laboratory (GRAIL) mission," *AIAA/AAS astrodynamics specialist conference*, 2010, p. 8383.

[8] D. J. Scheeres, "Proximity Operations About Apophis Through Its 2029 Earth Flyby," *The Journal of the Astronautical Sciences*, Vol. 69, 2022, pp. 1514–1536, 10.1007/s40295-022-00360-w.

[9] D. A. Lorenz, R. Olds, A. May, C. Mario, M. E. Perry, E. E. Palmer, and M. Daly, "Lessons learned from OSIRIS-REx autonomous navigation using natural feature tracking," *2017 IEEE Aerospace Conference*, IEEE, 2017, pp. 1–12.

[10] J. Sullivan and C. D'Souza, "Extended Kalman filter performance on the Artemis-1 mission," *45th Rocky Mountain AAS GN&C Conference*, No. AAS 23-053, 2023.

[11] K. Wang, J. Liu, H. Su, A. El-Mowafy, and X. Yang, "Real-time LEO satellite orbits based on batch least-squares orbit determination with short-term orbit prediction," *Remote Sensing*, Vol. 15, No. 1, 2022, p. 133.

[12] B. D. Tapley, B. E. Schutz, and G. H. Born, *Statistical Orbit Determination*. Elsevier Academic Press, 2004.

[13] D. Simon, *Nonlinear Kalman filtering*, ch. 13, pp. 393–431. Hoboken, NJ: John Wiley & Sons, Ltd, 2006, https://doi.org/10.1002/0470045345.ch13.

[14] R. Gaskell, O. Barnouin-Jha, D. J. Scheeres, A. Konopliv, T. Mukai, S. Abe, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, *et al.*, "Characterizing and navigating small bodies with imaging data," *Meteoritics & Planetary Science*, Vol. 43, No. 6, 2008, pp. 1049–1061.

[15] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: An engineering perspective," *The International Journal of Advanced Manufacturing Technology*, Vol. 117, No. 5, 2021, pp. 1327–1349.

[16] L. Feld, J. Lyzhoft, and K. Ho, "Time-of-Flight Free Model Predictive Control Application for Polyhedral Asteroid Landings," *47th Annual AAS Guidance, Navigation and Control (GN and C) Conference*, 2025.

[17] M. Brozović, L. A. Benner, J. G. McMichael, J. D. Giorgini, P. Pravec, P. Scheirich, C. Magri, M. W. Busch, J. S. Jao, C. G. Lee, L. G. Snedeker, M. A. Silva, M. A. Slade, B. Semenov, M. C. Nolan, P. A. Taylor, E. S. Howell, and K. J. Lawrence, "Goldstone and Arecibo radar observations of (99942) Apophis in 2012–2013," *Icarus*, Vol. 300, 2018, pp. 115–128, https://doi.org/10.1016/j.icarus.2017.08.032.

[18] R. A. Werner and D. J. Scheeres, "Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia," *Celestial Mechanics and Dynamical Astronomy*, Vol. 65, 1996, pp. 313–344.

[19] V. Reddy, M. S. Kelley, J. Dotson, D. Farnocchia, N. Erasmus, D. Polishook, J. Masiero, L. A. M. Benner, J. Bauer, M. R. Alarcon, D. Balam, D. Bamberger, D. Bell, F. Barnardi, T. H. Bressi, M. Brozovic, M. J. Brucker, L. Buzzi, J. Cano, D. Cantillo, R. Cennamo, S. Chastel, O. Chingis, Y.-J. Choi, E. Christensen, L. Denneau, M. Dróżdż, L. Elenin, O. Erece, L. Faggioli, C. Falco, D. Glamazda, F. Graziani, A. N. Heinze, M. J. Holman, A. Ivanov, C. Jacques, P. J. v. Rensburg, G. Kaiser, K. Kamiński, M. K. Kamińska, M. Kaplan, D.-H. Kim, M.-J. Kim, C. Kiss, T. Kokina, E. Kuznetsov, J. A. Larsen, H.-J. Lee, R. C. Lees, J. d. León, J. Licandro, A. Mainzer, A. Marciniak, M. Marsset, R. A. Mastaler, D. L. Mathias, R. S. McMillan, H. Medeiros, M. Micheli, A. Mokhnatkin, H.-K. Moon, D. Morate, S. P. Naidu, A. Nastasi, A. Novichonok, W. Ogłoza, A. Pál, F. Pérez-Toledo, A. Perminov, E. Petrescu, M. Popescu, M. T. Read, D. E. Reichart, I. Reva, D.-G. Roh, C. Rumpf, A. Satpathy, S. Schmalz, J. V. Scotti, A. Serebryanskiy, M. Serra-Ricart, E. Sonbas, R. Szakáts, P. A. Taylor, J. L. Tonry, A. F. Tubbiolo, P. Veres, R. Wainscoat, E. Warner, H. J. Weiland, G. Wells, R. Weryk, L. F. Wheeler, Y. Wiebe, H.-S. Yim, M. Żejmo, A. Zhornichenko, S. Zoła, and P. Michel, "Apophis Planetary Defense Campaign," *The Planetary Science Journal*, Vol. 3, may 2022, p. 123, 10.3847/PSJ/ac66eb.

[20] O. Montenbruck and E. Gill, *Satellite Orbits: Models, Methods, and Applications*. Springer, 2012.

[21] NASA JPL, "Solar System Dynamics Horizons Web Application," 2026. Accessed: January 5th, 2026.

[22] S. Bhaskaran, S. Nandi, S. Broschart, M. Wallace, L. A. Cangahuala, and C. Olson, "Small body landings using autonomous onboard optical navigation," *The Journal of the Astronautical Sciences*, Vol. 58, No. 3, 2011, pp. 409–427.

[23] C. R. Gates, "A simplified model of midcourse maneuver execution errors," tech. rep., 1963.

[24] S. V. Wagner, "Cassini Maneuver performance assessment and execution-error modeling through 2015," 2016.

[25] D. R. Wibben, K. E. Williams, J. V. McAdams, P. G. Antreasian, J. M. Leonard, and M. C. Moreau, "Maneuver strategy for OSIRIS-REx proximity operations," *GNC 2017: International ESA Conference on Guidance, Navigation & Control Systems*, No. GSFC-E-DAA-TN43065, 2017.