Planetary Defense Conference 2013

IAA-PDC13-04-28

Optical Navigation and Fuel-Efficient Orbit Control Around an Irregular-Shaped Asteroid

Tim Winkler[a,1,*], Brian Kaplinger[a,1], Bong Wie[a,2]

[a]*Iowa State University, 2271 Howe Hall, Ames, IA 50011-2271, USA*

Abstract

As space exploration missions become increasingly complex, there is a growing need for autonomous operation capabilities to support stringent mission requirements. For long-duration asteroid exploration missions, a key concern is how to reliably and efficiently keep the spacecraft around a target asteroid. Given the large distance from the Earth, ground commands for navigation and guidance cannot be issued in real time, and are not a practical solution. Fuel-efficient feedback control can be used to stabilize an arbitrary orbit about an asteroid provided on-board estimates of the spacecraft's position and velocity vectors with respect to the asteroid center of mass. This paper investigates methods to estimate these relative navigation state vectors using a combination of optical cameras and LIDAR technologies for an autonomous orbit control implementation.

Keywords: Optical navigation, irregular-shaped asteroids, optical flow, fuel-efficient orbit control

1. Introduction

Since the beginning of space exploration, robotic probes have been invaluable for probing into the secrets of the solar system. As our interest to explore continues to grow, spacecraft are requiring greater and greater autonomy due to the distance from Earth to be able to return the desired data. Long-duration missions close to asteroids are an example of an area which requires this autonomy due to the uncertainty and sensitivity of the surrounding dynamical environment. Reliably keeping the spacecraft in close proximity to the asteroid is no simple task as perturbing forces such as irregular gravitational fields are not well known in advance. Figure 1 illustrates graphically how an object's orbit can vary drastically for different but similar initial conditions [1]. For the orbits shown, the initial position around the asteroid is varied, and the velocity is assumed to be the local circular velocity. The first orbit starts directly on the X-axis, and remains stable for great lengths of time. The second orbit is shifted 45 degrees from the X-axis, and after nearly a week escapes the asteroid's gravity. The final orbit begins on the Y-axis, and impacts the asteroid after two orbits. Given how such an irregular gravity field affects the trajectory of a spacecraft, some form of control is necessary to ensure the spacecraft can safely execute its mission.

One option that has been thoroughly investigated is finding open-loop natural, stable orbits around these small bodies [2, 3, 4, 5, 6]. While it is possible to find stable orbits using this option, the solutions can be sensitive to orbit injection errors, and depend on prior knowledge of the body. When such information is not available, closed-loop feedback control becomes the most reliable method to ensure orbital stability. One form of active control is body-fixed hovering much as was done during JAXA's Hayabusa mission to Itokawa [7, 8]. This type of control, though, is only practical if you desire to land at a particular point on the body as the propellant requirements for hovering can be quite high. Another form of active control, as investigated by Winkler et al [1], is to employ simple proportional derivative (PD) control augmented by disturbance-accommodating filtering. It has been shown that this type of control can effectively keep a spacecraft in a stable orbit around an irregular-shaped body with very

*Corresponding author

Email addresses: twinkler@iastate.edu (Tim Winkler), bkaplin@iastate.edu (Brian Kaplinger), bongwie@iastate.edu (Bong Wie)

[1]Graduate Research Assistant

[2]Vance Coffman Endowed Chair Professor, Asteroid Deflection Research Center, Department of Aerospace Engineering

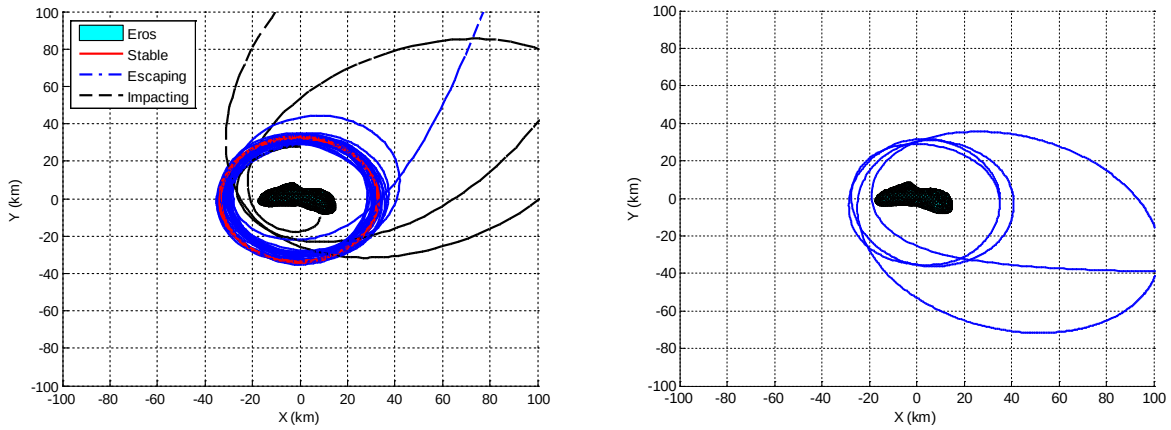


Figure 1: (Left) Comparison of stable, escaping, and impacting orbits around asteroid 433 Eros. (Right) Stable orbit with a 5 percent velocity injection error.

little control effort provided the spacecraft's position and velocity vector with respect the asteroid's center of mass can be accurately measured or estimated. The spacecraft state vector can be estimated using a combination of optical cameras and LIDAR technologies.

Optical navigation already has some basis in spacecraft guidance. One prominent example would be its previous use in the Deep Impact mission to ensure a successful impact with comet Tempel 1. Other examples include autonomous tracking and navigation tests on the Deep Space 1 and Stardust missions [9, 10, 11]. More recently, it has been announced that the future NASA mission OSIRIS-REx, which is scheduled to launch in 2016, will use a 3D Flash LIDAR camera for navigation and mapping purposes above the asteroid 1999 RQ36. Once properly developed and implemented, optical navigation and guidance methods would be applicable to a wide variety of asteroid exploration missions. For example, these capabilities could be utilized in station-keeping, hovering, pinpoint landing, sample return, etc. Autonomous optical navigation and guidance algorithms could also be employed in NASA's planned manned mission to an asteroid in 2025.

An important challenge to be addressed in optical navigation is the lighting conditions present around the target body. For an asteroid tumbling through space, only the side facing the sun will be visible excluding any areas affected by shadowing caused by geological features on the surface. Currently at the Asteroid Deflection Research Center (ADRC) of Iowa State University, an in-house program to simulate lighting conditions present around an asteroid tumbling through space has been developed and implemented on a GPU (Graphic Processing Unit) computer. With the GPU-based image models, it is possible to simulate optical measurements similar to those provided by navigation cameras and LIDAR technologies. With these estimated state vectors, a fuel-efficient and reliable orbit control strategy will be tested and evaluated for close-proximity operations around a reference irregular-shaped asteroid on a GPU computer.

2. Orbit Dynamics Around an Irregular-Shaped Asteroid

When operating in close-proximity to an asteroid, it can be advantageous to use a body-fixed, rotating coordinate system, which is fixed to the asteroid's center of mass. The z-axis of this system coincides with the axis of maximum inertia and the x- and y-axis complete an orthogonal, right-handed coordinate system. For an arbitrary angular velocity vector, the equation of motion of the spacecraft is often described by

$$\ddot{\vec{r}}|_B + \dot{\vec{\omega}} \times \vec{r} + \vec{\omega} \times (\vec{\omega} \times \vec{r}) + 2\vec{\omega} \times \dot{\vec{r}}|_B = \vec{g} + \vec{u} \tag{1}$$

where $\vec{r}$ is the spacecraft's position vector from the asteroid center of mass, $\dot{\vec{r}}|_B$ is the time derivative of $\vec{r}$ in a body-fixed rotating reference frame B, $\ddot{\vec{r}}|_B$ is the second time derivative of $\vec{r}$ in B, $\vec{\omega}$ is the angular velocity vector, $\vec{g}$ is the gravitational acceleration, and $\vec{u}$ is the applied control acceleration. For simplicity, $\vec{\omega}$ will be assumed to be constant with only a component along the axis of maximum inertia.

2.1. Gravitational Modeling

The gravitational attraction of the asteroid can be characterized using the polyhedron shape model derived by Werner and Scheeres [12]. Although computationally slower, this model can be more advantageous over more

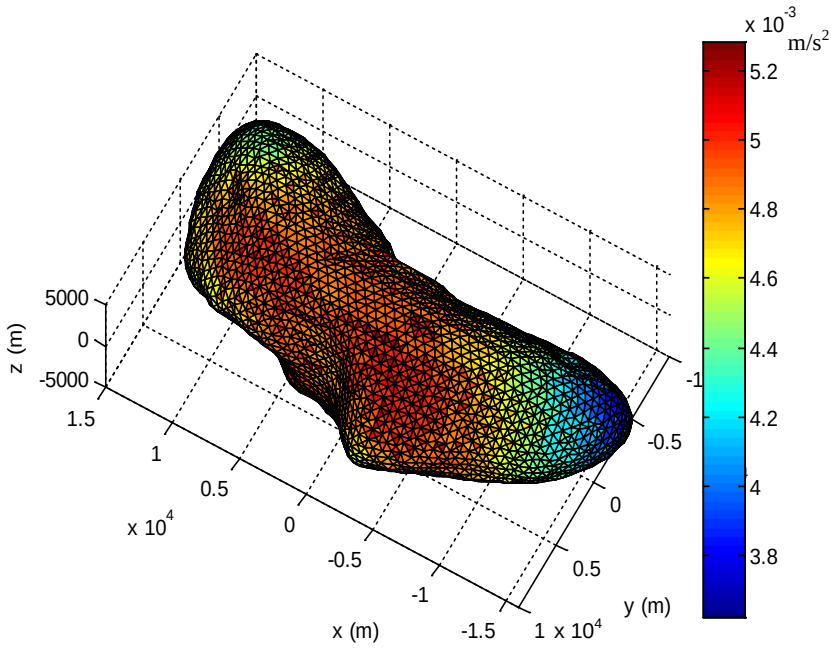


Figure 2: Surface gravitational acceleration of an irregular-shaped asteroid from the polyhedron shape model.

classical methods such as spherical harmonic expansions. One such advantage is that the gravitational field of a complex, irregular-shaped body can be readily computed given only a shape model and estimation of the constant density. The model is also valid anywhere around the body whereas spherical harmonics are limited to outside of a minimum circumscribing sphere. Figure 2 demonstrates how this model can capture variations in the gravitational field across irregular bodies.

Assuming constant density and a shape model with triangular facets, the potential function of the polyhedron model is given as

$$U = \frac{1}{2}G\rho \sum_{e\in edges} L_e \vec{r}_e \cdot \widetilde{E}_e \cdot \vec{r}_e - \frac{1}{2}G\rho \sum_{f\in facets} \omega_f \vec{r}_f \cdot \widetilde{F}_f \cdot \vec{r}_f \tag{2}$$

where G is the gravitational constant, ρ is the density, $\vec{r}$ is a vector from a point in space to an edge or facet on the polyhedron, $\widetilde{E}_e$ is an edge dyad, L_e is a dimensionless per-edge factor, $\widetilde{F}_f$ is a facet dyad, and ω_f is a dimensionless per-facet factor. $\widetilde{E}_e$ and $\widetilde{F}_f$ are given by

$$\widetilde{E}_e = \vec{n}_A \vec{n}_{21}^A + \vec{n}_B \vec{n}_{12}^B \tag{3}$$

$$\widetilde{F}_f = \vec{n}_f \vec{n}_f \tag{4}$$

where $\vec{n}_f$, $\vec{n}_A$, and $\vec{n}_B$ are outward facing unit normal vectors and $\vec{n}_{21}^A$ and $\vec{n}_{12}^B$ are outward facing edge unit normal vectors. A graphical illustration of these unit vectors is shown in Fig. 3. For a coordinate system, whose origin lies within the polyhedron, the outward facing normal vectors can be calculated using

$$\vec{n}_f = (\vec{r}_2 - \vec{r}_1) \times (\vec{r}_3 - \vec{r}_2) = \vec{r}_1 \times \vec{r}_2 + \vec{r}_2 \times \vec{r}_3 + \vec{r}_3 \times \vec{r}_1 \tag{5}$$

$$\vec{n}_{ij}^f = (\vec{r}_j - \vec{r}_i) \times \vec{n}_f \tag{6}$$

where $\vec{r}_1$, $\vec{r}_2$, and $\vec{r}_3$ are vectors defining the position of the facet vertices. To ensure that the facet normal vector is outward facing, the dot product of the normal with any of the three vertex vectors. If the result is positive, the vector is outward facing. Besides the normal vector is facing inwards, it should be multiplied by -1 to give the correct direction. Next the dimensionless factors L_e and ω_f are given as

$$L_e = \ln \frac{r_i + r_j + e_{ij}}{r_i + r_j - e_{ij}} \tag{7}$$

$$\omega_f = 2 \arctan \frac{\vec{r}_i \cdot \vec{r}_j \times \vec{r}_k}{r_i r_j r_k + r_i(\vec{r}_j \cdot \vec{r}_k) + r_j(\vec{r}_k \cdot \vec{r}_i) + r_k(\vec{r}_i \cdot \vec{r}_j)} \tag{8}$$

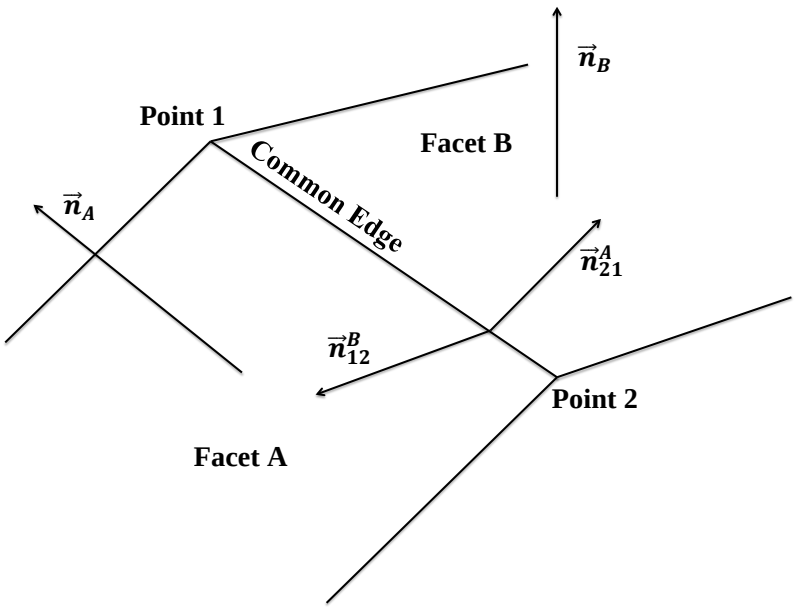


Figure 3: Illustration of the unit facet normal and unit edge normal vectors.

where e_{ij} is the length of the edge connecting $\vec{r}_i$ and $\vec{r}_j$. The gradient of the potential in Equation 2 is then

$$\nabla U = -G\rho \sum_{e \in edges} L_e \tilde{E}_e \cdot \vec{r}_e + G\rho \sum_{f \in facets} \omega_f \tilde{F}_f \cdot \vec{r}_f \tag{9}$$

The derivation of this model also leads to a simple method for determining whether a point is inside or outside of the polyhedron by means of the Laplacian of the potential function.

$$\nabla^2 U = -G\rho \sum_{f \in facets} \omega_f \tag{10}$$

The sum of ω_f goes to zero outside of the polyhedron, and 4π inside the polyhedron. This term can be calculated at little additional cost as ω_f is already required in Equation 8.

3. Optical Guidance and Navigation

3.1. Rendering the Simulated Camera Data

3.1.1. 3D Target Polygon Model

This section addresses the rendering of the images in simulation memory that correspond to camera data to be processed by the spacecraft. We begin with a triangulation of surface points represented as a 3D wireframe polygon model. The target model is derived from data for the asteroid 433 Eros [13] collected by the NEAR mission, and has 200,700 faces. This number of faces corresponds to the variable n_f of the computer model. Generally, the number of vertices needed to fill out the model, n_v, is strictly less than n_f, so it is more computationally efficient to do calculations on the vertices where possible. However, much of our information depends in some sense on an orientation of the body surface, for which the faces are necessary. A connectivity array stores the relationships between each set of vertex data and which face it applies to. If $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$ are the position vectors for the vertices of a face, as shown in Figure 4, then we can define for $1 \leq i \leq n_f$:

$$\mathbf{X}_i = \mathbf{R}_2 - \mathbf{R}_1 \qquad \mathbf{Y}_i = \mathbf{R}_3 - \mathbf{R}_1 \qquad \mathbf{Z}_i = (\mathbf{R}_1 + \mathbf{R}_2 + \mathbf{R}_3)/3 \tag{11}$$

The unit normal vectors of each face are calculated and stored using

$$\mathbf{N}_i = \mathbf{X}_i \times \mathbf{Y}_i, \qquad \mathbf{N}_i \cdot \mathbf{Z}_i > 0 \tag{12}$$

The wireframe model uses a known rotation state of the target to propagate measurements forward in time. A rotation matrix is applied to the initial conditions so that a simulated state is known for the 3D model. A measurement of this rotation state is assumed to have been made by optical or other systems prior to terminal operations. To complete this system description, we require a unit sun vector, $\widehat{\mathbf{S}}$, in the direction of the sun from the target and a unit view vector, $\widehat{\mathbf{V}}$, in the direction of the spacecraft from the target. We assume that the true center of mass of the target is at the origin on this coordinate system.

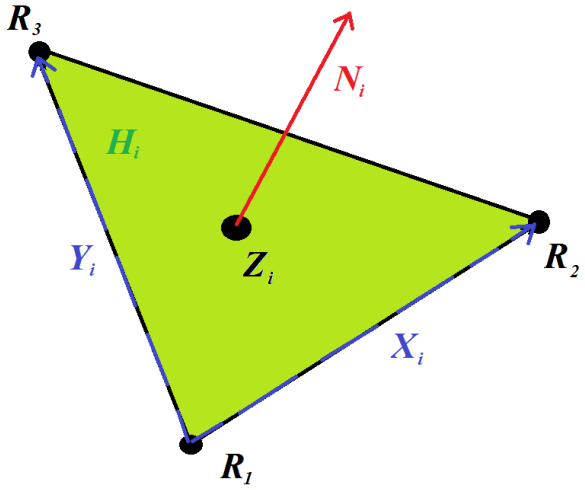


Figure 4: Facet geometry definitions.

3.1.2. Camera Pointing and Focus Plane

In order to decouple the attitude and translation mechanics of the present simulation system, a perfect pointing was adopted for the camera. Thus, the choice of estimated center of mass or previously computed center of brightness is at the center of the camera focal plane, with the camera needing zero time to adjust to the new settings. Modifications to this scheme, including integration with spacecraft attitude are expected in the near future.

The camera focal plane distance was chosen arbitrarily as the distance from the spacecraft to target. Since this plane is perpendicular to $\widehat{\mathbf{V}}$ we can project the 3D model onto this plane by subtracting the component along this vector, thus the projected locations of the vertices are

$$\mathbf{P} = \mathbf{R}_j - \widehat{\mathbf{V}}, \qquad 1 \leq j \leq 3 \tag{13}$$

To save computational time and storage space, a condensed array of these values are computed only for the faces satisfying $\mathbf{N}_i \cdot \widehat{\mathbf{V}} > 0$. This results in only the faces visible to the spacecraft. Due to the decoupling of the camera dynamics, a direct orientation of this plane is not possible given the currently available information. Thus, a direction for camera up must be chosen. We choose unit vectors $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{W}}$ such that

$$\widehat{\mathbf{W}} = \widehat{\mathbf{S}} \times \widehat{\mathbf{V}}, \qquad \widehat{\mathbf{U}} = -\widehat{\mathbf{V}} \times \widehat{\mathbf{W}} \tag{14}$$

Therefore, the sunlight will always come into frame from a horizontal direction, and up will always be orthogonal to the plane containing the target, the sun, and the spacecraft. The coordinates of each vertex in this plane can be computed as

$$\widetilde{\mathbf{P}}_j = [u_j \;\; w_j]^T = [\mathbf{P}_j \cdot \widehat{\mathbf{U}} \quad \mathbf{P}_j \cdot \widehat{\mathbf{W}}]^T \tag{15}$$

At this point, what is visible to the spacecraft depends on camera parameters that interpret the real system. The resolution and field of view for the cameras simulated are listed in Table 1. At each time step, these are used to compute the half resolution of the image plane, R_h. If $\mathbf{R}_T$ is the position vector representing the spacecraft in the target frame, then this can be computed as

$$R_h = |\mathbf{R}_T| \tan(f_v/2) \tag{16}$$

where f_v represents the field of view in radians (assumed isotropic). If x_r and y_r represent the x and y resolutions in the camera 2D pixel frame, then the information represented by each pixel corresponds to a size of $2R_h/x_r$ in the horizontal direction and $2R_h/y_r$ in the vertical direction.

Table 1: Simulation Camera Parameters

	Medium Resolution	Infrared	Flash LIDAR
Resolution (x_r, y_r) (pixels)	1024×1024	512×512	32×32
Field of View (f_v) (radians)	10.0×10^{-3}	10.0×10^{-3}	52.4×10^{-3}
Pixel Size at 15 km (m)	0.15	0.29	24.6

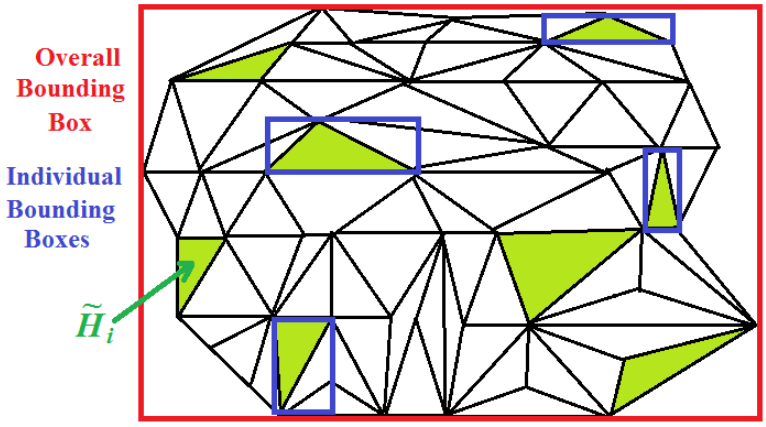


Figure 5: Parallel trim reduction technique.

3.1.3. Lighting Conditions

A simple flat shading model is used to calculate the brightness of the target surface. Given a diffuse lighting coefficient, k_d, and an ambient lighting coefficient, k_a, the corresponding brightness attributed to each face of the 3D model is

$$C_i = \begin{cases} k_d(\mathbf{N}_i \cdot \widehat{\mathbf{S}}) + k_a, & \mathbf{N}_i \cdot \widehat{\mathbf{S}} > 0 \\ 0, & \mathbf{N}_i \cdot \widehat{\mathbf{S}} \leq 0 \end{cases} \tag{17}$$

where a value C_i of 1 corresponds to perfect reflectivity of the incident sunlight at that distance. A greyscale colormap is used for human visualization. The diffuse coefficient works closely in line with the albedo, and is chosen to be 0.25 for the simulations. Since the cameras we are using have very little time to resolve detail of the body, we assume that their lower threshold for brightness is very low and therefore $k_a = 0$. This model is very fast, and can be computed completely in parallel for each face.

When additional lighting detail is desired, the following interpolation model due to Gouraud [14] is used. Let each vertex have a normal $\widetilde{\mathbf{N}}_1$, $\widetilde{\mathbf{N}}_2$ and $\widetilde{\mathbf{N}}_3$ computed as an average of the adjacent faces, found from inverting the connectivity array. Then, a corresponding value for brightness at each vertex $\widetilde{C}_1$, $\widetilde{C}_2$, and $\widetilde{C}_3$ is computed using the flat shading model equation. If a pixel representing the image plane at $[u \quad w]^T$ is calculated to intercept this triangle, then an inverse squared weighted average is assigned to the pixel as

$$C(u, w) = \frac{\alpha_1}{\alpha_T}\widetilde{C}_1 + \frac{\alpha_2}{\alpha_T}\widetilde{C}_2 + \frac{\alpha_3}{\alpha_T}\widetilde{C}_3 \tag{18}$$

where $\alpha_T = \alpha_1 + \alpha_2 + \alpha_3$ and

$$\alpha_j = \frac{1}{(u - u_j)^2 + (w - w_j)^2}, \qquad 1 \leq j \leq 3 \tag{19}$$

3.1.4. Pixel Value Assignment

For either the flat or interpolated shading models, the facet at which a pixel intercepts the target projection must be determined. This is done on the GPU in parallel to reduce computational time, especially using a higher fidelity target model. Each pixel is assigned coordinates in the u, w plane representing the center of the box over which the pixel is integrated. This works well when the size of a pixel is on the scale of the surface facets, but an averaged value needs to be used at greater distances. To determine the intercepted triangle, we first compute the upper and lower bounds of each visible triangle in parallel. Then, a reduction is performed to get the outer limits of the target in the UW system. This process is represented in Figure 5. Pixels outside of this range can automatically be assigned a value of 0. In fact, only the location and value of nonzero image components are stored in the present computational model.

The arrays containing the limits of each facet are sorted, and then a scan of these arrays is completed. This lets us assign a vector of possible intersections for each pixel. This vector contains the IDs of all facets through which the pixel passes through the bounding box. This is generally a low number. For each candidate intersection, the convex hull of the triangle, H_i, is identified, as shown in Figure 4. This is projected onto the UW plane, forming the set $\widetilde{H}_i$. If $\widetilde{\mathbf{X}}_i$ and $\widetilde{\mathbf{Y}}_i$ are the projections of $\mathbf{X}_i$ and $\mathbf{Y}_i$ in the UW plane, then the coordinates of the pixel can be

expressed as

$$\begin{bmatrix} u \\ w \end{bmatrix} = \widetilde{\mathbf{P}}_1 + d_1\widetilde{\mathbf{X}}_i + d_2\widetilde{\mathbf{Y}}_i \tag{20}$$

which can be viewed as a transformation into the affine system defined by these two vectors. It is a well-known result from geometry that the point $[u \;\; w]^T$ lies within the triangle if $d_1 > 0$, $d_2 > 0$, and $d_1 + d_2 < 1$. This generally assigns a single facet as the possibility for intersection. However, at pixels near the boundary of the target, several potential intersections may occur. In these cases, the closest triangle ($\mathbf{Z}_i \cdot \widehat{\mathbf{V}}$ largest) is chosen.

For the 3D LIDAR measurement, we interpolate based on the distances to the vertices of the intercepted triangle in the UW plane. Using the same weighting constants as the interpolated brightness model, we compute

$$\mathbf{L}_k = \frac{\alpha_1}{\alpha_T}\mathbf{R}_1 + \frac{\alpha_2}{\alpha_T}\mathbf{R}_2 + \frac{\alpha_3}{\alpha_T}\mathbf{R}_3 = [x_k \quad y_k \quad z_k]^T \tag{21}$$

where $\mathbf{L}_k$ is a $k-$th vector measurement of the reflection in Cartesian coordinates centered on the target. We also have $1 \leq k \leq n_p$, where the total number of measurements, n_p, grows with time.

3.2. Estimation of Target Reference Frame

Flash LIDAR measurements are desired as a way to reinforce optical measurements of target size and shape data. Unfortunately, we must be able to interpolate this from undersampled areas of the target. The present simulation model builds a least-squares energy ellipsoid representing the LIDAR measurements. Let $\mathbf{A}$ be a symmetric, positive definite matrix formed by the vector $\mathbf{a}$

$$\mathbf{a} = [a_1 \;\; a_2 \;\; a_3 \;\; a_4 \;\; a_5 \;\; a_6]^T, \qquad \mathbf{A} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_4 & a_5 \\ a_3 & a_5 & a_6 \end{bmatrix} \tag{22}$$

and E represent the quadratic form produced by $\mathbf{A}$

$$E_k = \mathbf{L}_k^T\mathbf{A}\mathbf{L}_k = a_1x_k^2 + 2a_2x_ky_k + 2a_3x_kz_k + a_4y_k^2 + 2a_5y_kz_k + a_6z_k^2 \tag{23}$$

Then, for nonzero integer values, m, the residual function defined as

$$\mathbf{r}_k = r(\mathbf{L}_k) = E^m - 1 \tag{24}$$

represents an energy measure of distance from the ellipsoid $E = 1$. Various values of m have interesting properties and are readily solved by nonlinear least squares methods. We chose the Levenberg-Marquardt method with line search, requiring QR factorization of the Jacobian at each linearization [15]. Results are shown for the case of $m = 1$. In this case, the residual is a linear function of the parameter vector $\mathbf{a}$, reducing the problem to a linear least squares parameter optimization. The rows of the constant Jacobian are readily computed as

$$\mathbf{J}_k = \frac{\partial \mathbf{r}_k}{\partial \mathbf{a}} = [x_k^2 \quad 2x_ky_k \quad 2x_ky_k \quad y_k^2 \quad 2y_kz_k \quad z_k^2] \tag{25}$$

This process works exceptionally for random input data. However, using simulated LIDAR measurements results in an overshoot by an order of magnitude followed by a slow correction process. Due to measurements being taken at close angular separation, heteroscedasticity in the input data is observed. This results from oversampling in some regions of the target and vastly undersampling other regions. Therefore, using a conditional variance, Ω, of the expected value $-\mathbf{r}_k$ given $\mathbf{J}_k$ a weighted least squares problem is defined as

$$\min_A \sum \mathbf{r}_k^T\Omega^{-1}\mathbf{r}_k \tag{26}$$

Assuming that Ω is diagonal, and that individual measurements are not dependent on one another, we can form the solution parameter vector

$$\mathbf{a} = -(\mathbf{J}^T\Omega^{-1}\mathbf{J})^{-1}\mathbf{J}^T\Omega^{-1}\mathbf{r}_k \tag{27}$$

Choosing a weighting scheme such that

$$\omega_k = \Omega_{kk}^{-1/2} \tag{28}$$

Choosing a weighting scheme such that

$$\tilde{\mathbf{r}}_k = \omega_k\mathbf{r}_k \qquad \widetilde{\mathbf{J}}_k = \omega_k\mathbf{J}_k \tag{29}$$

While initial attempts to determine weights used a measure of data density to determine variance, it was quickly determined that this was not an ideal implementation. For large numbers of measurements, the density calculation is equivalent to a nearest neighbor search, which scales at best as $n_p \log n_p$ and at worst as n_p^2. The formulation is particularly vulnerable to the weighting problem due to E being positive definite. Therefore, under the original approach $|\mathbf{r}_k| < 1$ for all $\mathbf{r}_k < 0$, while $|\mathbf{r}_k|$ is unbounded for $\mathbf{r}_k > 0$. This tends to result in overshoot of the longest ellipsoid axis to minimize errors in oversampled regions, while allowing for large physical deviations for points with $E < 1$. Taking into account this issue, a weighting scheme to force inclusion of these points was used as

$$\omega_k = \frac{1}{1 + E_k} \tag{30}$$

Once suitable values for the matrix $\mathbf{A}$ are found, the representative ellipsoid can inform us of the target orientation vector frame. We have the eigenvector decomposition

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T \tag{31}$$

where the elements of the matrix $\mathbf{D}$ contain the eigenvalues of $\mathbf{A}$, which correspond to the ellipsoid semiaxis parameters, a, b, and c, as follows:

$$\mathbf{D} = diag\left(\frac{1}{a^2}, \frac{1}{b^2}, \frac{1}{c^2}\right) \tag{32}$$

and the matrix $\mathbf{V}$ contains the unit eigenvectors of $\mathbf{A}$, which corresponds to the new target coordinate frame as

$$\mathbf{V} = [\hat{\mathbf{b}}_1 \;\; \hat{\mathbf{b}}_2 \;\; \hat{\mathbf{b}}_3] \tag{33}$$

This coordinate frame can then be used for orbit determination and operations, and the combined measurements can be used for a surface bound estimate to the first order.

3.3. Velocity Estimation

3.3.1. Landmark Tracking

Landmark tracking is a method that can be used to aid in optical navigation for estimating the velocity of an orbiting spacecraft [16, 17]. To detect features on the surface, we first divide the optical image into smaller search windows containing M pixels. Now a matrix $\mathbf{C}$ is defined base upon the partial derivatives of the image intensity at each pixel k as

$$\mathbf{C} = \begin{bmatrix} \sum_{k=1}^{M} I_x^2(k) & \sum_{k=1}^{M} I_x(k)I_y(k) \\ \sum_{k=1}^{M} I_x(k)I_y(k) & \sum_{k=1}^{M} I_y^2(k) \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{34}$$

where I_x and I_y are the partial derivatives of the pixel intensity function $I(x, y)$ of the image in the x- and y-directions respectively. These partial derivatives are given by

$$I_x(x, y) = \frac{I(x + 1, y) - I(x - 1, y)}{2} \tag{35a}$$

$$I_y(x, y) = \frac{I(x, y + 1) - I(x, y - 1)}{2} \tag{35b}$$

The search window can be considered a potential feature if the two eigenvalues of $\mathbf{C}$ are greater than some predefined threshold value λ_t. This criteria can also be modified so as to avoid solving the eigenvalue problem, as follows:[16, 18]

$$(c_{11} - \lambda_t) \cdot (c_{22} - \lambda_t) > 0 \quad and \quad c_{11} > \lambda_t \tag{36}$$

Once a feature has been identified used the above condition, it must then be found in the subsequent frame. This can be accomplished using the Shi-Tomasi-Kanade feature tracker, which attempts to locate the same feature between successive images by comparing the intensity differences between the search windows in the two frames, as follows:

$$\epsilon = \sum_{k=1}^{M} (J(k) - I(k))^2 \tag{37}$$

When ϵ reaches a minimum value, the two search windows should be centered on the same feature. As the spacecraft is orbiting above the asteroid, new features will have to be detected and tracked every two images as the previous feature may no longer be suitable for tracking purposes, or has disappeared from the image.

3.3.2. Optical Flow Estimation

Once the landmark has been identified in both images, the 2D optical flow, or image velocity estimate, is computed. Many algorithms for optical flow have been proposed and evaluated [19, 20, 21]. A usual starting point is to assume that the intensity values between the two frames is conserved as

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{38}$$

where u and v are the two components of the optical flow. In actuality, this assumption is not always true, but still works remarkable well. As the intensity is assumed to be conserved, the time derivative should become zero, as follows:

$$\frac{d}{dt}I(x, y, t) = \frac{\partial I}{dx}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = I_x u + I_y v + I_t = 0 \tag{39}$$

Equation 39 is referred to as the intensity conservation constraint, and serves as the primary equation for recovering the optical flow field. This constraint is then applied to every pixel in the search window containing the landmark to be tracked, which are then gathered into a squared sum of the form as

$$E(u, v) = \sum_{x,y} g(x, y)[I_x u + I_y v + I_t]^2 \tag{40}$$

where $g(x, y)$ is a weighting function for each pixel in the window. It is common to let $g(x, y)$ be a Gaussian distribution which gives higher influence to terms near the center of the window. Equation 40 is now an over constrained problem to solve for the optical flow. Thus u and v will be found using a least squares estimate of velocity. First take the partial derivatives of Equation 40 as

$$\frac{\partial E(u, v)}{\partial u} = \sum_{x,y} g(x, y)[I_x^2 u + I_x I_y v + I_x I_t] = 0 \tag{41a}$$

$$\frac{\partial E(u, v)}{\partial v} = \sum_{x,y} g(x, y)[I_y^2 v + I_x I_y u + I_y I_t] = 0 \tag{41b}$$

These equations can then be rewritten in matrix form as

$$\mathbf{Mu} + \mathbf{b} = 0 \quad \textit{where} \quad \mathbf{M} = \begin{bmatrix} \sum_{x,y} g(x, y)I_x^2 & \sum_{x,y} g(x, y)I_x I_y \\ \sum_{x,y} g(x, y)I_x I_y & \sum_{x,y} g(x, y)I_y^2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \sum_{x,y} g(x, y)I_x I_t \\ \sum_{x,y} g(x, y)I_y I_t \end{bmatrix} \tag{42}$$

The least-squares estimate then becomes

$$\hat{\mathbf{u}} = -\mathbf{M}^{-1}\mathbf{b} \tag{43}$$

The 2D optical flow velocity can then be related to the 3D motion of the object via

$$\vec{V} = -\vec{T} - \vec{\omega} \times \vec{P} \tag{44}$$

where $\vec{T}$ is the instantaneous translational motion of the object being tracked, $\vec{\omega}$ is the instantaneous rotation, and $\vec{P}$ is the 3D position of the landmark. Using the pinhole camera model where $x = f\frac{X}{Z}$ and $y = f\frac{Y}{Z}$ and Equation 44, we obtain

$$u = \frac{xT_z - fT_x}{Z} + \omega_x\left(\frac{xy}{f}\right) - \omega_y\left(f + \frac{x^2}{f}\right) + \omega_z y \tag{45a}$$

$$v = \frac{yT_z - fT_y}{Z} + \omega_x\left(f + \frac{y^2}{f}\right) - \omega_y\left(\frac{xy}{f}\right) + \omega_z x \tag{45b}$$

where f is the focal length of the camera. In order to completely recover the 3D velocity, a minimum of three landmarks will have to be tracked simultaneously in order to give the six required equations to solve for $\vec{T}$ and $\vec{\omega}$.

3.4. Feedback Control

3.4.1. Proportional Derivative (PD) Control

Proportional derivative (PD) control is a common, simple form of full-state feedback control logic, which seeks to minimize the error between some chosen variable and its derivative to their desired reference values by feeding back the difference multiplied by some controller gain. To implement this control, the equations of motion are first expressed in state-space form. As the gravitational terms from the polyhedron shape model are highly nonlinear, they are included as disturbances $\mathbf{w}$ to give a simple dynamical model as

$$\dot{\mathbf{x}}_p = \mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\mathbf{u} + \mathbf{w} \tag{46}$$

where $\mathbf{x}_p$ is the state vector containing the position and velocity estimates. The control input is then assumed to be of the form

$$\mathbf{u} = -\mathbf{K}(\mathbf{x}_p - \mathbf{x}_r) \tag{47}$$

where $\mathbf{K}$ is the control gain matrix to be determined, and $\mathbf{x}_r$ is the reference state vector.

3.4.2. Disturbance-Accommodating Control

Disturbance-accommodating or disturbance-rejection control is a technique which can be used to reduce control effort in the presence of persistent disturbances [22, 23]. This is accomplished by eliminating periodic portions of the control accelerations, which allows the spacecraft to follow a trajectory that is closer to a natural periodic solution of the nonlinear equations of motion. These periodic accelerations arise because we are forcing the spacecraft to follow a particular reference orbit in a spinning, nonlinear gravity field.

To apply this control logic, we first allow the spacecraft to orbit the asteroid using only PD control. After some length of time, the control accelerations are put through a Fast Fourier Transform (FFT) function. The FFT plots along each axis exhibit spectral components, which are being repeatedly damped out via the control accelerations. After identifying these frequencies, we can design periodic disturbance-accommodating filters of the form

$$\ddot{\alpha}_i + \omega_{xi}^2 \alpha_i = u_x \tag{48a}$$

$$\ddot{\beta}_i + \omega_{yi}^2 \beta_i = u_y \tag{48b}$$

$$\ddot{\gamma}_i + \omega_{zi}^2 \gamma_i = u_z \tag{48c}$$

where ω_{xi}, ω_{yi}, and ω_{zi} represent the ith frequency component in each axis, and α_i, β_i, and γ_i are the ith filter states along the x-, y-, and z-axes respectively. For constant disturbances, the filters take the form:

$$\dot{\alpha}_x = u_x \tag{49a}$$

$$\dot{\beta}_y = u_y \tag{49b}$$

$$\dot{\gamma}_z = u_z \tag{49c}$$

After identifying all the frequencies present in the FFT plots, the filter can also be expressed in state-space form as

$$\dot{\mathbf{x}}_d = \mathbf{A}_d \mathbf{x}_d + \mathbf{B}_d \mathbf{u} \tag{50}$$

where $\mathbf{x}_d$ is the disturbance filter state vector. The state-space forms in Equations 46 and 50 can then be combined to produce

$$\begin{bmatrix} \dot{\mathbf{x}}_p \\ \dot{\mathbf{x}}_d \end{bmatrix} = \begin{bmatrix} \mathbf{A}_p & 0 \\ 0 & \mathbf{A}_d \end{bmatrix} \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_d \end{bmatrix} + \begin{bmatrix} \mathbf{B}_p \\ \mathbf{B}_d \end{bmatrix} \mathbf{u} \tag{51}$$

The new control input for the augmented system is then obtained as

$$\mathbf{u} = -\mathbf{K} \begin{bmatrix} \mathbf{x}_p - \mathbf{x}_r \\ \mathbf{x}_d \end{bmatrix} \tag{52}$$

The process is then repeated in an iterative fashion should any new frequencies arise in the FFT plots of the control accelerations due to suppressing certain frequency components. Using this approach, a robust disturbance-accommodating controller can be designed that results in a drastic reduction of ΔV.

3.4.3. Linear Quadratic Regulator Control Design

One way to find the control gain matrix $\mathbf{K}$ is using the linear quadratic regulator (LQR) method [22, 23]. This method works by calculating a $\mathbf{K}$ that minimizes the following optimal control performance index

$$J = \frac{1}{2} \int_0^\infty (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \tag{53}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are user-chosen weighting matrices. Once selected, the control gain matrix $\mathbf{K}$ is obtained as

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{X} \tag{54}$$

where $\mathbf{X}$ is the symmetric positive semidefinite solution of the algebraic Riccati equation (ARE) of the form

$$\mathbf{A}^T \mathbf{X} + \mathbf{X} \mathbf{A} - \mathbf{X} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{X} + \mathbf{Q} = 0 \tag{55}$$

4. Simulations and Results

The results section can be separated into two distinct categories. The first category demonstrates the effectiveness of disturbance-accommodating control given information about the relative position and velocity of the spacecraft with respect to the target asteroid. This will be done for the case of a polar orbit around a irregularly-shaped, spinning asteroid. The second category tests and evaluates the optical methods for estimating these quantities. We first estimate the physical parameters of a small asteroid such as size and mass before estimating the spacecraft's velocity using optical flow.

4.1. Closed-Loop Simulation

To demonstrate the effectiveness of disturbance-accommodating filtering, a spacecraft is placed into a polar orbit around the large, rotating near-Earth asteroid 433 Eros. A polar orbit was selected as it can be considered one of the more useful orbits in terms of scientific return and imaging. This is due to the fact that a polar orbit would be able to view the entire surface of the body due to the asteroid's spin, whereas lower inclined or XY-plane orbits would be limited viewing only parts of the surface. To enable this simulation, a shape model for 433 Eros produced by the NEAR Shoemaker mission was used [24] as well as the estimated density of the asteroid of 2.67 g/cm^3. The controlled orbit using PD control and disturbance-accommodating filtering is plotted in the body-fixed frame and shown in Figure 6. The plots for the corresponding control inputs and FFT plots are then given in Figures 7 and 8. Observing the trajectories in Figure 6, we can see that the spacecraft is constrained to a spherical surface with only PD control. Applying disturbance-accommodating filters does not change the shape of the orbit drastically, though slight differences are present. Looking at the control inputs, a more noticeable difference can be observed. With only PD control, the x- and y-control inputs seem to oscillate around a predominant sinuisoid with occasional large spikes, that are twice as large as the smaller peaks surrounding them. Integrating the control inputs results in a total per week ΔV requirement of 16.59 m/s. Then using the FFT plots to identify the spectral

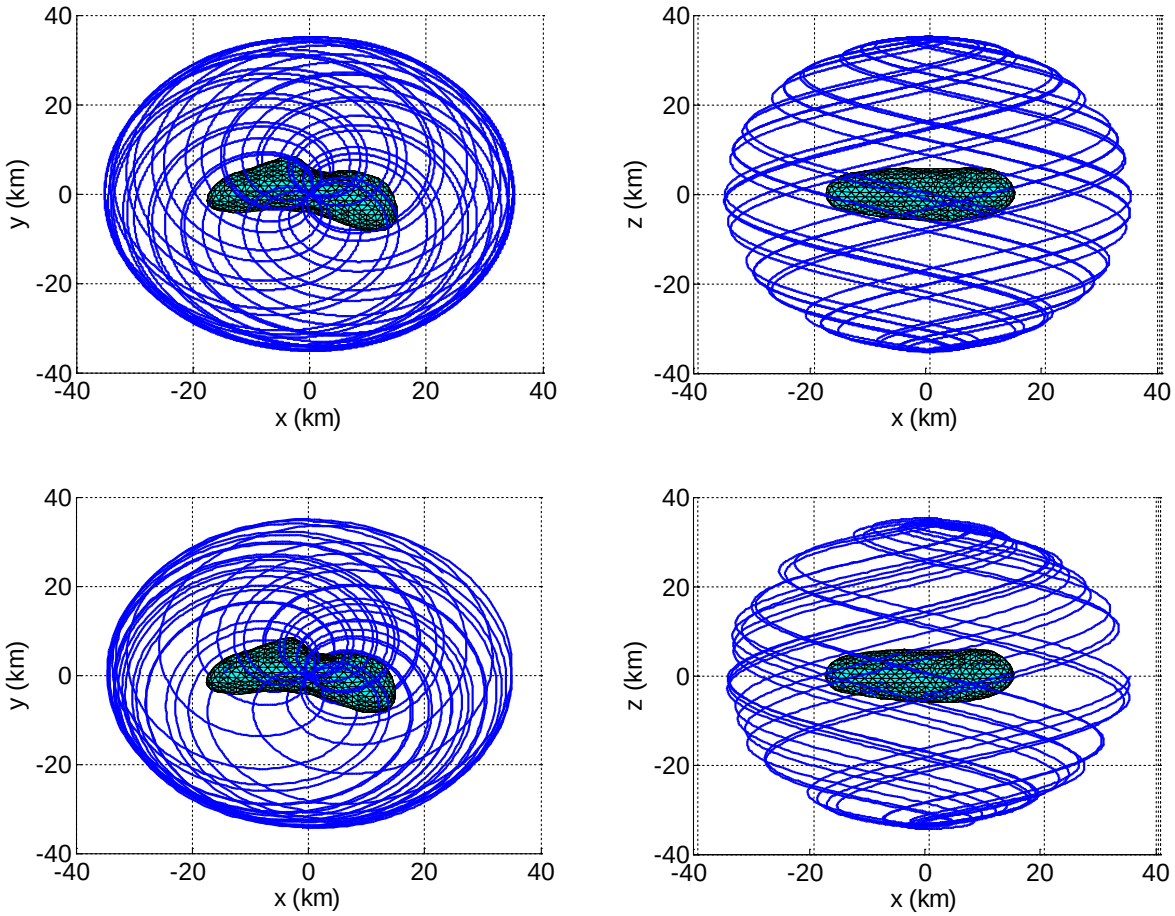


Figure 6: Polar orbit in the body-fixed frame around a large, irregular-shaped asteroid. The orbit with only PD control is shown on top and disturbance-accommodating filtering on the bottom.

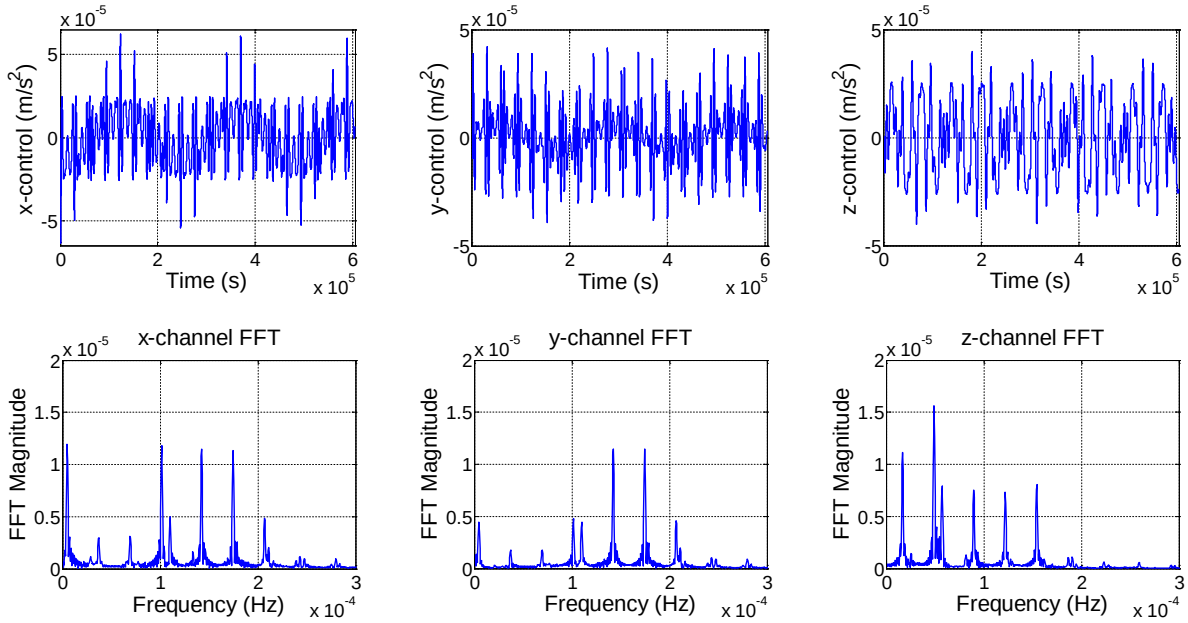


Figure 7: Control inputs and their associated FFT plots in the body-fixed frame for PD control.

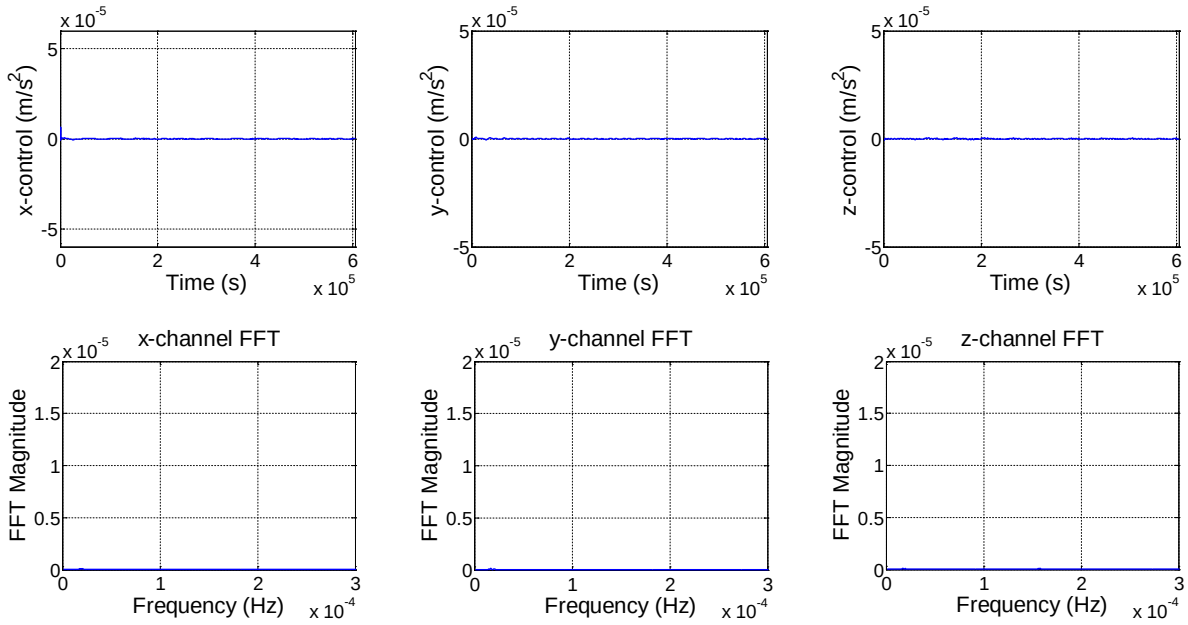


Figure 8: Control inputs and their associated FFT plots in the body-fixed frame for disturbance-accommodating control.

components of the control we wish to filter out, disturbance-accommodating filters can be created to obtain the new control inputs in Figure 8. As can be seen, the prominent frequencies have been suppressed, which reduces the ΔV requirements around 0.13 m/s per week.

4.2. Asteroid Shape Determination

Using simulated 3D Flash LIDAR measurements, Figure 9-11 shows the asteroid shape and coordinate frame estimation for a small, irregular body with a maximum radius near 300 m and known rotation rate. While we are able to converge to an estimate for the mass of the asteroid, the solution takes a long time to converge. On the other hand, the ellipsoid axes estimates of the body and the body-fixed vector estimates converge much quicker to an accurate, stable solution. These estimations of the shape and coordinate frame can then be used for position and velocity estimation in the body-fixed frame.

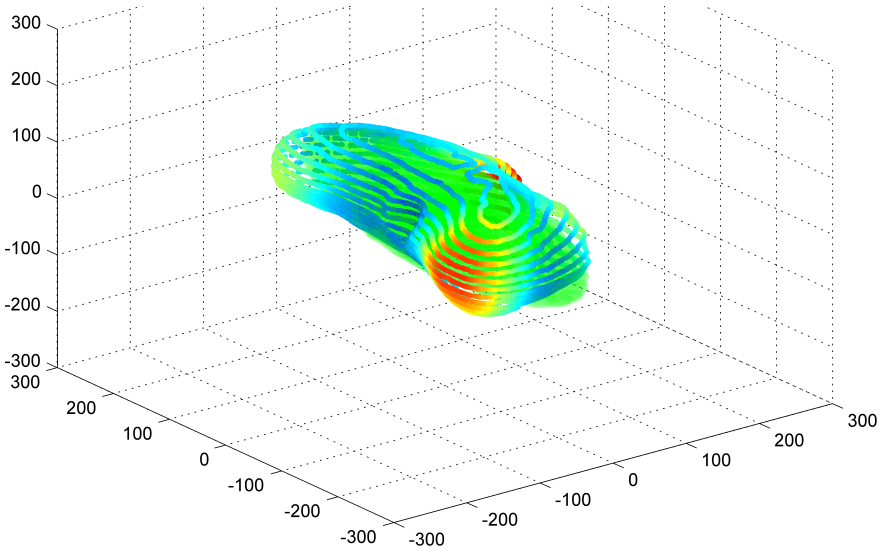


Figure 9: Surface estimation using LIDAR measurements.

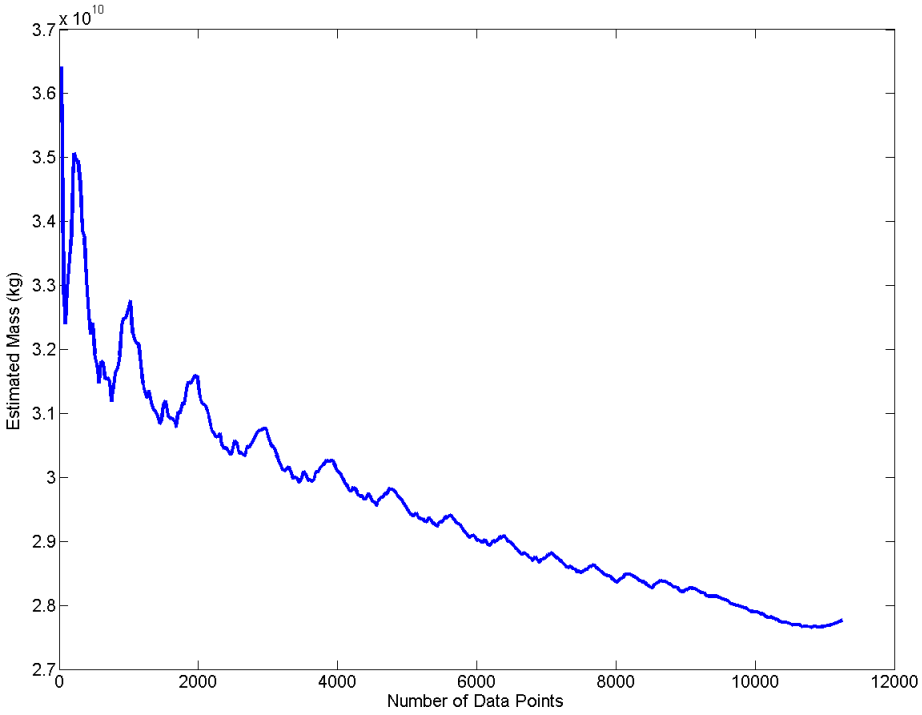
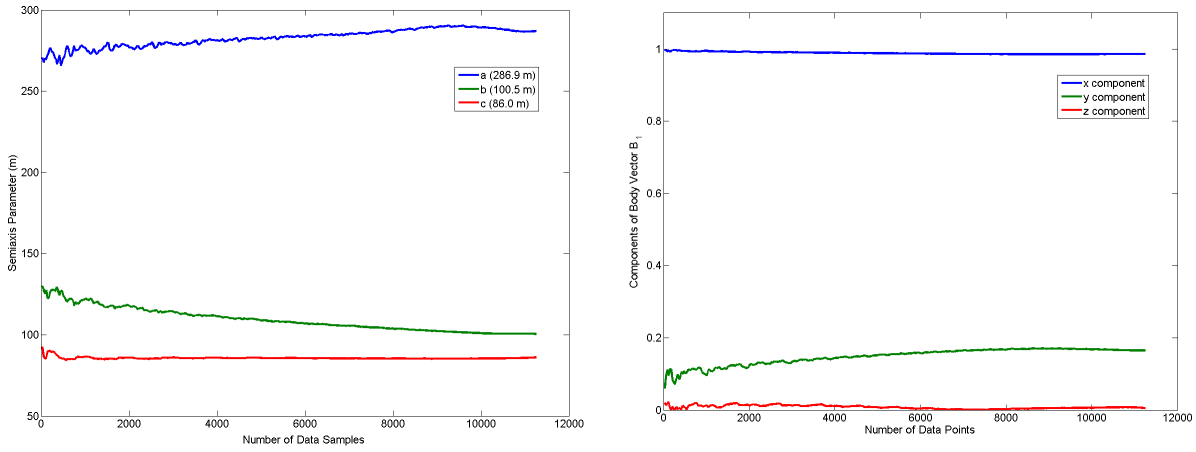


Figure 10: Asteroid Mass Estimation.



(a) Ellipsoid Axes Estimate.

(b) Body-Fixed Vector Estimate.

Figure 11: Asteroid Shape Parameters Estimation.

4.3. Landmark Tracking and Velocity Estimation Results

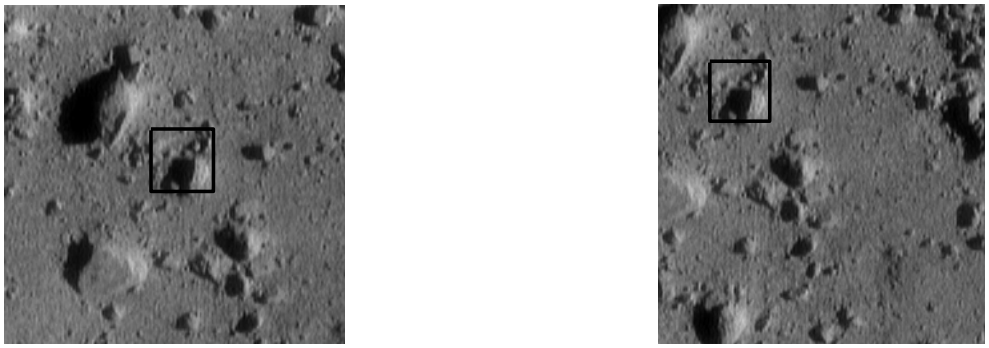
4.3.1. Landmark Detection and Tracking Evaluation

As a test run, the landmark detection and tracking algorithm was tested using images taken by the NEAR Shoemaker mission during its descent onto the surface of 433 Eros. The full image is 256x256 pixels, and the search window is 48x48 pixels. The motion of the orbiting satellite is simulated by using two different portions of the same image which both contain the landmark to be tracked. Figure 12 shows the first image with one cluster of rocks identified as the best landmark in the image to track. The second image then locates this feature using only the intensity functions of the images.

As a secondary test, the same algorithms were then applied to the simulated camera images of an asteroid. The spacecraft is put into a 35 km circular orbit about a rotating model of 433 Eros. The exact position of the spacecraft in the body-fixed frame was not important here as long as the cameras could view the lit portion of the asteroid surface. The image is a full 1024x1024 pixels with a search window size of 64x64 pixels. From the angle shown in Figure 13, the landmark detection algorithm tended to choose a point right on the edge of the asteroid body where the intensity change is greatest. While such a location may be adequate for tracking, the optical flow in these regions is not always consistent with the motion of the body as will be seen later. Therefore for illustrative purposes, a preselected crater was chosen to be tracked until the feature was lost. Up to 25 seconds, the program is able to locate the same crater in subsequent images. By 26 seconds, though, the feature tracking fails by a large degree to locate the same crater due to the changes in the images becoming too great for the intensity conservation assumption to handle. Figure 14 shows how the squared minimum tracking error from Equation 37 for this example changes over time. This plot grows in a nearly linear fashion up until feature tracking fails around 26 seconds. Here there is a noticeable dip before starting to curve back up.

4.3.2. Velocity Estimation

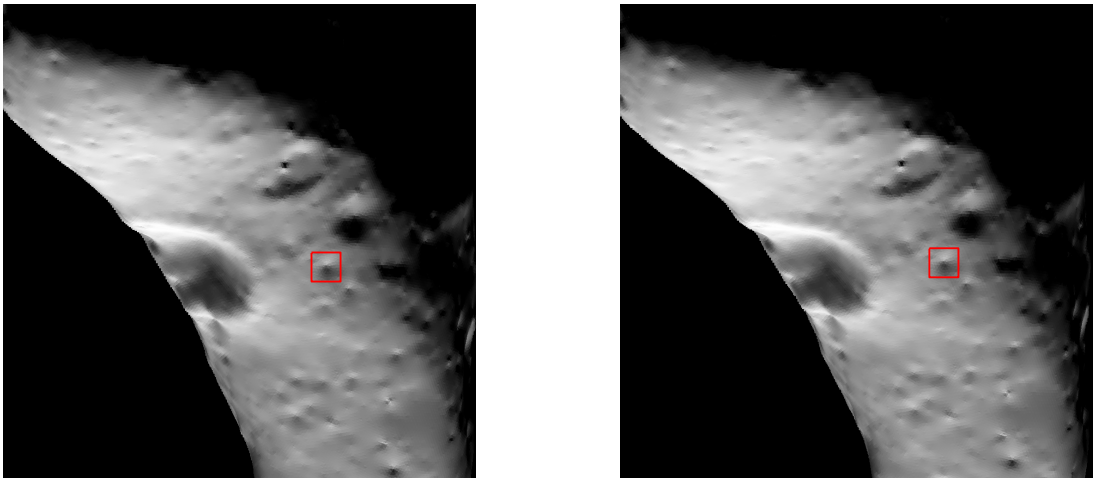
By tracking features on the body using optical images, the instantaneous velocity of the spacecraft will now be estimated using the same simulation parameters as the controlled orbit example. The asteroid in this case, though, is assumed to be non-rotating for simplicity. We first look at a sequence of images from time t to $t+1$. Identified landmarks in the first image are located in the second image, and used to calculate the optical flow and 3D velocity of the objects. Once accomplished, the first image is discarded, and a new image sequence from time $t+1$ to $t+2$ is used instead. Figure 15 illustrates the optical flow of an image when the spacecraft is moving to the image's left. The flow vectors on the body, particularly those on the top half, all tend to point in the direction of motion as expected. As was mentioned before, there are some areas at the asteroid edges that vary greatly with the flow of the surrounding areas. This is most likely because the image gradients at these areas are ill-conditioned for optical flow calculation. At any given time, optical flow vectors for up to 30 landmarks were computed from which a least square estimate of velocity could be calculated. Figure 16 shows the estimated velocity magnitudes along each axis using this method compared to the known spacecraft velocity. As can bee seen, the estimate reaches a stable value fairly quickly, but retains a bias of 1 m/s below what the true velocity is known to be. Although the exact reason for this is currently unknown, a possible explanation is improper tuning in the estimation scheme, and will be further investigated.



(a) First image.

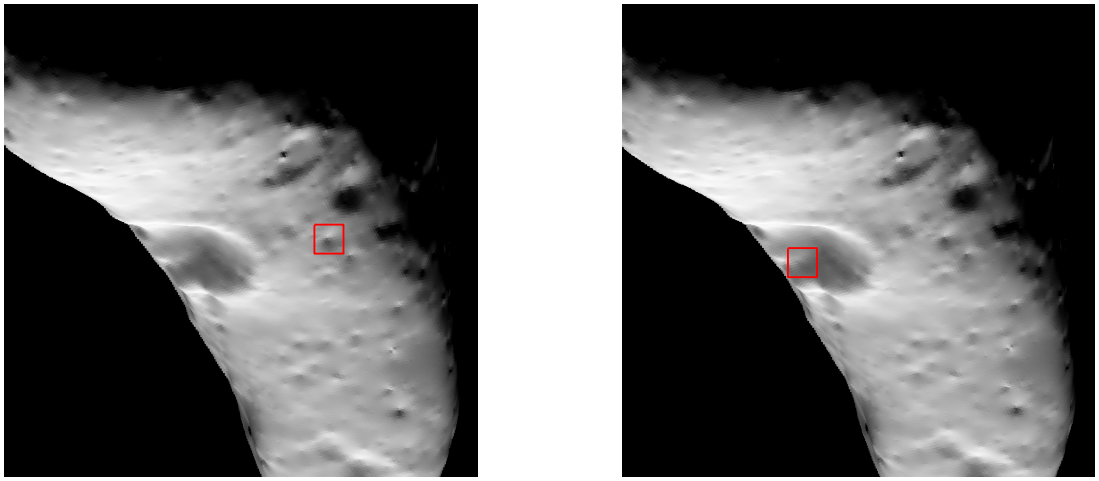
(b) Second image.

Figure 12: Landmark detection and tracking test using successive images. Images curtesy of NASA.



(a) t = 0 s

(b) t = 5 s



(c) t = 25 s

(d) t = 26 s

Figure 13: Crater tracking example using GPU simulated optical images of a rotating asteroid.

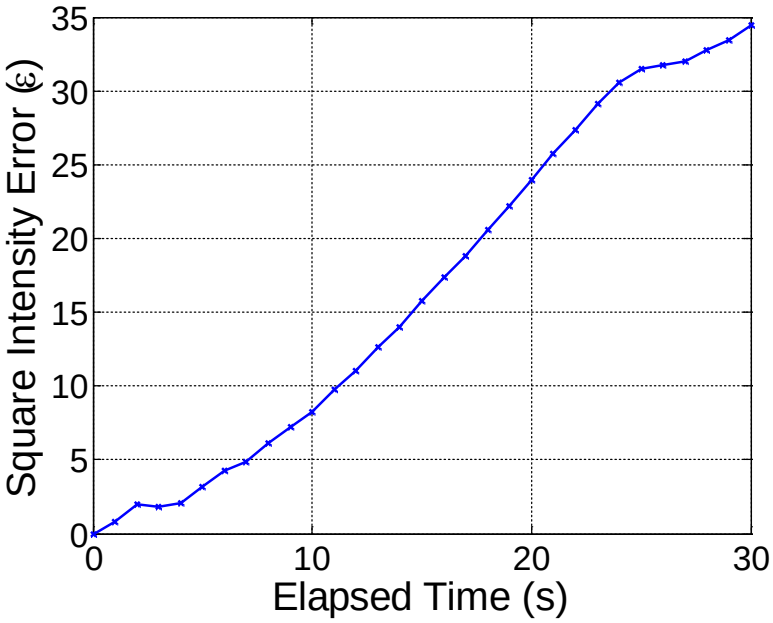


Figure 14: Shi-Tomasi-Kanade feature tracking intensity matching error over time.

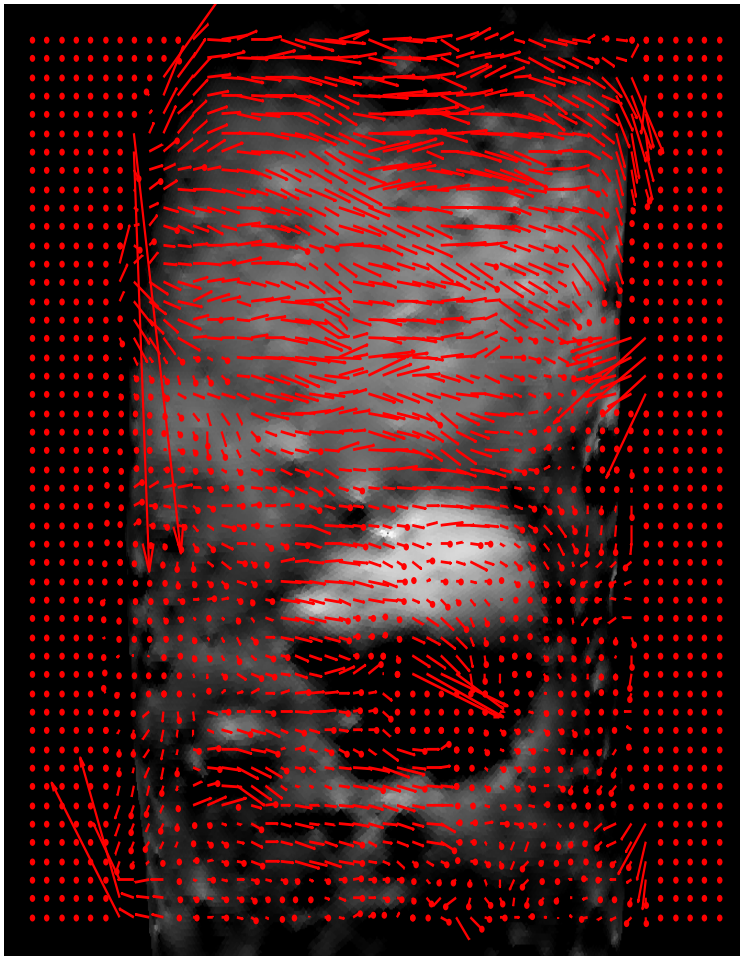


Figure 15: Optical flow vectors for a sample image (scaled for better visibility).

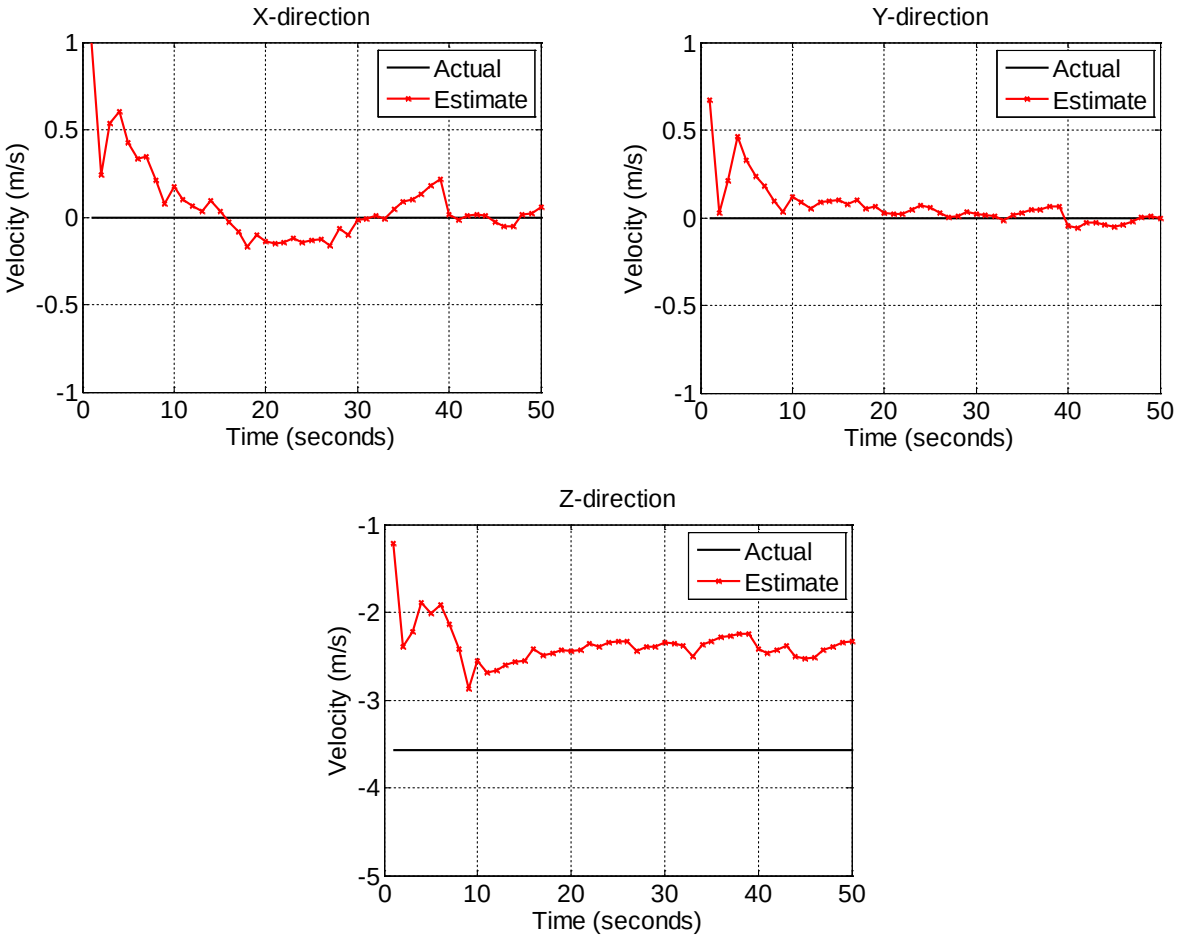


Figure 16: Optical flow estimation of 3D velocity.

5. Conclusion

In this paper, a strategy for controlling the orbital motion of spacecraft around asteroids was introduced and tested. The concept of disturbance-accommodating filtering was validated for a spacecraft operating in close-proximity to a target body with ΔV requirements under 0.15 m/s per week. This method of control works very well given the spacecraft's state vector. Therefore, methods for estimating a spacecraft's state vector around an asteroid using only optical data and LIDAR technologies were studied and tested. First, estimates of the asteroid body's shape using LIDAR measurements were shown to converge quickly to an accurate solution. While the mass can also be estimated in a similar manner, the rate of convergence is much slower, and other techniques might prove quicker and more effective. Then, the landmark detection and tracking algorithm was tested on actual and simulated images of an asteroid's surface. The algorithm was able to effectively track surface features, which were used to estimate 3D velocity based on 2D optical flow calculations. The current estimates show a constant bias in the solution of approximately 1 m/s. This could be due to mistuning of the estimation scheme, which will be further refined in future work. Once corrected, the estimates of a spacecraft's state vector shown here will be tested in the feedback control scheme to further verify the validity of this approach.

Acknowledgments

This research work has been supported by a NASA Innovative Advanced Concepts (NIAC) Phase 2 project. The authors would like to thank Dr. John (Jay) Falker, the NIAC Program Executive, for his support.

References

[1] Winkler, T., Hawkins, M., Lyzhoft, J., and Wie, B., Fuel-Efficient Feedback Control of Orbital Motion Around an Irregular-Shaped Asteroid, in: Proceedings of the 2012 AIAA/AAS Guidance, Navigation, and Control Conference, Minneapolis, MN. August 13-16, 2012, AIAA Paper Number 2012-5044.

[2] Scheeres, D. J., Orbital Mechanics About Small Bodies, Acta Astronautica27 (2012) 1-14.
[3] Scheeres, D. J., Ostro, S. J., Hudson, R. S., and Werner, R. A., Orbits Close to Asteroid 4769 Castalia, Icarus 121 (1996) 67-87.
[4] Scheeres, D. J., Williams, B. G., and Miller, J. K., Evaluation of the Dynamics Environment of an Asteroid: Applications to 433 Eros, Journal of Guidance, Control, and Dynamics 23 (2000) 466-475.
[5] Hu, W. and Scheeres, D. J., Periodic Orbits in Rotating Second Degree and Order Gravity Fields, Chinese Journal of of Astronomy and Astrophysics 8 (2008) 108-118.
[6] Scheeres, D. J., Miller, J. K., and Yeomans, D. K., The Orbital Dynamics Environment of 433 Eros: A Case Study for Future Asteroid Missions, InterPlanetary Network Progress Report 42-152, 2003.
[7] Kawaguchi, J. Hayabusa, Summary of Guidance, Navigation, and Control Achievement in its Proximity Phase, in: Proceedings of the 2006 AIAA/AAS Astrodynamics Specialist Conference, Keystone, CO. August 21-24, 2006. AIAA Paper Number 2006-6533.
[8] Broschart, S. B. and Scheeres, D. J., Control of Hovering Spacecraft Near Small Bodes: Application to Asteroid 35143 Itokawa, Journal of Guidance, Control, and Dynamics 27 (2005) 343-354.
[9] Kubitschek, D. et al., Deep Impact Autonomous Navigation: The Trials of Targeting the Unknown, in: Proceedings of the 29^{th} Annual AAS Guidance and Control Conference, Breckenridge, CO. February 4-8, 2006. AAS 06-081.
[10] Bhaskaran, S., Riedel, J. E., and Synnot, S. P. Autonomous Nucleus Tracking for Comet/Asteroid Encounters: The STARDUST Example, in : Proceedings of the 1997 AAS/AIAA Astrodynamics Specialist Conference, Sun Valley, ID. August 4-7, 1997 AAS 97-628.
[11] Bhaskaran, S. et al., Orbit Determination Performance Evaluation of the Deep Space 1 Autonomous Navigation System, in: Proceedings of the 1998 AAS/AIAA Spaceflight Mechanics Meeting, Monterrey, CA. February 9-11, 1998. AAS 98-193.
[12] Werner, R. A. and Scheeres, D. J., Exterior Gravitation of a Polyhedron Derived and Compared with Harmonic and Mascon Gravitation Representations of Asteroid 4769 Castalia, Celestial Mechanics and Dynamical Astronomy 65 (1997) 313-344.
[13] Thomas, P.C., et al, Eros: Shape, Topography and Slope Processes, Icarus 155 (2002) 18-37.
[14] Gouraud, H. Continuous Shading of Curved Surfaces, IEEE Transactions on Computers C-20 (1971) 623629.
[15] Nocedal, J. and Wright, S.J., Numerical Optimization, Springer Series in Operations Research, Springer Science + Business Media, 2^{nd} edition, 2006.
[16] Shuang, L., Hutao, C., and Pingyuan, C., Autonomous Optical Navigation for Landing on Asteroids, Aircraft Engineering and Aerospace Technology 77 (2005) 317-323.
[17] Misu, T., Hashimoto, T., and Ninomiya, K., Optical Guidance for Autonomous Landing of Spacecraft, Aerospace and Electronic Systems 35 (1999) 459-473.
[18] Benedetti, A. and Perona, P., Real-Time 2-D Feature Detection on a Reconfigurable Computer, in: Proceedings of the 1998 IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA. June 23-25, 1998.
[19] Barron, J. L, Fleet, D. J., and Beauchemin, S. S., Performance of Optical Flow Techniques, Internation Journalc of Computer Vision 12 (1994) 43-77.
[20] Horn, B. K. and Schunck, B. G., Determining Optical Flow, Artificial Intelligence 17 (1981) 185-203.
[21] Lucas, B. D. and Kanade, T., An Iterative Image Registration Technique with an Application to Stereo Vision, in: IJCAI'81 Proceedings of the 7th Internation Joint Conference on Artificial Intelligence, volume2, Morgan Kaufmann Publishers Inc., San Francisco, CA. 1981, pp. 674-679.
[22] Cielaszyk, D. and Wie, B., New Approach to Halo Orbit Determination and Control, Journal of Guidance, Control, and Dynamics 19 (1996) 266-273.
[23] Wie, B., Space Vehicle Dynamics and Contro, AIAA Education Series, American Institute of Aeronautics and Astronautics, 2^{nd} edition, 2008.
[24] NEAR-A-5-COLLECTED-MODELs-v1.0, URL: http://sbn.psi.edu/pds/resource/nearbrowse.html.