THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF AEROSPACE ENGINEERING

ATTITUDE DYNAMICS MODELING FOR DOCKING OPERATIONS WITH UNCOOPERATIVE OBJECTS

PATRICK WITTICK
SPRING 2015

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Aerospace Engineering
with honors in Aerospace Engineering

Reviewed and approved* by the following:

David B. Spencer
Professor of Aerospace Engineering
Thesis Supervisor

Robert G. Melton
Professor of Aerospace Engineering
Honors Adviser

George A. Lesieutre
Professor and Head of Aerospace Engineering

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

The capture of an uncooperative object by a man-made spacecraft presents a unique dynamics and control challenge. The research detailed here focuses on a method of mass property characterization that takes place after such a rendezvous has occurred. The theoretical and practical grounding for the problem is detailed, including some exploration of NASA's Asteroid Redirect Mission (ARM). A dynamics model for the spacecraft-asteroid system is then developed. The simulation algorithm used to complete the mass characterization is described. The results of the simulation run for three different asteroid masses follow. The method presented here was able to obtain the mass and mass center to within ten percent of true values for some special cases, while its general accuracy requires further investigation. Following an analysis of the results, several potential extensions of this research are discussed along with the concluding remarks.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Completing a piece of research never occurs in a vacuum, so I will take this opportunity to thank the several people that made this thesis possible.

First, I offer my sincere thanks to my thesis supervisor, Dr. David Spencer, for his invaluable help in defining the problem addressed herein and his feedback during the research work itself. Having such a knowledgeable and experienced source of input on this project has been indispensable.

I would also like to thank my honors advisor, Dr. Robert Melton for his input and academic guidance during my years as an aerospace engineering student. He also doubled as my instructor for my first astronautics and dynamics and control courses, and those classes were significant in my decision to pursue orbit and attitude mechanics for this thesis and in my future graduate studies.

I owe special thanks to my parents for providing an environment during my childhood in which I could develop my long-held interest in space science and engineering, and for their eternal support during my undergraduate studies.

Chapter 1

Introduction

The challenge of rendezvousing and docking with uncooperative objects has been of increased interest during the second decade of the twenty-first century in the field of spacecraft orbit and attitude control. This interest has several contributing factors, not the least of which is the proposed Asteroid Redirect Mission (ARM). The mission has been proposed by the President as a stepping stone toward developing technologies for a future Mars landing. It involves the rendezvous, capture, and return to cis-lunar orbit of a Near-Earth Asteroid (NEA) or a piece thereof for the purpose of manned exploration. The capture of an uncooperative object by a man-made spacecraft presents a unique dynamics and control challenge currently under investigation and development by NASA and others in the field. Other potential applications of uncooperative docking technology are satellite repair/refueling, asteroid mining, and autonomous distributed space systems.

This research addresses the problem of asteroid capture in particular. It considers a spacecraft which has just docked with an asteroid of unknown mass, moments of inertia, principal axis system, and center of mass location. While some of these quantities can be roughly estimated by visual observation, accurate determination of these mass properties is crucial in providing the ability of the capturing spacecraft to control the new composite spacecraft-asteroid body in both attitude and trajectory. The action of docking itself provides a unique interface through which the spacecraft can detect the new system's response to a given input. In this thesis a linear deterministic mass property characterization method is used to find the location of the new center of mass and total mass with a goal of determining these quantities to within 10% of their true values.

There is some history of control of uncooperative objects by human spacecraft. One such event occurred during the ill-fated Apollo 13 mission, when the descent engine and attitude control system of the Lunar Module (LM) were forced to assume responsibility for the control of the entire spacecraft, including the Command and Service Module (CSM). However, the mass properties of the CSM were well known to the engineers in Houston, meaning the challenge of Apollo 13 was less about the dynamics of the spacecraft and more about resource consumption and conservation. With ARM, new technologies and methods must be developed to develop an unmanned spacecraft to autonomously characterize the docked composite body and adjust its control system accordingly. Some studies have begun to address this need. A study completed by Bandyopadhyay, Chung, and Hadaegh[1] considered the optimal control problem of despinning a captured asteroid, but did so with known mass characteristics (within 10% margins, generally) and aligned principal axis systems of both the spacecraft and asteroid. Merrill, et al.[2] designed low-thrust interplanetary trajectories for returning part of an asteroid without considering the dynamics of the capture itself. Fourie, et al.[3] examined purely vision-based (no GPS) control with the SPHERES experiement aboard the International Space Station (ISS) for the purpose of noncooperative object inspection and circumnavigation. Gaias, D'Amico, and Ardaens[4] even designed and ran a test of an angles-only navigation and rendezvous technique on the PRISMA mission – although their approach made use of GPS data, which would not be available during an asteroid capture mission.

This project directly addresses the problem of mass property characterization for a spacecraft which has captured an uncooperative asteroid. The principal axis systems of the two objects are considered to be misaligned at an arbitrary relative orientation in order to simulate an inexact execution of docking maneuvers. A simulation of the body's behavior under a 5 second thrust is presented, while a method of recovering the system's mass and location of the mass center via simulated outputs is also discussed. For the purposes of this research, it is assumed that the spacecraft has already made a successful rendezvous with the asteroid, any residual forces and vibrations due to ill-matched angular velocities have subsided, and that the body is initially motionless and irrotational. Any influence of orbital

motion is also neglected. The linear deterministic mass property characterization method developed for this work is shown to accurately recover the total system mass and the location of the system center of mass to within about ten percent error in specific circumstances. The general applicability and functionality of this method is likely very narrow due to this fact. Future work is needed to expand this method and increase its robustness and accuracy.

Chapter 2

Background

Before embarking on a description of the system at hand, a summary of the general goals and methods of spacecraft attitude control and a review of the dynamic equations governing the problem at hand will prove useful.

Spacecraft Attitude Control

The ability to accurately control the orientation of a spacecraft is a crucial component to many space missions. Whether the spacecraft is conducting observations of a certain region of the sky, gazing intently at a certain location or locations on the body it orbits, or simply needs to communicate via a directional antenna, the spacecraft's ability to control its attitude is a mission-critical component of the system design. In almost no case would it be acceptable or desirable for a spacecraft to spin in an unknown or uncontrollable manner. For these reasons it is important to consider and investigate the attitude control aspect of any novel space mission architecture such as the Asteroid Redirect Mission.

Classical descriptions of attitude dynamics generally require knowledge of two qualities of the system to be controlled: the system's mass and how that mass is distributed through the spacecraft. The mass can be expressed in a single number, while the mass distribution quality must be broken into quantities such as the location of the center of mass, the principal axis system, and the principal moments of inertia. With most spacecraft, knowledge of these quantities is relatively simple to obtain via direct measurement before launch. The mass properties may then be built into the spacecraft's control system without much need for adaptability. In the context of the capture of an uncooperative object, the capturing

spacecraft must leave Earth only with knowledge of its own mass properties, and is forced to obtain the new mass properties of the docked system through some other means.

For this research project, the means of determining the system mass properties is considered to be through physical interaction between the spacecraft's thrusters and the docked system's response to them. The capturing spacecraft is assumed to have the very typical features of a thruster along its major axis and several accelerometers placed around the spacecraft bus. Because of this, the method discussed in the following chapters does not require any advanced spacecraft technology not currently in development, enhancing its applicability to the real world.

Governing Dynamic Equations

The chief equation governing the attitude dynamics of a spacecraft is the conservation of angular momentum, expressed as follows:

$$\frac{d\vec{H}}{dt} = \frac{d}{dt}\left(\mathbf{I}_p \vec{\omega}\right) = \vec{\tau}_{net} \tag{1}$$

Equation (1) expresses the time rate of change of the angular momentum, $\frac{d\vec{H}}{dt}$, in terms of the net moment on the system, $\vec{\tau}_{net}$. This equation holds true for the rigid body assumption; that is, that the position vector between any two points on the body is constant in a body-fixed reference frame. In order to avoid the flexibility and vibration considerations associated with a flexible body, the spacecraft-asteroid composite body is assumed to be rigid within the scope of this research. Incorporation of these factors into this problem is left to future work.

By substituting the product of the principal inertia tensor, $\mathbf{I}_p$, and the angular velocity vector, $\vec{\omega}$, for the angular momentum in Eq. (1), Euler's equations of rigid body motion can be derived. The results of this derivation are:

$$
\begin{aligned}
M_1 &= I_1\dot{\omega}_1 + \left(I_3 - I_2\right)\omega_2\omega_3 \\
M_2 &= I_2\dot{\omega}_2 + \left(I_1 - I_3\right)\omega_1\omega_3 \\
M_3 &= I_3\dot{\omega}_3 + \left(I_2 - I_1\right)\omega_1\omega_2
\end{aligned}
\tag{2}
$$

Here, the subscripts i = 1, 2, and 3 indicate the three principal directions in the body's principal axis system, the M_i are the three components of the net torque, the I_i are the three principal moments of inertia, and the ω_i are the three components of the angular velocity vector. For a known input torque profile, these equations may be integrated over time to predict the behavior of a rigid body. See Chapter 4 for more detail on how this integration process is included in the simulation algorithm.

Chapter 3

System Dynamics Model

The system model will now be developed, including a simplified spacecraft design and basic mass distribution model of the asteroid. The mass properties of the two objects are then combined to form the overall system model.

Spacecraft Design

For the purposes of this research, the spacecraft is assumed to follow on a basic level the conceptual design put forth by the COMPASS team at NASA Glenn Research Center (GRC), on advice from the Keck Institute for Space Studies (KISS)[5]. This design is shown below in Fig. 1.

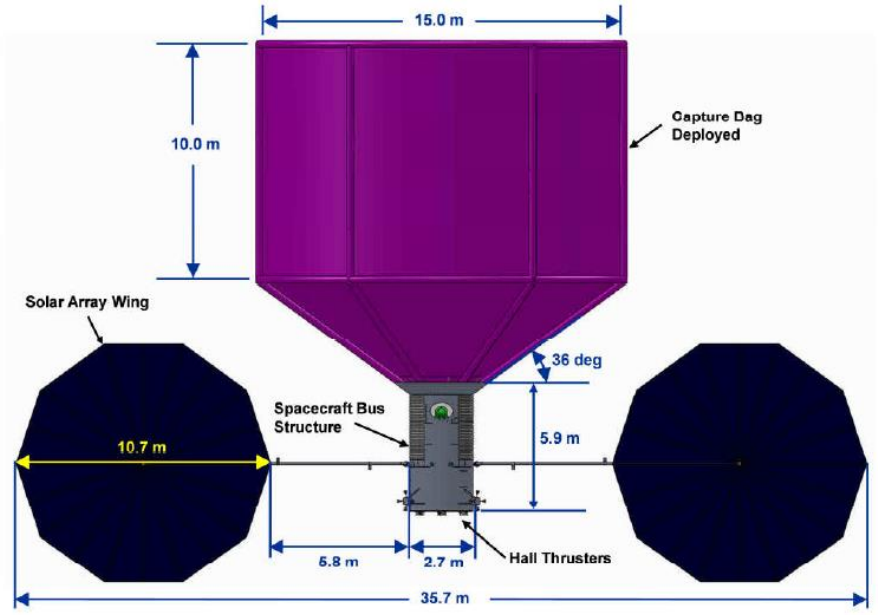


Fig. 1 Conceptual Design of Spacecraft with Deployed Capture Mechanism[5]

The one deviation from this design that is made is to assume that the spacecraft is completely symmetric about its major axis (i.e. the solar panels are neglected). This assumption is made for simplicity in the calculations only; in the final spacecraft design, the solar panels could easily be included, given knowledge of their exact sizes and masses. In effect, this assumption reduces the spacecraft to a basic cylinder. The spacecraft mass is assumed to be about 15 metric tons. Figure 2 shows the modeled spacecraft with dimensions shown as set forth in the KISS report.

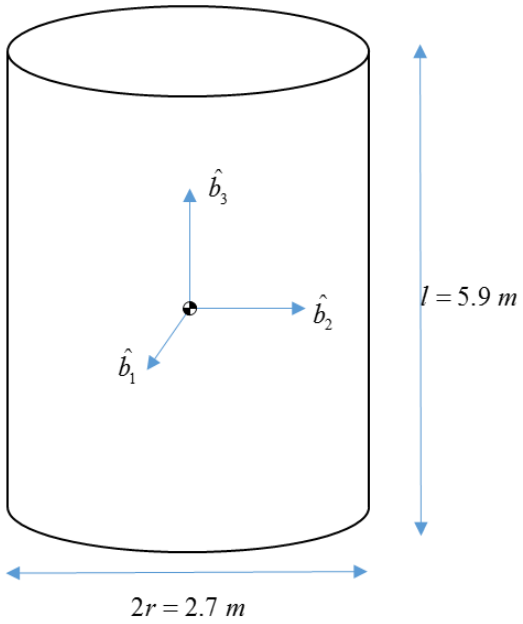


Fig. 2 Simplified Spacecraft Model

The $\hat{b}_n$ vectors shown in Fig. 2 define the principal axis system of the spacecraft, where $\hat{b}_1$ and $\hat{b}_2$ lie in a plane perpendicular to the central axis of the cylinder and which is equidistant from the cylinder's two ends, and $\hat{b}_3$ lies along the central axis. Given the mass of the spacecraft, M_s, and the dimensions and coordinate system described above, and assuming the mass distribution inside the spacecraft is uniform, the principal inertia tensor for the spacecraft is given as:

$$\mathbf{I}_s = \begin{bmatrix} I_{s_1} & 0 & 0 \\ 0 & I_{s_2} & 0 \\ 0 & 0 & I_{s_3} \end{bmatrix} = \begin{bmatrix} \frac{M_s l^2}{12} & 0 & 0 \\ 0 & \frac{M_s l^2}{12} & 0 \\ 0 & 0 & \frac{M_s r^2}{2} \end{bmatrix} \tag{3}$$

Asteroid Model

While some studies[5,6] have assumed a spherical asteroid model for similar asteroid capture scenarios, the mass distribution here is assumed to be similar to a dumbbell. This shape approximates the irregularity of an asteroid such as Itokawa, the asteroid explored by JAXA's Hayabusa probe. Because ARM aims to capture a small asteroid (~2-10 m in diameter), it is safe to assume that the asteroid will have an irregular shape due to gravity's minimal ability to shape celestial bodies on the meter scale. A diagram of the model is shown in Fig. 3.

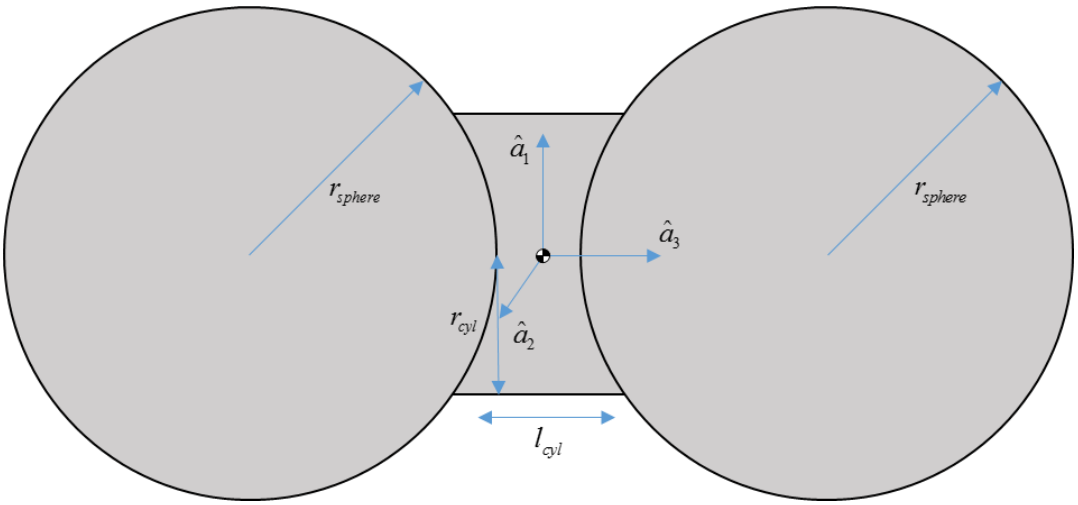


Fig. 3 Model of Asteroid Mass Distribution

The $\hat{a}_n$ vectors describe the principal axis system of the asteroid, while the ends of the dumbbell are modelled as spheres of radius r_{sphere}, and the middle of the dumbbell is modelled as a cylinder of

length l_{cyl} and radius r_{cyl}. Assuming that all three segments of the mass distribution model contain equal portions of the total mass, M_a, and a uniform density, the principal inertia tensor of the asteroid is given as:

$$\mathbf{I}_a = \begin{bmatrix} \frac{4}{15}M_a r_{sphere}^2 + \frac{M_a l_{cyl}^2}{36} + \frac{2}{3}M_a(r_{sphere} + \frac{l_{cyl}}{2})^2 & 0 & 0 \\ 0 & \frac{4}{15}M_a r_{sphere}^2 + \frac{M_a l_{cyl}^2}{36} + \frac{2}{3}M_a(r_{sphere} + \frac{l_{cyl}}{2})^2 & 0 \\ 0 & 0 & \frac{M_a r_{cyl}^2}{6} + \frac{4}{15}M_a r_{sphere}^2 \end{bmatrix} \quad (4)$$

For ease of notation the inertia tensor can be expressed in the following terms:

$$\mathbf{I}_a = \begin{bmatrix} I_{a_1} & 0 & 0 \\ 0 & I_{a_2} & 0 \\ 0 & 0 & I_{a_3} \end{bmatrix} \quad (5)$$

It is important to note that while the above tensor describes the assumed mass distribution of the asteroid, the physical extent of the asteroid is not constricted to the boundaries of the above model in this research.

Composite Body

The key difficulty in characterizing the captured asteroid results from the lack of exact knowledge by the spacecraft of its orientation relative to the principal axis system of the asteroid. In order to simulate the realistic situation that this lack of knowledge will be present, the vector between the center of mass of the spacecraft and the center of mass of the asteroid will be assumed not lie along one of the spacecraft's principal axes and that the two principal coordinate systems are oriented in different, arbitrary directions. This relative orientation and general physical setup of the problem considered are shown in Fig. 4.

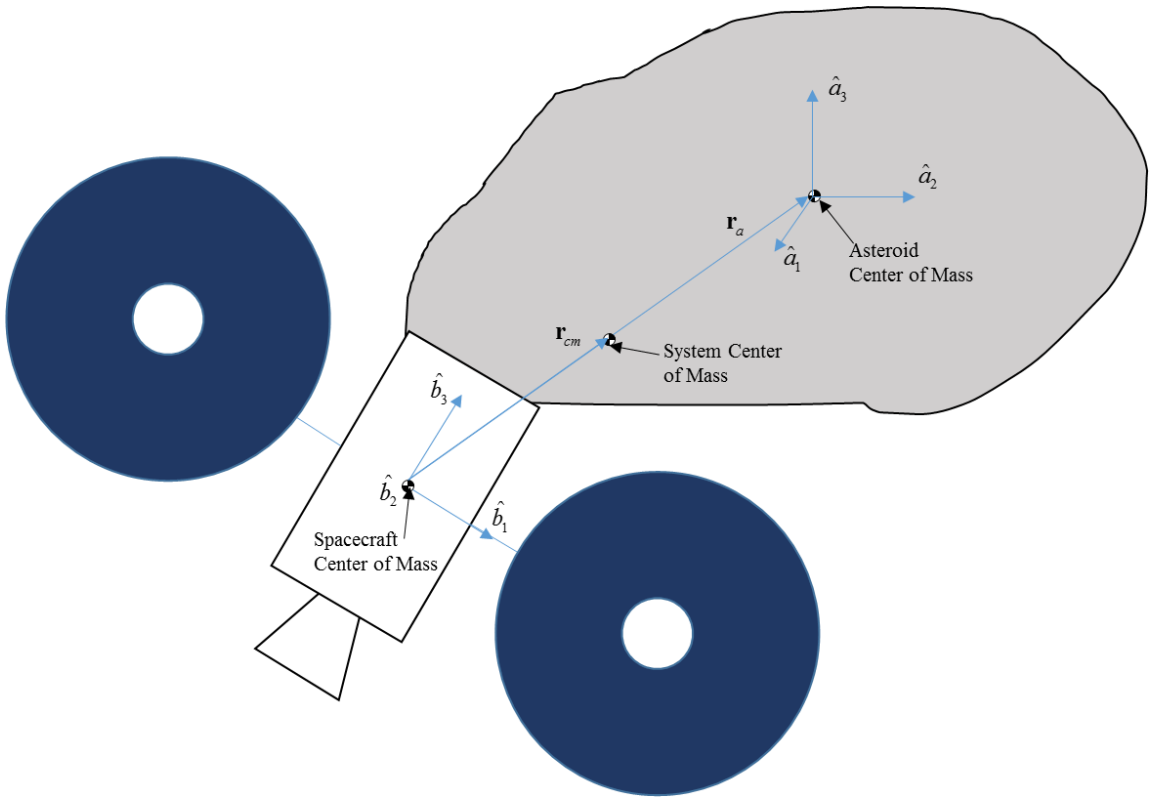


Fig. 4 Docked Spacecraft and Asteroid with Principal Axis Systems

Here the vector between the spacecraft center of mass and the asteroid center of mass is represented as $\mathbf{r}_a$ and the vector from the spacecraft center of mass to the system center of mass (post-docking) as $\mathbf{r}_{cm}$. These two vectors lie in the same direction and must be scalar multiples of one another because there are only two bodies in the system. The $\hat{a}$ and $\hat{b}$ frames are shown at arbitrary orientations, and $\hat{b}_2$ is out of the plane. The system center of mass may be located by employing

$$\mathbf{r}_{cm} = \frac{\sum m_i \mathbf{r}_i}{\sum m_i} \,, \tag{6}$$

which is valid for a system of rigid bodies whose mass centers are known. Expressing and evaluating this equation in the $\hat{b}$ frame, the following equation results.

$$\mathbf{r}_{cm} = \frac{M_a}{M_a + M_s}(r_{a_1}\hat{b}_1 + r_{a_2}\hat{b}_2 + r_{a_3}\hat{b}_3) \tag{7}$$

Next, the $\hat{b}$ frame is set at an arbitrary 3-1-3 Euler rotation from the $\hat{a}$ frame, with respective rotation angles of φ, θ, and ψ. The use of Euler angles in this attitude description is justified despite the singularity at θ = ±90 degrees because the level of accuracy in the rendezvous navigation system can reasonably be expected to not induce high degrees of misalignment. This allows the creation of the direction cosine matrix (DCM) which transforms from the $\hat{a}$ to the $\hat{b}$ frame, shown below:

$$\mathbf{C}_{b/a} = \begin{bmatrix} \cos\phi\cos\varphi - \sin\phi\cos\theta\sin\varphi & \sin\phi\cos\varphi + \cos\phi\cos\theta\sin\varphi & \sin\theta\sin\varphi \\ -\cos\phi\sin\varphi - \sin\phi\cos\theta\cos\varphi & -\sin\phi\sin\varphi + \cos\phi\cos\theta\cos\varphi & \sin\theta\cos\varphi \\ \sin\phi\sin\theta & -\cos\phi\sin\theta & \cos\theta \end{bmatrix} \tag{8}$$

Equation (8) is employed to rotate the asteroid's principal inertia tensor to align with the $\hat{b}$ frame using the following relation:

$$\mathbf{I}_a^b = \mathbf{C}_{b/a}\mathbf{I}_a\mathbf{C}_{b/a}^T \tag{9}$$

With the inertia tensors of the two bodies expressed about parallel axis systems, the parallel axis theorem may be used to find the total inertia tensor about the system center of mass. This new inertia tensor is expressed as $\mathbf{I}_{cm}$ with elements $I_{cm_{jk}}$, with j = 1, 2, 3, indexing rows, and k = 1, 2, 3, indexing columns. The expression below gives inertia tensor element $I_{cm_{jk}}$ from the elements of $\mathbf{I}_s$ and $\mathbf{I}_a^b$, using the parallel axis theorem.

$$I_{cm_{jk}} = I_{s_{jk}} + I_{a_{jk}}^b + r_{a_j} r_{a_k}\left[M_s(1-\chi)^2 + M_a\chi^2\right] \tag{10}$$

The constant χ is a simple substitution utilizing the following relation and is made purely for convenience.

$$\chi = 1 - \frac{M_a}{M_a + M_s} \tag{11}$$

A single inertia matrix for the composite body has now been developed. In the interest of calculation simplicity, the principal moments of inertia and principal axes for the composite body must now be found by solving the eigenvalue-eigenvector problem for the $\mathbf{I}_{cm}$ matrix.

As with any eigenvalue problem, the principal axis vectors and principal moments of inertia must satisfy the following relation:

$$\mathbf{I}_{cm}\hat{p}_i = I_{p_i}\hat{p}_i \tag{12}$$

We denote the principal axis vectors as $\hat{p}_i$ and the associated principal moments of inertia I_{p_i}. To solve this problem for the I_{p_i}, the following equation must be met:

$$\det\begin{bmatrix} I_{p_i}-I_{cm_{11}} & I_{cm_{12}} & I_{cm_{13}} \\ I_{cm_{12}} & I_{p_i}-I_{cm_{22}} & I_{cm_{23}} \\ I_{cm_{13}} & I_{cm_{23}} & I_{p_i}-I_{cm_{33}} \end{bmatrix} = 0 \tag{13}$$

Expanding this determinant results in a cubic polynomial of the form $ax^3+bx^2+cx+d' = 0$. The constants of this polynomial have the values given below:

$$\begin{aligned} a &= 1 \\ b &= -\left(I_{cm_{11}}+I_{cm_{22}}+I_{cm_{33}}\right) \\ c &= I_{cm_{11}}I_{cm_{22}}+I_{cm_{11}}I_{cm_{33}}+I_{cm_{22}}I_{cm_{33}}-\left(I_{cm_{12}}^2+I_{cm_{13}}^2+I_{cm_{23}}^2\right) \\ d' &= I_{cm_{11}}I_{cm_{23}}^2+I_{cm_{22}}I_{cm_{13}}^2+I_{cm_{33}}I_{cm_{12}}^2-I_{cm_{11}}I_{cm_{22}}I_{cm_{33}}+2\left(I_{cm_{12}}I_{cm_{13}}I_{cm_{23}}\right) \end{aligned} \tag{14}$$

The polynomial can be solved with the following equations used to find cubic roots of polynomials:

$$\begin{aligned} I_{p_i} &= \frac{-1}{3a}\left(b+u_iC+\frac{\nabla_0}{u_iC}\right) \\ \nabla_0 &= b^2-3ac \\ \nabla_1 &= 2b^3-9abc+27a^2d' \\ C &= \sqrt[3]{\frac{\nabla_1+\sqrt{\nabla_1^2-4\nabla_0^3}}{2}} \\ u_1 &= 1, \quad u_2 = \frac{-1+i\sqrt{3}}{2}, \quad u_3 = \frac{-1-i\sqrt{3}}{2} \end{aligned} \tag{15}$$

The principal axes of the system can then be solved using:

$$\begin{bmatrix} I_{p_i} - I_{cm_{11}} & I_{cm_{12}} & I_{cm_{13}} \\ I_{cm_{12}} & I_{p_i} - I_{cm_{22}} & I_{cm_{23}} \\ I_{cm_{13}} & I_{cm_{23}} & I_{p_i} - I_{cm_{33}} \end{bmatrix} \begin{bmatrix} \hat{p}_{i1} \\ \hat{p}_{i2} \\ \hat{p}_{i3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{16}$$

With the principal moments of inertia and principal axis system for the docked system defined, a full description of the properties of the system is now in place. To complete the loop and facilitate the simulation, the DCM between the $\hat{p}$ frame and the $\hat{b}$ frame is presented.

$$\mathbf{C}_{p/b} = \begin{bmatrix} \hat{b}_1 \cdot \hat{p}_1 & \hat{b}_1 \cdot \hat{p}_2 & \hat{b}_1 \cdot \hat{p}_3 \\ \hat{b}_2 \cdot \hat{p}_1 & \hat{b}_2 \cdot \hat{p}_2 & \hat{b}_2 \cdot \hat{p}_3 \\ \hat{b}_3 \cdot \hat{p}_1 & \hat{b}_3 \cdot \hat{p}_2 & \hat{b}_3 \cdot \hat{p}_3 \end{bmatrix} \tag{17}$$

The DCM from the $\hat{b}$ frame to the $\hat{p}$ frame is simply the transpose of the DCM in Eq. (17) due to the orthogonality of rotation matrices.

$$\mathbf{C}_{b/p} = \mathbf{C}_{p/b}^T \tag{18}$$

Chapter 4

Simulation Algorithm

The mass characteristics, coordinate systems, relative positions, and composite properties of a spacecraft and uncooperative asteroid in a docking scenario have now been presented in detail. All of this knowledge is now integrated into a simulation designed to demonstrate the ability of the spacecraft to recover the mass properties of the system through known inputs. In order to summarize the information presented thus far, Table 1 shows various properties of the two bodies as they are represented in the simulation (some data adapted from Brophy, et al.[5]).

Table 1 Various Initial Properties of the Spacecraft and Asteroid

Property	Spacecraft	Asteroid
Mass	15000 kg	10000 kg
Dimensions	5.9 meters (length) x 2.7 meters (diameter)	2-10 m
Shape	Cylindrical	Dumbbell mass distribution; irregular physical shape
Initial Rotation Speed	0 rad/s	0 rad/s

In order to maintain the integrity of the mass property recovery algorithm, the model developed above must be isolated from the rest of the simulation such that no mass properties of the asteroid are accessible to calculations that would be done by the spacecraft in a real-world scenario. To achieve this goal, the above system properties are isolated in a MATLAB function that takes thrust inputs with spacecraft mass properties and yields outputs of angular velocity and acceleration vectors at various points on the spacecraft. Those outputs could feasibly be measured by instruments onboard the spacecraft, and their values within this simulation are generated via the system properties contained within the function.

In reality, the angular velocity measurement could be made by one or more star trackers, while the acceleration vectors would be generated by six sets of three accelerometers. Each set would be oriented to match the principal axes of the spacecraft; therefore, they would take their measurements in the $\hat{b}$ frame. The layout of the accelerometers is approximated in Fig. 5 by the six red dots.

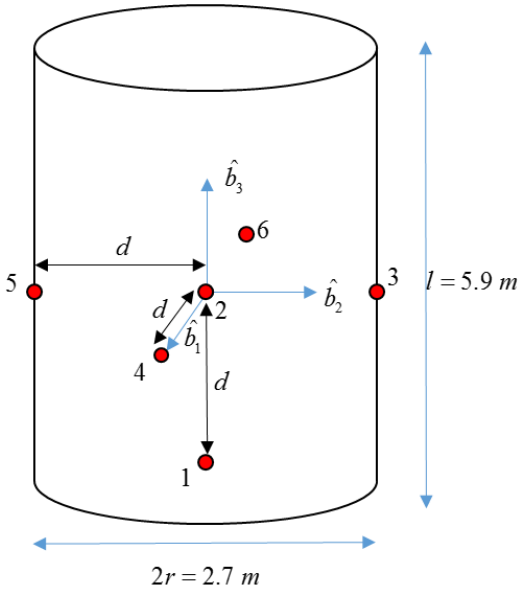


Fig. 5 Locations of Accelerometer Sets (Red Dots) on Spacecraft

The accelerometers are equidistant from the center of mass, separated by distance d, which is equal to the spacecraft's radius of 1.35 m. Accelerometers 3 and 5 lie along the positive and negative $\hat{b}_2$ directions, respectively, while accelerometers 4 and 6 lie along the positive and negative $\hat{b}_1$ axes, respectively. Accelerometer 2 lies at the center of mass, while accelerometer 1 lies along the negative $\hat{b}_3$ axis. This organization was chosen because it allows the simulated sensors to read data in three mutually orthogonal planes, ensuring that the measurements made can be used to calculate the full set of system mass properties.

For the purposes of this simulation, material flexibility is ignored and assume that the docked body behaves as a rigid body. That is, it follows the Newton-Euler Laws of Motion. Primarily, this scenario is governed by Newton's Second Law:

$$\begin{aligned} \sum \vec{F} &= m\vec{a} \\ \sum \vec{M} &= \mathbf{I}\vec{\alpha} \end{aligned} \tag{19}$$

and the expression for angular momentum:

$$\vec{H} = \mathbf{I}\vec{\omega} \tag{20}$$

All thrusts in this simulation are considered to last 5 seconds, and the integration of the equations of motion for the composite rigid body was completed for this entire period of constant thrust.

Given the above sensor locations, the algorithm for the simulation is as follows:

1. A 5 second thrust of known magnitude along the $\hat{b}_3$ axis is passed to the function containing the system dynamics.
2. The MATLAB function calculates the system's behavior in the $\hat{p}$ frame by integrating Eqs. (2).
3. The function outputs what the 6 accelerometer sets would read during the 5 second thrust based on system behavior.
4. The function outputs the angular velocity of the spacecraft in the $\hat{b}$ frame over the duration of the 5 second thrust.
5. The angular acceleration of the spacecraft during the thrust is calculated via differences in acceleration vectors measured simultaneously.
6. Taking acceleration data measured at the same location at two instances in time, the linear system of equations resulting from rigid body kinematics is solved for the position vector of the system center of mass and the acceleration of the system center of mass in the $\hat{b}$ frame.

7. The acceleration of the center of mass is employed in Newton's Second Law to find the total mass of the system.

A graphical representation of the simulation algorithm is displayed below in Fig. 6.

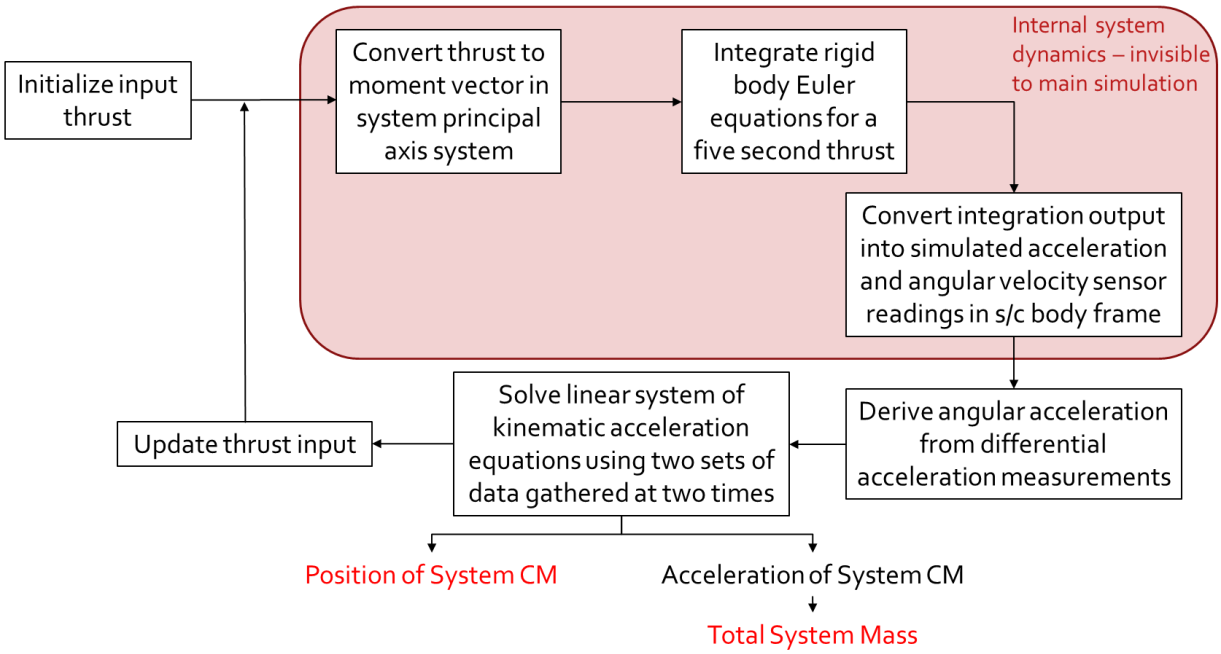


Fig. 6 Overview of the Simulation Algorithm

From this point forward, focus is placed on the conversion of the spacecraft data into system mass properties. A key equation in this process is the kinematic expression for acceleration of a three dimensional rotating rigid body.

$$\mathbf{a}_A = \mathbf{a}_B + \vec{\alpha} \times \mathbf{r}_{A/B} + \vec{\omega} \times (\vec{\omega} \times \mathbf{r}_{A/B}) \tag{21}$$

This equation can be modified to more specifically fit the docking scenario, and we will express it in the $\hat{b}$ frame.

$$\mathbf{a}_Q = \mathbf{a}_{cm} + \vec{\alpha} \times \mathbf{r}_{Q/cm} + \vec{\omega} \times (\vec{\omega} \times \mathbf{r}_{Q/cm}) \tag{22}$$

The point Q can be anywhere on the rotating body. The $\mathbf{a}_Q$ for this system come directly from the accelerometer readings for the six locations shown in Fig. 5, and the $\vec{\omega}$ vectors also come from the simulation output. The angular acceleration vector, $\vec{\alpha}$, can be solved for by subtracting Eq. (22) from

itself for two different measurement points taking simultaneous measurements. Choosing points 1 and 2 for example:

$$\mathbf{a}_1 - \mathbf{a}_2 = \vec{\alpha} \times (-d\ \hat{b}_3) + \vec{\omega} \times [\vec{\omega} \times (-d\ \hat{b}_3)] \tag{23}$$

All quantities except $\vec{\alpha}$ are known in Eq. (23), enabling the recovery of two out of the three terms of $\vec{\alpha}$. By subtracting two other acceleration vectors in a different plane, the third term can be found. The three terms are as follows:

$$\begin{aligned} \alpha_1 &= \omega_2 \omega_3 - \frac{(\mathbf{a}_1 - \mathbf{a}_2)_2}{d} \\ \alpha_2 &= \frac{(\mathbf{a}_1 - \mathbf{a}_2)_1}{d} - \omega_1 \omega_3 \\ \alpha_3 &= \omega_1 \omega_2 - \frac{(\mathbf{a}_2 - \mathbf{a}_3)_1}{d} \\ \vec{\alpha} &= \alpha_1\ \hat{b}_1 + \alpha_2\ \hat{b}_2 + \alpha_3\ \hat{b}_3 \end{aligned} \tag{24}$$

With $\vec{\alpha}$ found, there remain six unknown quantities inside $\mathbf{a}_{cm}$ and $\mathbf{r}_{i/cm}$, which will be the final pieces in determining the mass properties of the docked system. By taking acceleration measurements from one virtual sensor (the central one, in this case) and an angular velocity measurement at two different times, the following linear system derived from Eq. (22) may be solved for the components of $\mathbf{a}_{cm}$ and $\mathbf{r}_{i/cm}$:

$$\begin{bmatrix} 1 & 0 & 0 & -\left(\omega_2(t_1)^2 + \omega_3(t_1)^2\right) & \omega_1(t_1)\omega_2(t_1) - \alpha_3(t_1) & \omega_1(t_1)\omega_3(t_1) - \alpha_2(t_1) \\ 0 & 1 & 0 & \omega_1(t_1)\omega_2(t_1) + \alpha_3(t_1) & -\left(\omega_1(t_1)^2 + \omega_3(t_1)^2\right) & \omega_2(t_1)\omega_3(t_1) - \alpha_1(t_1) \\ 0 & 0 & 1 & \omega_1(t_1)\omega_3(t_1) - \alpha_2(t_1) & \omega_2(t_1)\omega_3(t_1) + \alpha_1(t_1) & -\left(\omega_1(t_1)^2 + \omega_3(t_1)^2\right) \\ 1 & 0 & 0 & -\left(\omega_2(t_2)^2 + \omega_3(t_2)^2\right) & \omega_1(t_2)\omega_2(t_2) - \alpha_3(t_2) & \omega_1(t_2)\omega_3(t_2) - \alpha_2(t_2) \\ 0 & 1 & 0 & \omega_1(t_2)\omega_2(t_2) + \alpha_3(t_2) & -\left(\omega_1(t_2)^2 + \omega_3(t_2)^2\right) & \omega_2(t_2)\omega_3(t_2) - \alpha_1(t_2) \\ 0 & 0 & 1 & \omega_1(t_2)\omega_3(t_2) - \alpha_2(t_2) & \omega_2(t_2)\omega_3(t_2) + \alpha_1(t_2) & -\left(\omega_1(t_2)^2 + \omega_3(t_2)^2\right) \end{bmatrix} \begin{bmatrix} a_{cm_1} \\ a_{cm_2} \\ a_{cm_3} \\ r_{cm_1} \\ r_{cm_2} \\ r_{cm_3} \end{bmatrix} = \begin{bmatrix} a_{2_1}(t_1) \\ a_{2_2}(t_1) \\ a_{2_3}(t_1) \\ a_{2_1}(t_2) \\ a_{2_2}(t_2) \\ a_{2_3}(t_2) \end{bmatrix} \tag{25}$$

The $\mathbf{r}_{cm}$ vector found by solving Eq. (25) fulfills one of the objectives of this project; the location of the center of mass of the composite body is now known. To achieve the second goal of obtaining the overall system mass, Newton's second law may be applied as:

$$M_{total} = \frac{F_{thrust}}{a_{cm_z}} \tag{26}$$

Because the five second engine burn is directed entirely in the $\hat{b}_3$ direction, only that component of Newton's second law is shown in Eq. (26). With this final step in place, the simulation achieves its objective of recovering both total system mass and the location of the center of mass.

Chapter 5

Simulation Results and Analysis

In order to test the method's sensitivity to thrust magnitude variations, the above simulation is run for thrusts ranging from 100 N to 110000 N, with intervals of 100 N. This is done for three different asteroid masses, called the low, medium, and high density cases here. The medium density case uses the asteroid mass shown in Table 1, while the low density case uses an asteroid mass of 5000 kg, and the high density case is run for an asteroid with a mass of 15000 kg. For all three cases, the position vector from the center of mass of the spacecraft to the center of mass of the system is fixed as:

$$\mathbf{r}_{cm} = 0.6\hat{b}_1 + 0.4\hat{b}_2 + 2.8\hat{b}_3 \tag{27}$$

Four plots are presented for each case: calculated mass versus thrust, calculated mass error versus thrust, calculated position vector error versus thrust, and calculated distance versus thrust. This set of graphs allows for an evaluation of the efficacy of this method in reliably determining the quantities being pursued, namely, the total system mass, M_{total}, and the position vector to the system center of mass, $\mathbf{r}_{cm}$. The calculated system mass averaged over the full range of thrust also proved to be an important quantity, and is presented and discussed alongside the plots just mentioned.

Low Density Case

For this case, the theoretical value of the total system mass is 20000 kg. As seen in Fig. 7, the calculated value for the system mass depends heavily on the input thrust. There are two locations on the plot which exhibit asymptotic behavior, and even those thrust values which are distant from those singularities do not yield very accurate measurements of the system's mass. As Fig. 8 shows, there is a

single thrust value near 66300 N which gives small errors (<4%) in the mass calculation, with nearly all other thrust magnitudes yielding errors much larger than 5%.

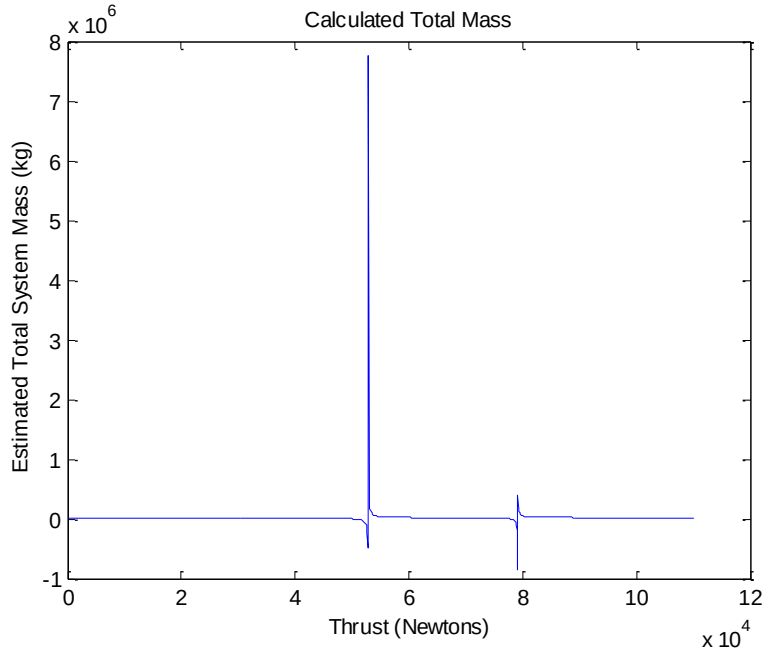


Fig. 7 Calculated Total Mass for Full Range of Input Thrusts (Low Density)

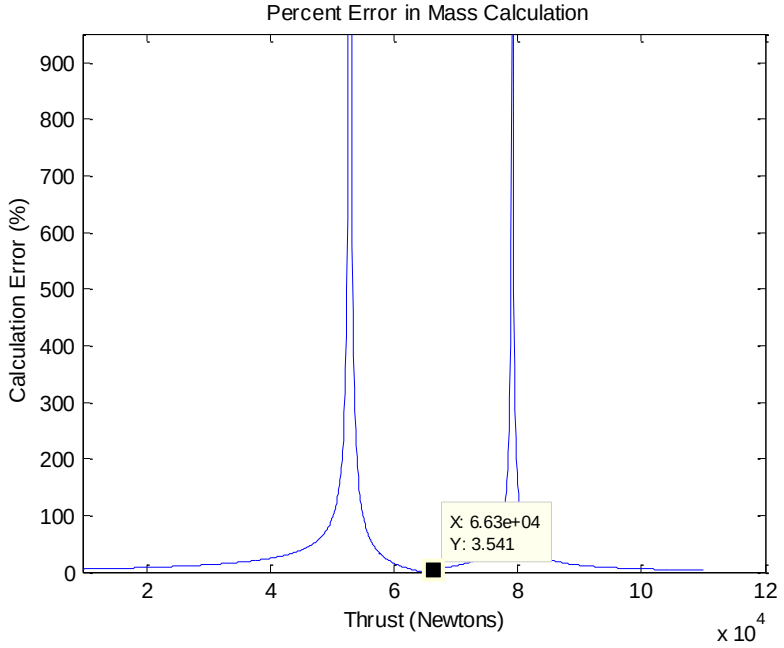


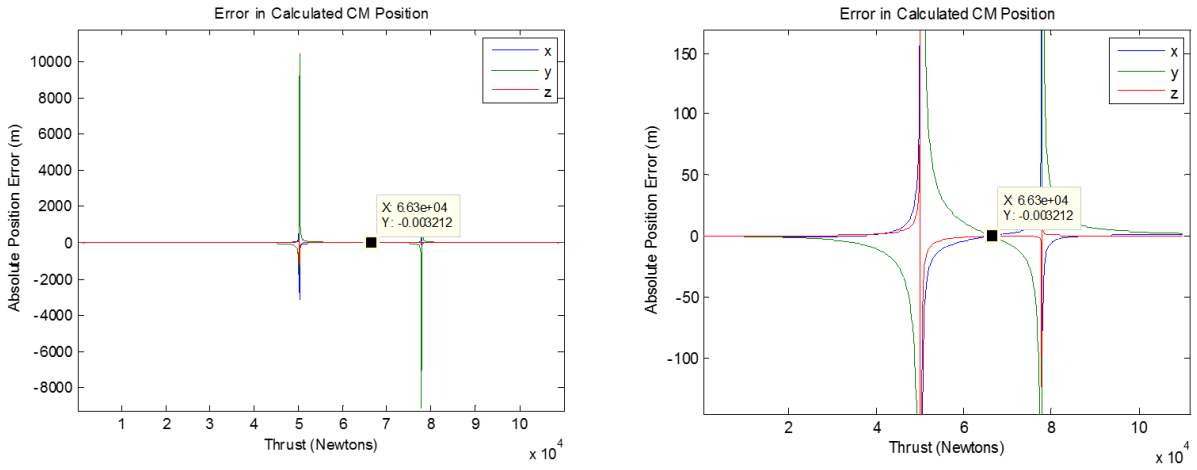
Fig. 8 Mass Calculation Error for Full Range of Input Thrusts (Low Density)

Table 2 Comparison of Calculated vs. True Mass (Low Density)

True Total Mass	20000 kg
Thrust-Averaged Total Mass	25370 kg
% Error	26.9%

When averaged over thrust, calculated system mass is higher by almost 27% than the true mass value for this case (Table 2). This value is almost certainly skewed upward due to the singularities, but nonetheless demonstrates the limited ability of this algorithm to accurately determine the total system mass.

When the error for each component of the calculated center of mass position vector is plotted against input thrust in Fig. 9(a) and (b), two singularities once again appear at locations which are very close to the singularities in Figs. 7 and 8. Upon closer examination, it is revealed that there is a location on this plot at the same thrust value that gives the smallest mass errors (66300 N) that also results in the smallest position errors for all three components. This point is also evident on the CM distance versus thrust plot in Fig. 10.



(a) **(b)**

Fig. 9 (a) Wide view of calculation error for the three components of the CM position vector; (b) Zoomed view of low error point between singularities (all Low Density)

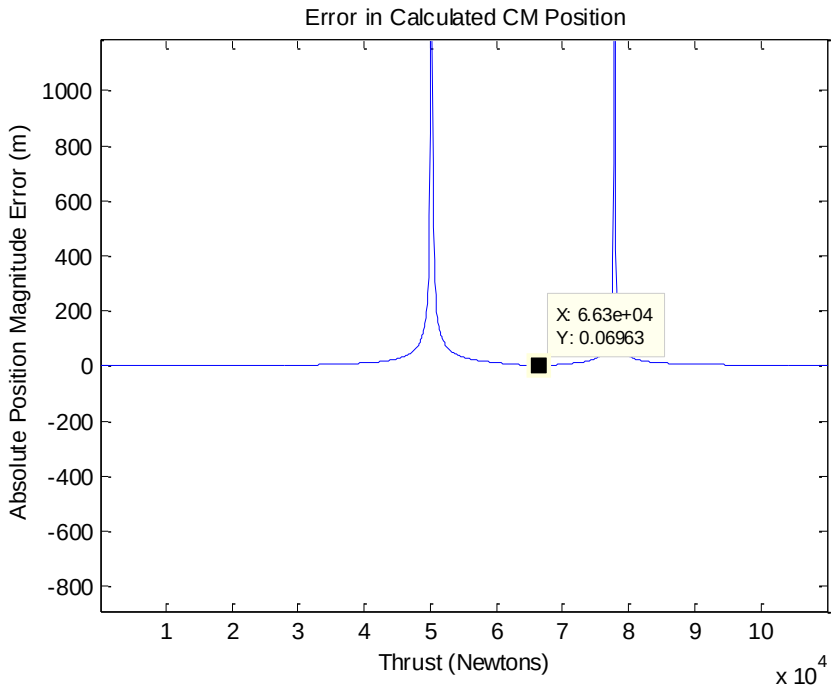


Fig. 10 Calculated CM Distance Error (Low Density)

Medium Density Case

The medium density case mirrors the proposed ARM most closely; an asteroid mass of 10000 kg is assumed, yielding an expected total mass of 25000 kg for the docked system. The behavior of the simulation differs greatly for this case when compared against the low density case. The most evident difference is the oscillatory behavior of the calculated mass as thrust increases (Fig. 11). The oscillations do vary around a mean value presented in Table 3, but they are not very large and tend to stay on the same order of magnitude as the true total system mass, unlike the calculation performed for the low density case. This yields a maximum mass calculation error of 28.1% for the entire input thrust range and several regions where mass calculation error falls below the stated goal of 10%, as seen in Fig. 12.

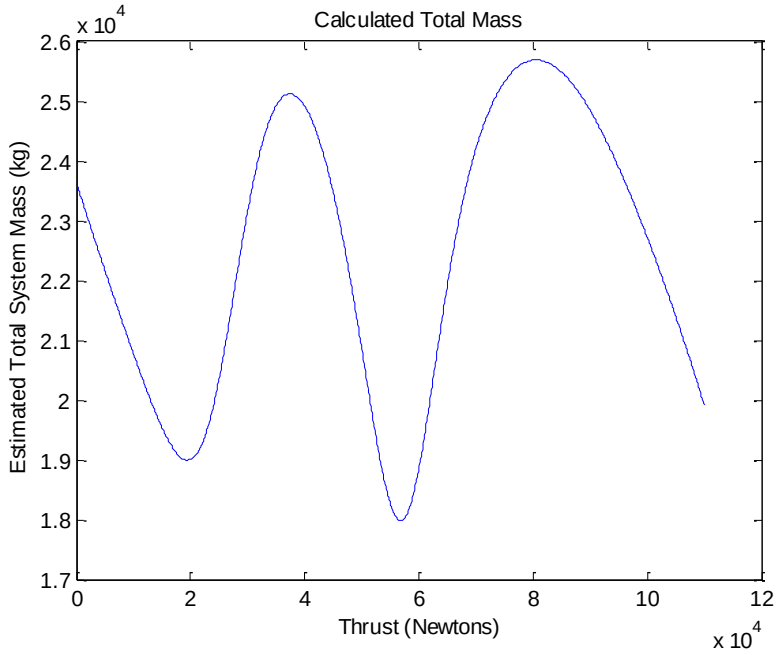


Fig. 11 Calculated Total Mass for Full Range of Input Thrusts (Medium Density)

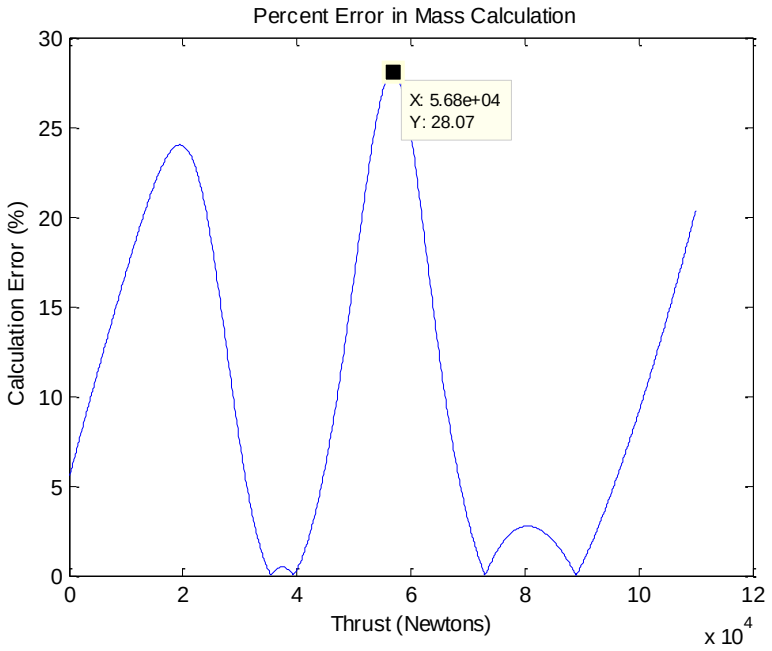


Fig. 12 Mass Calculation Error for Full Range of Input Thrusts (Medium Density)

Table 3 Comparison of Calculated vs. True Mass (Medium Density)

True Total Mass	25000 kg
Thrust-Averaged Total Mass	22433 kg
% Error	10.3%

The thrust-averaged calculated total mass differs by just over 10% from the true value. This error comes very close to the stated goal of 10%. None of these plots contain singularities reminiscent of the low density case. This is likely a cause of the overall lower errors achieved with this asteroid mass.

Both oscillatory behavior and the higher accuracy achieved in the determination of the system mass is also evident in the calculation of the center of mass position vector. As shown in both Figs. 13 and 14, the three components of the CM position vector reach points of minimum error at nearly the same three thrust values over the range of thrusts tested. Those thrust values are, in ascending order, 1500 N, 45100 N, and 98500 N.

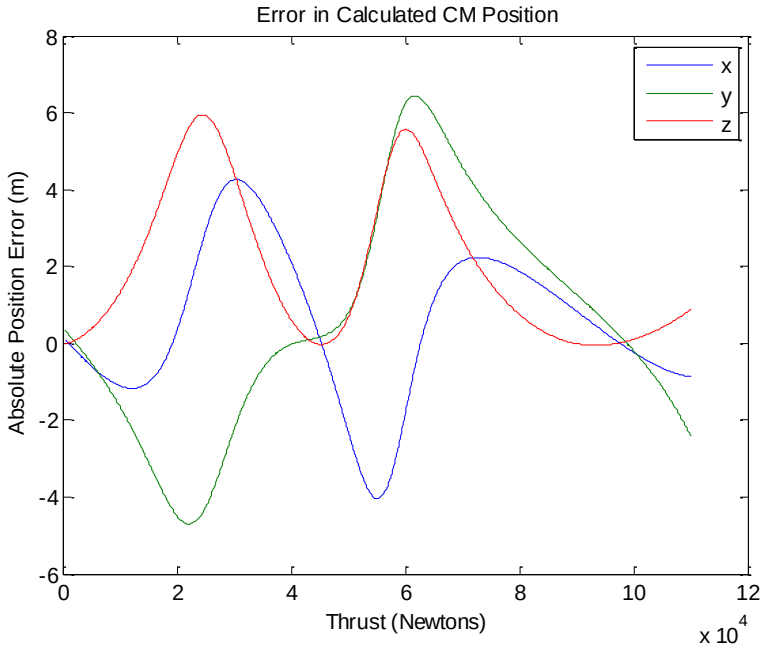


Fig. 13 Calculation error for the three components of the CM position vector (Medium Density)

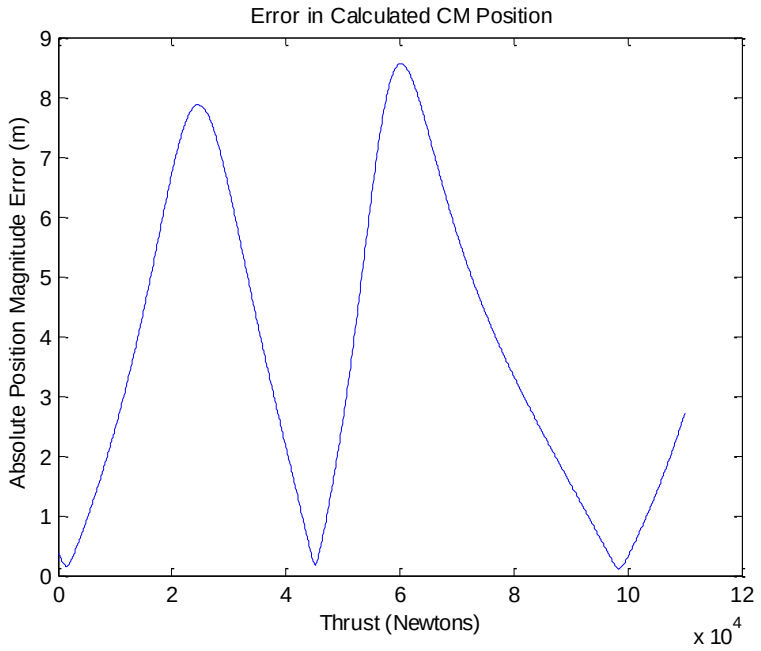


Fig. 14 Calculated CM Distance Error (Medium Density)

High Density Case

For this case, the asteroid was assumed to have an even higher mass of 15000 kg, giving an actual total mass of 30000 kg. The simulation behaved similarly for this case when compared with the medium density case, but still differed greatly from the low density case. Oscillatory behavior is still present, but oscillations in the calculated mass grow more quickly than for the medium density case (Fig. 15). The maximum error for this case reached nearly 56%, significantly higher than the medium density maximum error. Once again there are several regions in which error in the mass calculation does not exceed 10%, but these regions are smaller than for the medium density case (Fig. 16).

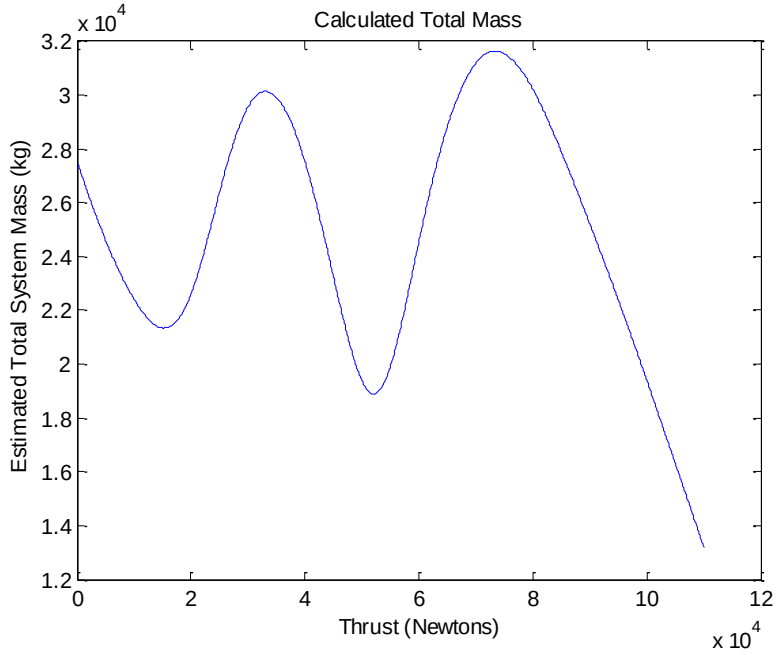


Fig. 15 Calculated Total Mass for Full Range of Input Thrusts (High Density)

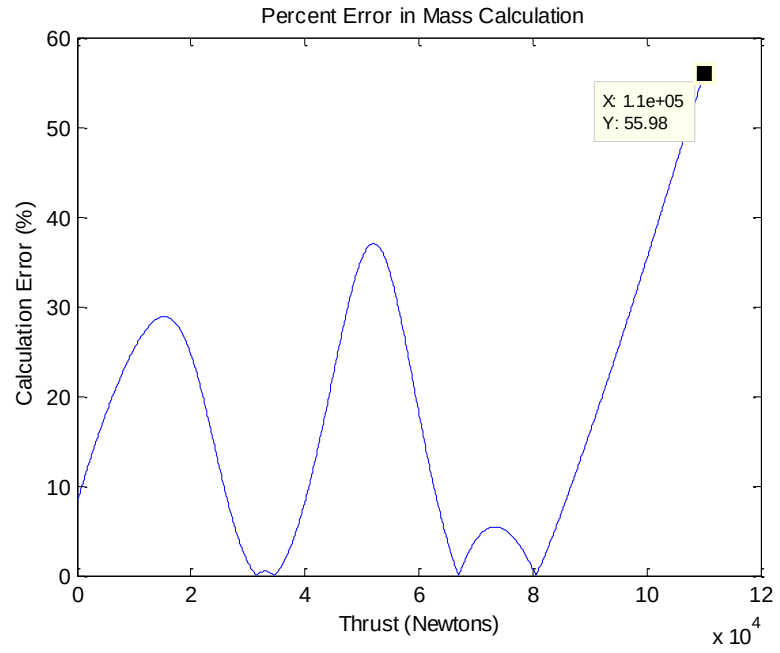


Fig. 16 Mass Calculation Error for Full Range of Input Thrusts (High Density)

When averaged over input thrust, the calculated mass was about 24760 kg, which gives a 17.5% error when compared with the actual mass of 30000 kg (Table 4).

Table 4 Comparison of Calculated vs. True Mass (High Density)

True Total Mass	30000 kg
Thrust-Averaged Total Mass	24760 kg
% Error	17.5%

The error in the mass for this case exceeds 10%, but not so severely as in the low density case. However, the increased error appears despite the absence of singularities, indicating that the simulation becomes less effective as the real total mass of the system increases.

Calculation of the CM position vector resulted in essentially the same kind of behavior observed for the medium density case: large oscillations and common points of minimum calculation error. One standout difference was that the high density case had larger maximum errors in calculating the CM position vector than the medium density case did. Figures 17 and 18 show the oscillatory behavior along with the three low-error thrust points.

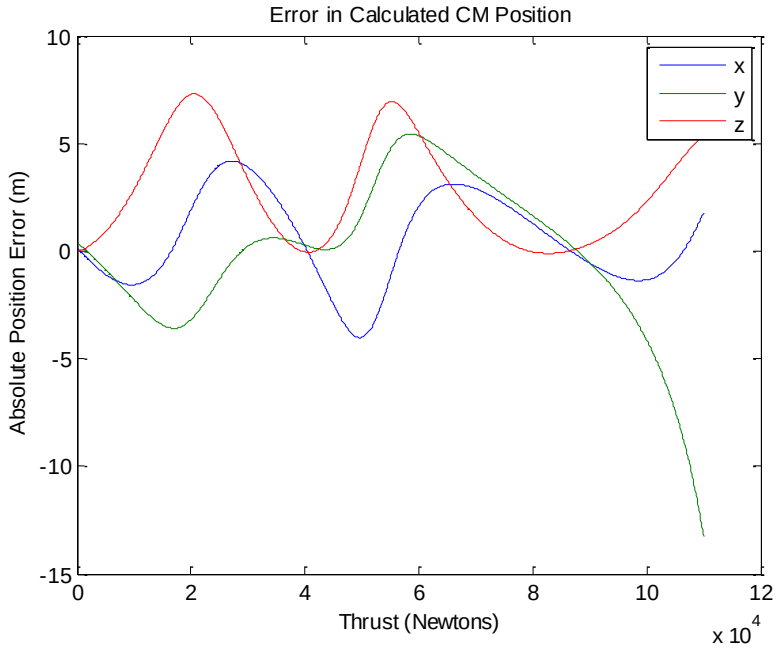


Fig. 17 Calculation error for the three components of the CM position vector (High Density)

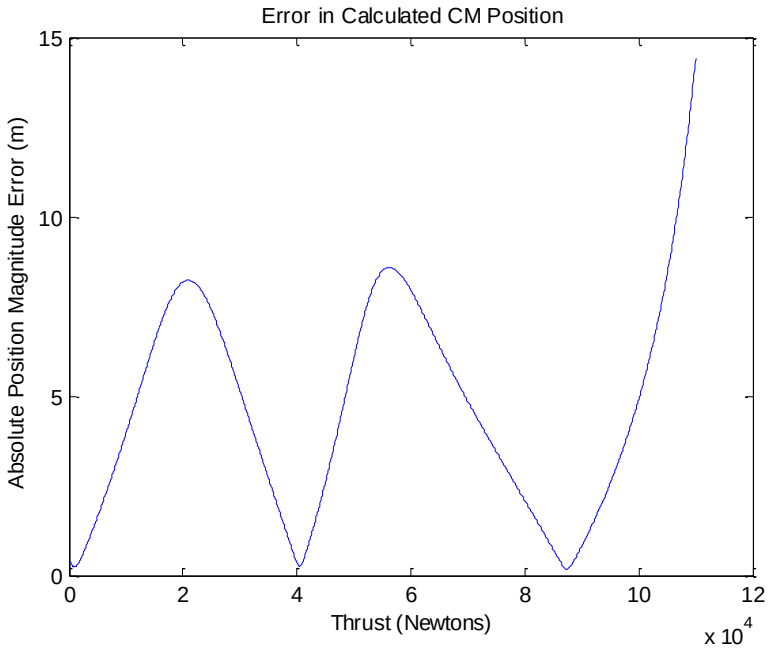


Fig. 18 Calculated CM Distance Error (High Density)

Comparison of Case Results

The most striking difference in the simulation output described above is that between the low density case and medium and high density case. The low density case yields singularities in both the calculation of the total system mass and the position vector to the system center of mass, while the medium and high density cases result in oscillatory behavior in those same quantities. It is unknown at this time what the cause of the singularities is – though it is possible that it is more difficult to determine system mass properties for a lower mass asteroid due to its reduced effect on those properties. The singularities do, however, demonstrate that this method is not effective for low density asteroids of the size prescribed here.

Oscillations in the calculated mass properties for the medium and high density cases were unexpected. Despite the oscillations, there were extensive regions of thrust for both cases that yielded mass estimates within 10% of their true value. Those regions were notably larger for the medium density

case than for the high density case. For both higher density cases, obtaining an accurate estimate of the system center of mass is difficult due to the ill-matched oscillations in the three vector components. There were three points over the range of thrust used here at which an accurate estimate of the CM location could be obtained. In a realistic application of this method, this limitation could prove very difficult to implement. It would require very exact knowledge of the thrust values resulting in minimum error in order to avoid destabilization of the system through inaccurate determination of the CM location.

Chapter 6

Conclusions and Future Work

The linear, deterministic mass property characterization method used in this research has been proven to have the ability to determine the total system mass and the location of the system center of mass to within 10% error for specific cases. The method functions best for an asteroid of medium density, such as the one NASA is considering for the ARM. It also functions relatively well for asteroids of higher density, while the low density region is likely inoperable within the context of this research.

Further research into the cause of the singularities and oscillations is necessary to further evaluate the efficacy of this method in determining system mass properties. Investigations of the low thrust region (< 100 N) and its impact on this method should also be undertaken, since the real Asteroid Redirect Mission will likely employ low-thrust electric propulsion instead of high-thrust chemical propulsion. Other ways of increasing the robustness and accuracy of the method could be to represent the attitude of the spacecraft with quaternions to eliminate singularities or use a more realistic model for the asteroid mass distribution. Beyond more accurately determining the system mass and the location of the center of mass, the next leap for this research path is to determine the principal moments of inertia of the docked system and the orientation of the principal axis system. With these four mass properties, it should be possible to reliably and precisely control the attitude and orbital dynamics of a docked system involving an uncooperative body.

Appendix A

torqued_euler.m

```
% Integrates the 3D equations of motion for a rigid body of initial angular
% velocity, w0, and principal inertia matrix, I, over time span, tspan, in
% response to input moment vector, M.
function [t,w] = torqued_euler(w0,tspan,I,M)

options = odeset('RelTol',1e-9);
[t,w] = ode45(@euler_eqns,tspan,w0,options);

    function wdot = euler_eqns(t,w)
        wdot = zeros(3,1);
        wdot(1) = (M(1)/I(1,1))+((I(2,2)-I(3,3))/I(1,1))*w(2)*w(3);
        wdot(2) = (M(2)/I(2,2))+((I(3,3)-I(1,1))/I(2,2))*w(1)*w(3);
        wdot(3) = (M(3)/I(3,3))+((I(1,1)-I(2,2))/I(3,3))*w(1)*w(2);
    end

end
```

Appendix B

objectEOMs.m

```
function [w_b, accel_top, accel_center, accel_side1, accel_side2, t, r_cm, M_tot] =
objectEOMs(imp_thrust, M_s, I_s, d,tspan)

%Set  up body centered principal axes of spacecraft
b = [1 0 0; 0 1 0; 0 0 1];

%Model asteroid as a dumbbell shape (two spheres at the ends of a cylinder)
M_a = 15000; %total mass of asteroid
m_component = M_a/3;
radius_dumb = 0.5; %meters
length_dumb = 1; %meters
radius_sphere = 1; %meters
I_sphere = (2/5)*(m_component*radius_sphere^(2)).*[1,0,0;0,1,0;0,0,1]; %Inertia matrix of spheres
I_dumb =
[m_component*length_dumb^(2)/12,0,0;0,m_component*length_dumb^(2)/12,0;0,0,m_component*radius_dum
b^(2)/2]; %Inertia matrix of middle cylinder
I_a = zeros(3,3);
for i = 1:3
    if i == 1 || i == 2
        I_a(i,i) =
(2*I_sphere(i,i))+I_dumb(i,i)+(2*m_component*(radius_sphere+(length_dumb/2))^(2));
    else
        I_a(i,i) = I_dumb(i,i) + 2*I_sphere(i,i);
    end
end

M_tot = M_s + M_a; %Total mass of composite body

%Specify orientation of asteroid
phi = pi/6;
theta = pi/4;
psi = pi/3;

%Vector between CMs
r_a = [1.5;1;7];%meters, in b frame

%Vector to composite CM in b-frame
r_cm = (M_a/(M_s+M_a)).*r_a;

DCM_ab = [(cos(phi)*cos(psi)-sin(phi)*cos(theta)*sin(psi)),
(sin(phi)*cos(psi)+cos(phi)*cos(theta)*sin(psi)), (sin(theta)*sin(psi));
          (-cos(phi)*sin(psi)-sin(phi)*cos(theta)*cos(psi)), (-
sin(phi)*sin(psi)+cos(phi)*cos(theta)*cos(psi)), (sin(theta)*cos(psi));
          (sin(phi)*sin(theta)), (-cos(phi)*sin(theta)), (cos(theta))];

%Reorient inertia matrix of asteroid to align with that of s/c
I_ab = DCM_ab*I_a*DCM_ab';

%Use parallel axis theorem to bring inertia matrix to new center of mass
I_cm = zeros(3,3);
chi = 1 - (M_a/(M_a+M_s)); %Create proxy variable for simplicity
for j = 1:3
    for k = 1:3
        I_cm(j,k) = I_s(j,k)+I_ab(j,k) + (r_a(j)*r_a(k))*(M_s*(1-chi)^(2) + M_a*chi^(2));
    end
end
```

```
%From the eigenvalue problem, a cubic polynomial results. We solve here.
alpha = 1;
beta = -I_cm(1,1)-I_cm(2,2)-I_cm(3,3);
gamma = I_cm(1,1)*I_cm(2,2) + I_cm(1,1)*I_cm(3,3) + I_cm(2,2)*I_cm(3,3)-I_cm(1,2)^(2)-
I_cm(1,3)^(2)-I_cm(2,3)^(2);
delta = I_cm(1,1)*I_cm(2,3)^(2)+I_cm(2,2)*I_cm(1,3)^(2)+I_cm(3,3)*I_cm(1,2)^(2)-
(I_cm(1,1)*I_cm(2,2)*I_cm(3,3))+2*(I_cm(1,2)*I_cm(1,3)*I_cm(2,3));

del0 = beta^(2)-3*alpha*gamma;
del1 = 2*beta^(3)-9*alpha*beta*gamma+27*alpha^(2)*delta;

C = ((del1+sqrt(del1^(2)-4*del0^(3)))/2)^(1/3);

%Cubic roots of unity
im  = sqrt(-1);
u = [1, (-1+im*sqrt(3))/2, (-1-im*sqrt(3))/2];

%Principal inertia tensor of composite body
I_p = zeros(3,3);
for n = 1:3
    I_p(n,n) = real((-1/(3*alpha))*(beta + u(n)*C + (del0/(u(n)*C))));
end

%Check calculation against MATLAB eigenvalue/vector calculator, and find
%eigenvectors
[princ_axes,I_pcheck] = eig(I_cm);
if I_pcheck ~= I_p
    fprintf('Error calculating principal MOIs')
end



%Find DCM between b frame and new principal axis frame.
DCM_princb = zeros(3,3);
DCM_bprinc = zeros(3,3);
for p = 1:3
    for q = 1:3
        DCM_princb(p,q) = dot(b(:,p),princ_axes(:,q));
        DCM_bprinc(p,q) = dot(princ_axes(:,p),b(:,q));
    end
end

%With the composite body now defined within this function, we must now
%return the measurements of angular velocity and acceleration that could be
%measured on-board the spacecraft. We assume the thrust is along the b3
%axis.
thrust_p = (DCM_bprinc*[0;0; imp_thrust])'; %Impulsive thrust is considered to cause a finite
change in momentum
r_cm_p = (DCM_bprinc*(-r_cm))'; %vector FROM composite CM TO s/c CM

moment_p = cross(r_cm_p,thrust_p);

%Use Euler's equations to find angular velocity over the period of the
%thrust.
[t,w_p] = torqued_euler([0;0;0],tspan,I_p,moment_p);

ang_accel = zeros(length(w_p),3);
for Q = 1:length(w_p)
    ang_accel(Q,1) = (moment_p(1)/I_p(1,1))+((I_p(2,2)-I_p(3,3))/I_p(1,1)).*w_p(Q,2).*w_p(Q,3);
    ang_accel(Q,2) = (moment_p(2)/I_p(2,2))+((I_p(3,3)-I_p(1,1))/I_p(2,2)).*w_p(Q,1).*w_p(Q,3);
    ang_accel(Q,3) = (moment_p(3)/I_p(3,3))+((I_p(1,1)-I_p(2,2))/I_p(3,3)).*w_p(Q,1).*w_p(Q,2);
end

w_b = zeros(length(w_p),3);
for q = 1:length(w_p)
```

```
    w_b(q,:) = real(DCM_princb*w_p(q,:)')'; %These should be the approximate ang. velocity
measurements returned by sensors on the s/c
end

accel_center = zeros(length(w_p),3);
accel_top = zeros(length(w_p),3);
accel_side1 = zeros(length(w_p),3);
accel_side2 = zeros(length(w_p),3);

r_top = (r_cm_p) + ((DCM_bprinc*[0; 0; d]))'; %along b3 axis
r_side1 = (r_cm_p) + ((DCM_bprinc*[0; d; 0]))'; %along b2 axis
r_side2 = (r_cm_p) + ((DCM_bprinc*[d; 0; 0]))'; %along b1 axis
%Find acceleration of CM in 3D from thrust input and system mass. We
%consider the acceleration measurements to have been taken during the
%impulsive thrust.
accel_cm = (thrust_p./M_tot); %Acceleration of composite CM
for s = 1:length(w_p)
    %Measured acceleration of CM of spacecraft in p frame
    accel_center(s,:) = [accel_cm(1)+(-(w_p(s,2)^(2)+w_p(s,3)^(2)))*r_cm_p(1)+(w_p(s,1)*w_p(s,2)-
ang_accel(s,3))*r_cm_p(2)+(w_p(s,1)*w_p(s,3)+ang_accel(s,2))*r_cm_p(3),
                         accel_cm(2)+(w_p(s,1)*w_p(s,2)+ang_accel(s,3))*r_cm_p(1)+(-
(w_p(s,1)^(2)+w_p(s,3)^(2)))*r_cm_p(2)+(w_p(s,2)*w_p(s,3)-ang_accel(s,1))*r_cm_p(3),
                         accel_cm(3)+(w_p(s,1)*w_p(s,3)-
ang_accel(s,2))*r_cm_p(1)+(w_p(s,2)*w_p(s,3)+ang_accel(s,1))*r_cm_p(2)+(-
(w_p(s,1)^(2)+w_p(s,2)^(2)))*r_cm_p(3)];
    %Measured acceleration at b3 accelerometers in p frame
    accel_top(s,:) = [accel_cm(1)+(-(w_p(s,2)^(2)+w_p(s,3)^(2)))*r_top(1)+(w_p(s,1)*w_p(s,2)-
ang_accel(s,3))*r_top(2)+(w_p(s,1)*w_p(s,3)+ang_accel(s,2))*r_top(3),
                      accel_cm(2)+(w_p(s,1)*w_p(s,2)+ang_accel(s,3))*r_top(1)+(-
(w_p(s,1)^(2)+w_p(s,3)^(2)))*r_top(2)+(w_p(s,2)*w_p(s,3)-ang_accel(s,1))*r_top(3),
                      accel_cm(3)+(w_p(s,1)*w_p(s,3)-
ang_accel(s,2))*r_top(1)+(w_p(s,2)*w_p(s,3)+ang_accel(s,1))*r_top(2)+(-
(w_p(s,1)^(2)+w_p(s,2)^(2)))*r_top(3)];
    %Measured acceleration at b2 accelerometers in p frame
    accel_side1(s,:) = [accel_cm(1)+(-(w_p(s,2)^(2)+w_p(s,3)^(2)))*r_side1(1)+(w_p(s,1)*w_p(s,2)-
ang_accel(s,3))*r_side1(2)+(w_p(s,1)*w_p(s,3)+ang_accel(s,2))*r_side1(3),
                         accel_cm(2)+(w_p(s,1)*w_p(s,2)+ang_accel(s,3))*r_side1(1)+(-
(w_p(s,1)^(2)+w_p(s,3)^(2)))*r_side1(2)+(w_p(s,2)*w_p(s,3)-ang_accel(s,1))*r_side1(3),
                         accel_cm(3)+(w_p(s,1)*w_p(s,3)-
ang_accel(s,2))*r_side1(1)+(w_p(s,2)*w_p(s,3)+ang_accel(s,1))*r_side1(2)+(-
(w_p(s,1)^(2)+w_p(s,2)^(2)))*r_side1(3)];
    %Measured acceleration at b1 accelerometers in p frame
    accel_side2(s,:) = [accel_cm(1)+(-(w_p(s,2)^(2)+w_p(s,3)^(2)))*r_side2(1)+(w_p(s,1)*w_p(s,2)-
ang_accel(s,3))*r_side2(2)+(w_p(s,1)*w_p(s,3)+ang_accel(s,2))*r_side2(3),
                         accel_cm(2)+(w_p(s,1)*w_p(s,2)+ang_accel(s,3))*r_side2(1)+(-
(w_p(s,1)^(2)+w_p(s,3)^(2)))*r_side2(2)+(w_p(s,2)*w_p(s,3)-ang_accel(s,1))*r_side2(3),
                         accel_cm(3)+(w_p(s,1)*w_p(s,3)-
ang_accel(s,2))*r_side2(1)+(w_p(s,2)*w_p(s,3)+ang_accel(s,1))*r_side2(2)+(-
(w_p(s,1)^(2)+w_p(s,2)^(2)))*r_side2(3)];
end

%If the three sets of accelerometers are spaced equally at known distances
%apart from one another along the main axis of the spacecraft, we can
%adjust the radius vectors to fit the location of the measurement.
for count = 1:length(w_p)
    accel_center(count,:) = (DCM_princb*(accel_center(count,:))')'; %Measured acceleration of CM
of spacecraft in b frame
    accel_top(count,:) = (DCM_princb*(accel_top(count,:))')'; %in b frame
    accel_side1(count,:) = (DCM_princb*(accel_side1(count,:))')'; %in b frame
    accel_side2(count,:) = (DCM_princb*(accel_side2(count,:))')'; %in b frame
end
end
```

Appendix C

main_simulation.m

```
Thrust = 100:100:110000;%Newtons
tspan = 0:0.02:5;
X_vect = zeros(length(Thrust),6);
M_s = 15000; %kg
rad_sc = 1.35; %meters
length_sc = 5.9; %meters
I_s = [M_s*length_sc^(2)/12,0,0;0,M_s*length_sc^(2)/12,0;0,0,M_s*rad_sc^(2)/2];
d = 1.35; %meters, distance between sets of accelerometers
for j = 1:length(Thrust)
    T = Thrust(j); %Newtons

    [w,a1,a2,a3,a4,t,r_cm,M_theory] = objectEOMs(T,M_s,I_s,d,tspan);

    da_topcenter = a1-a2;
    da_centerside1 = a2-a3;
    alpha = zeros(length(tspan),3);
    for i = 1:length(tspan)
        alpha1 = w(i,2)*w(i,3)+(da_topcenter(i,2)/d); %These equations result from subtracting
kinematic equations for acceleration at the various sensing points
        alpha2 = -(da_topcenter(i,1)/d)-w(i,1)*w(i,3);
        alpha3 = w(i,1)*w(i,2)-(da_centerside1(i,1)/d);
        alpha(i,:) = [alpha1,alpha2,alpha3]; %angular acceleration of body expressed in b-frame
        %alpha_vect(j,i,:) = alpha;
        %w_vect(j,:,:) = w;
    end

    t1_loc = ceil(length(tspan)/2);
    t2_loc = t1_loc + 70;
    %"A" matrix is composite matrix consisting of 4 3x3 component matrices.
    A1133 = eye(3);
    A4163 = eye(3);
    A1436 = [-(w(t1_loc,2)^(2)+w(t1_loc,3)^(2)),w(t1_loc,1)*w(t1_loc,2)-
alpha(t1_loc,3),w(t1_loc,1)*w(t1_loc,3)+alpha(t1_loc,2);
            w(t1_loc,1)*w(t1_loc,2)+alpha(t1_loc,3),-
(w(t1_loc,1)^(2)+w(t1_loc,3)^(2)),w(t1_loc,2)*w(t1_loc,3)-alpha(t1_loc,1);
            w(t1_loc,1)*w(t1_loc,3)-alpha(t1_loc,2),w(t1_loc,2)*w(t1_loc,3)+alpha(t1_loc,1),-
(w(t1_loc,1)^(2)+w(t1_loc,2)^(2))];
    A4466 = [-(w(t2_loc,2)^(2)+w(t2_loc,3)^(2)),w(t2_loc,1)*w(t2_loc,2)-
alpha(t2_loc,3),w(t2_loc,1)*w(t2_loc,3)+alpha(t2_loc,2);
            w(t2_loc,1)*w(t2_loc,2)+alpha(t2_loc,3),-
(w(t2_loc,1)^(2)+w(t2_loc,3)^(2)),w(t2_loc,2)*w(t2_loc,3)-alpha(t2_loc,1);
            w(t2_loc,1)*w(t2_loc,3)-alpha(t2_loc,2),w(t2_loc,2)*w(t2_loc,3)+alpha(t2_loc,1),-
(w(t2_loc,1)^(2)+w(t2_loc,2)^(2))];
    A = [A1133,A1436;A4163,A4466]; %linear system of equations
    b = [a2(t1_loc,:)';a2(t2_loc,:)'];%observation vector
    %We now have a system of linear equations of the form b = AX, with X
    %containing the three components of the CM position vector and the three components of the CM
    %acceleration vector. We solve using linsolve.
    opts.RECT = true;
    X = linsolve(A,b,opts);

    X_vect(j,:) = X;
end
%Mean calculated CM position vector
%fprintf('The mean calculated CM position vector in meters is:\n\n\trx = %d\n\try = %d\n\trz =
%d\n',mean(X_vect(:,4)),mean(X_vect(:,5)),mean(X_vect(:,6)));

%Finding total mass
%We will use acceleration in the b3 direction to find mass.
Mtotal = Thrust'./X_vect(:,3);
fprintf('The mean estimated mass is %d kg.\n',mean(Mtotal));
```

```
pos_error = zeros(length(X_vect),3);
mean_error = zeros(length(X_vect),1);
mass_error = zeros(length(X_vect),1);
for k = 1:length(X_vect)
    pos_error(k,:) = X_vect(k,4:6)+r_cm'; %Position vector error
    mean_error(k) = norm(pos_error(k,:)); %Absolute distance error
    mass_error(k) = (abs(Mtotal(k)-M_theory)/M_theory)*100; %Percent error in mass
end

figure
plot(Thrust, pos_error(:,1),Thrust, pos_error(:,2),Thrust, pos_error(:,3))
xlabel('Thrust (Newtons)')
ylabel('Absolute Position Error (m)')
title('Error in Calculated CM Position')
legend('x','y','z')

figure
plot(Thrust, mean_error)
xlabel('Thrust (Newtons)')
ylabel('Absolute Position Magnitude Error (m)')
title('Error in Calculated CM Position')

figure
plot(Thrust,Mtotal)
xlabel('Thrust (Newtons)')
ylabel('Estimated Total System Mass (kg)')
title('Calculated Total Mass')

figure
plot(Thrust,mass_error)
xlabel('Thrust (Newtons)')
ylabel('Calculation Error (%)')
title('Percent Error in Mass Calculation')
```

REFERENCES

[1]Bandyopadhyay, S., Chung, S., and Hadaegh, F., "Attitude Control and Stabilization of Spacecraft with a Captured Asteroid," *AIAA Guidance, Navigation, and Control Conference*. AIAA, Washington, DC, 2015. pp. 1-27.

[2]Merrill, R.G., Qu, M., Vavrina, M.A., Englander, J.A., Jones, C.A., "Interplanetary Trajectory Design for the Asteroid Robotic Redirect Mission Alternate Approach Trade Study," *AIAA/AAS Astrodynamics Specialist Conference*. AIAA, Washington, DC, 2014. pp. 1-13.

[3]Fourie, D., Tweddle, B.E., Ulrich, S., Saenz-Otero, A., "Flight Results of Vision-Based Navigation for Autonomous Spacecraft Inspection of Unknown Objects," *Journal of Spacecraft and Rockets*, Vol. 51, No. 6, 2014, pp. 2016-2026.

[4]Gaias G., D'Amico, S., Ardaens, J.-S., "Angles-Only Navigation to a Noncooperative Satellite Using Relative Orbital Elements," *Journal of Guidance, Control and Dynamics*, Vol. 37, No. 2, 2014, pp. 439-451.

[5]Brophy, J., Culick, F., and Friedman, L., "Asteroid Retrieval Feasibility Study," Tech. rep., Keck Institute for Space Studies, California Institute of Technology, Pasadena, CA, April 2012.

[6]Bazzocchi, M.C.F., Emami, M.R., "A Systematic Assessment of Asteroid Redirection Methods for Resource Exploitation," *8th Symposium on Space Resource Utilization*. AIAA, Washington, DC, 2015. pp. 2-4.

ACADEMIC VITA

Patrick Wittick
907 Old Boalsburg Road
State College, PA 16801
ptw5033@psu.edu

EDUCATION

The Pennsylvania State University, Schreyer Honors College
University Park, PA
Bachelor of Science, Aerospace Engineering
Minor in German
Anticipated Graduation: May 2015

PROFESSIONAL EXPERIENCE

Vibration Analysis Intern May 2014 - Present
KCF Technologies, Inc.
State College, PA

- Provided vibration analysis and machine diagnostics to customers in the oil & gas, automobile, and food manufacturing industries
- Represented the company through on-site sensor installations and regular customer interaction over e-mail, phone, and face-to-face meetings
- Implemented a new proprietary metric for tracking damage accumulation on machinery – this metric has become a primary sales driver and monitoring tool for the company
- Interaction with largest customer helped lead to decision to double business with KCF by December 2014, with possibility for seven-fold expansion in 2015

Undergraduate Researcher May 2014 - Present
Department of Aerospace Engineering, Penn State University

- Currently developing a dynamics model for the deep space capture of an uncooperative asteroid using MATLAB and Simulink
- Research project is main component of honors thesis

PROJECTS

NIA/NASA RASC-AL Design Competition August 2014-April 2015 (ongoing)
Team Lead, Mars EDL Pathfinder Mission Concept

- Responsible for overall systems engineering and team coordination for year-long national space system design competition
- Manages ground control subsystem concept design and assists with other subsystem tasks as needed

Software Engineering Autonomous Robot Challenge January 2015-April 2015 (ongoing)
Member, Coding Team

- Building code for autonomous balloon-popping robot
- Focus on wireless communication component

ASSOCIATION MEMBERSHIPS

AIAA Student Member — Fall 2014
Sigma Gamma Tau — Spring 2014

HONORS & AWARDS

DigitalGlobe Foundation Scholarship — Fall 2014
Dean's List — Fall 2011 – Present
Eagle Scout — August 2010