Automated Proximity Operations Using Image-Based Relative Navigation

Luke Walker
Georgia Institute of Technology, Atlanta, GA 30332-0510, USA

Advisor: David Spencer
Georgia Institute of Technology, Atlanta, GA 30332-0510, USA

This paper describes a system for relative navigation and automated proximity operations for a microsatellite using continuous thrust propulsion and low-cost visible and infrared imagers. Image processing algorithms provide range, range rate, and spherical angle estimates relative to a target spacecraft using knowledge of the target spacecraft's geometry. A differential correction batch filter is used to provide relative navigation and state estimation. These state estimates are used to provide input for the automated control of the chaser spacecraft via a Linear Quadratic Regulator. Propulsive maneuvers are accomplished via low-thrust, non-throttleable thrusters using pulse-width modulation and thrust vectoring. A waypoint logic controller is used to define intermediate goals to reach the final goal in order to limit operational risk from an error in estimation of the spacecraft's relative state. The system is described and simulation test results are shown.

I. Introduction

Advancements in automated orbit determination and safe trajectory control for close proximity operations are critical to Space Situational Awareness (SSA) capabilities. Previous efforts for the rendezvous and proximity operations of two spacecraft have involved ground in the loop during operations. It is highly desirable to increase the level of automation, such that future missions will be able to autonomously maneuver and inspect objects. Microsatellites are well-suited for on-orbit inspection applications in support of SSA objectives, satellite servicing, or as robotic service vehicles in support of human exploration missions.

A. Past Missions

Aspects of relative navigation, autonomous rendezvous and proximity operations have been demonstrated in several flight missions. These missions were primarily technology demonstrators, designed to demonstrate critical portions of the larger problem of autonomous behavior. These missions include DART , XSS-10 and XSS-11, and Orbital Express; their characteristics are examined below in

Table 1.

While these missions have demonstrated significant aspects of automated proximity operations, there are several key capabilities that have not been addressed. To perform relative navigation with another body, previous missions have used either an active sensor such as a laser range finder (DART, XSS-11, Orbital Express) or a precisely known state of the other spacecraft, either through GPS cross-link (DART) or an ejection point of the launch vehicle (XSS-10). A key capability for future missions involving orbital debris or non-cooperative spacecraft will be relative navigation without high-fidelity knowledge of the target object. Passive techniques are desirable for operations about non-cooperative spacecraft. Secondly, all previous missions have used impulsive thrust maneuvers to perform trajectory control. All missions have used some sort of hydrazine propulsion system, which allowed them to generate enough thrust for impulsive maneuvers. The use of electric propulsion and other low-thrust options is a desirable feature for future missions, given their ability to increase mission ΔV capability. Finally, most of these missions exhibited some form of "ground in the loop" behavior during critical maneuvers, such as Orbital Express' docking and fluid transfer. Increasing the level of autonomy for proximity operations is also desirable, as objectives for SSA and on-orbit inspection becomes more advanced.

Table 1. Current Proximity Operations Missions

Mission (Year)	Agency	Mass	Propulsion	Key Technology
DART (2005) [1]- [2]	NASA	363 kg	Hydrazine	Laser Reflectance GPS Cross-Link
XSS-10 (2003) [3]-[4]	AFRL	27 kg	MMH / N204	Relative Propagation Semi-Autonomous Behavior
XSS-11 (2005) [5]-[8]	AFRL	138 kg	Hydrazine	Lidar Autonomous Planner
Orbital Express (2007) [9]-[10]	DARPA	952 kg 226 kg	Hydrazine	Laser Ranging Laser Reflectance Autonomous Docking

B. Relative Navigation

A critical problem within the area of automated rendezvous and proximity operations is relative navigation, also referred to as relative orbit determination. A maneuver strategy for a close proximity orbit must be based upon knowledge of the orbit relative to the target object. A variety of strategies may be employed to determine these relative states. First, ground-based state estimates for spacecraft provide an inertial solution for positions; these estimates are often in the form of a Two Line Element (TLE) or Vector Covariance Message (VCM) from the US Joint Operations Center. However, TLEs and VCMs both offer positional accuracies of only kilometers, far too large for most proximity operations. [10] Alternatively, if all spacecraft are operating GPS receivers and are cooperative, the relative position of the spacecraft can be known to less than a meter. However, if one or more spacecraft are inoperable, uncooperative, or do not possess GPS receivers, this method is not valid. A combination of an active range finder (e.g. Lidar) and imagers offers solutions for both range and angle estimates of the relative position; this is the approach utilized by most of the previous proximity operations missions. As previously mentioned, there is an interest in removing the active sensing portion from this sensing process, particularly for uncooperative targets. However, the current usage of imagers has been limited to angle estimation. It can be shown that for given line-of-sight vectors provided by angle estimates, the solution for the relative orbit is non-unique; only families of relative orbits can be determined. [11] These non-unique solutions will not suffice for automated proximity operations; therefore a range estimate must be incorporated.

C. Automated Maneuver Planning

A second critical aspect of automated proximity operations is the maneuver planning required for the operations. The maneuver planning must take the spacecraft from its estimated relative state and move it to a desired position, given the designed mission profile, by defining a series of propulsive maneuvers. Within automated proximity operations missions, these propulsive maneuvers have been traditionally defined as impulsive maneuvers, applying a change in velocity (ΔV) instantaneously. This method allows for simpler maneuver definition and orbit propagation, especially in the linear Clohessy-Wiltshire framework. Low-thrust, non-impulsive automated maneuver planning has not been used on-orbit, although these systems present several advantages over high-thrust propulsion systems. Low-thrust systems offer high specific impulse, improved ΔV capability, and lower tank pressures than high-thrust propulsion systems. These attributes are particularly useful for mass constrained microsatellites designed for launch as secondary payloads.

D. Summary

It is clear that the field of automated proximity operations, including relative navigation and automated maneuver planning, is very important to future SSA capabilities and offers several important areas for development. The rest of this paper will describe a proposed system to advance these capabilities by utilizing imagers to perform range and angle estimate for relative navigation and a simple automated maneuver planning strategy utilizing continuous thrust propulsion. Section II will describe the system components and interactions, Section III will provide an overview of the Simulation, Analysis, and Testing required, and Section IV will describe the future work for the system.

II. System Overview

The Auto-Navigation System (AutoNav) was designed based upon a survey of the existing proximity operations capabilities and the requirements for future systems. As highlighted previously, the areas of passive imaging-based relative navigation and automated low-thrust propulsion maneuver planning were identified as key

areas of interest. The basic design goals of the system were developed from these areas:

1. Rendezvous and proximity operations with target spacecraft using continuous thrust propulsion.
2. Automated relative navigation and control on-board spacecraft.
3. Closed-loop attitude control based upon automated image processing.
4. Relative orbit determination using angle and range estimates from low-cost imagers and image processing.

These design goals influenced the design of the system and helped generate the requirements flowdown.

A. Subsystem Interaction

There are several spacecraft subsystems and software components that must interact for the system to perform effectively. A basic block diagram of the system interactions can be seen in Figure 1. The critical subsystems will be described in more detail in the following sections.

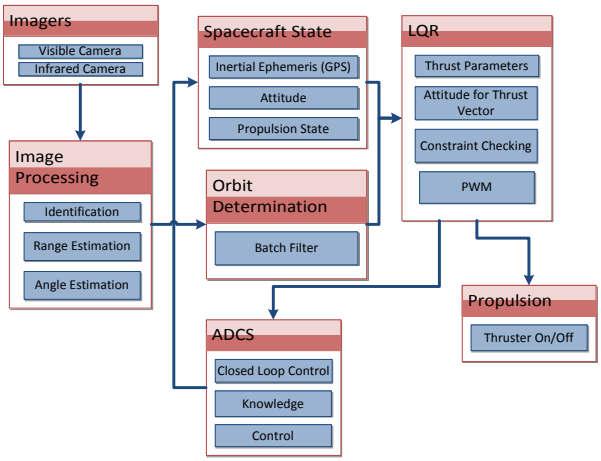

Figure 1. Block Diagram of AutoNav System

B. Image Processing Algorithms

Image Processing Algorithms (IPA) are used to process the images taken by the imaging instruments in order to provide positional information about the target spacecraft. This section summarizes the operations undertaken within each step; for a more detailed explanation, see [12].

Identification of Target Spacecraft

The first step in the AutoNav process is to identify the target spacecraft within acquired images. The "Blobber" algorithm is used to identify areas of contiguous pixels with similar intensities, blobs (Binary Large OBject), to find the target spacecraft on an image. Size and intensity screens can be used to filter out the blobs that cannot be the target spacecraft.

The successful results of the Blobber algorithm are a calculated area for the identified blob and the location of the Center of Brightness (COB) for the blob. The COB is similar to an area centroid and is used as the central location of the CubeSat. An example of the resulting images from the Blobber algorithm can be seen in Figure 2. The image on the left was taken with a thermal infrared camera of a heated 3U CubeSat from a range of 50m. The image on the right is the resulting image after processing.

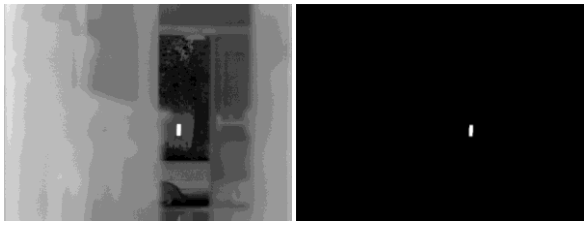

Figure 2. Blobber Algorithm Results from Infrared Camera

Unit Vector Determination

Once the target has been identified, its relative location with respect to the chaser may be determined. Using the COB coordinates, this location can be determined based upon knowledge of imager optics and alignment. The unit vector $\underline{u}$ can be determined using the focal length of the lens and the coordinates of the COB. Alternatively, the position vector of the target can be expressed in spherical coordinate. (1) and (2) defines the spherical angles from the COB coordinates, and (3) shows how the unit vector is calculated from the spherical angles.

$$\theta = \operatorname{atan}\left(\frac{\sqrt{p^2(x_{COB}^2 + y_{COB}^2)}}{f}\right) \quad (1)$$

$$\phi = \operatorname{atan2}(\mathrm{y_{COB}}, \mathrm{x_{COB}}) \quad (2)$$

$$\underline{u} = \begin{bmatrix} \hat{x}_{FPF} \\ \hat{y}_{FPF} \\ \hat{z}_{FPF} \end{bmatrix} = \begin{bmatrix} \sin(\theta_i)\cos(\phi_i) \\ \sin(\theta_i)\sin(\phi_i) \\ \cos(\theta_i) \end{bmatrix} \quad (3)$$

p = Pixel Pitch of Imager

f = Focal Length of Imager

Range Estimation

Once the target has been identified and its unit vector determined, the range from chaser to target is determined. Range estimation is based upon the ratio between the sensed area of the blob by the imager and the projected area of the target spacecraft. However, the actual projected area of the spacecraft is unknown, since the orientation of the spacecraft is unknown. The following steps are used to resolve this ambiguity.

1. **Determine the major and minor axes of the blob**
   A 2-D proxy for spacecraft orientation is the combination of major and minor axes of the blob; these axes are determined in conventional 2-D methods using the pixel locations and the COB coordinates.

2. **Calculate the ratio of axis lengths**
   Given that the image provides only two dimensions of information, it is impossible to determine accurately the orientation in all three axes, but this ratio provides a parameter useful for estimating projected area.

3. **Estimate minimum and maximum projected areas**
   Using the ratio of the axis lengths, a range of projected areas can be determined. In order to accomplish this, numerical approximations for the projected area as a function of axis ratio must be derived. For this case, a 3U (10cm x 10cm x 30cm) CubeSat was used. Numerical approximations for minimum, maximum, and mean projected areas were derived. Figure 3 shows these samples and numerical functions. The samples in the distribution were generated by stepping through possible orientations of the CubeSat from the perspective of an observer. The x-axis in the plot is the ratio of major axis length to minor axis length, and the y-axis is the projected area of the target, normalized by a reference area.

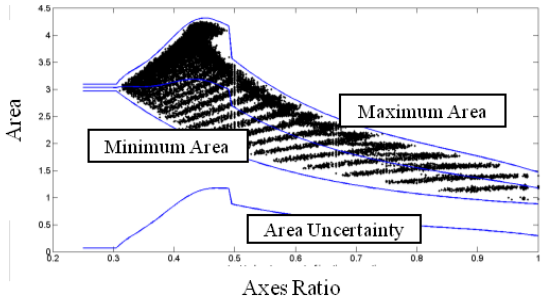

Figure 3. Projected Area as a Function of Axis Length Ratio

4. **Estimate Range and Uncertainty**
   Using the numerical approximation for projected area of the target, an estimate of the range can be determined, using (4), where A_{mean}(Axes Ratio) is the experimentally derived function for area.

$$\rho_{Mean} = f\sqrt{\frac{\mathrm{A_{mean}(Axes\ Ratio)}}{N_{blob}p^2}} \qquad (4)$$

The end result of the IPA function is a range estimate, a unit vector or rotation angles in BFF, and the uncertainty estimate for the range. This information will then be fed into the orbit determination filter that will help determine the relative motion of the chaser spacecraft.

C. Relative Navigation

The estimates from the IPAs provide the basis for relative navigation and orbit determination. Proximity operations will be accomplished using the Clohessy-Wiltshire (CW) equations in the Local-Vertical Local Horizontal (LVLH) frame, also known as the RSW frame. Therefore, unlike traditional orbit determination, this process will be focused not on inertial state but on relative state and relative orbital elements (ROEs).

Orbital Dynamics

The relative motion of two bodies in orbit about a third body in close proximity to each other has been studied and characterized. In particular, Hill [13] and Clohessy-Wiltshire [14] described a linear, time invariant system where a "chaser" spacecraft's relative motion about a "chief" spacecraft in circular orbit is described. The basic dynamics are described by (5) as modified by Vallado [15].

$$\begin{aligned} \ddot{x} - 2n\dot{y} - 3\mathrm{n}^2 x &= f_x \\ \ddot{y} + 2n\dot{x} &= f_y \\ \ddot{z} + n^2 z &= f_z \end{aligned} \qquad (5)$$

These dynamics are expressed using the RSW coordinate system. $\hat{R}$ is the radial component, collinear with the position vector. $\hat{S}$ is the in-track component, in the direction of the Chief's velocity vector for a circular chief; formally, it is $\hat{W}x\hat{R}$. $\hat{W}$ is the cross-track component, normal to the orbital plane, or$\hat{R}x\hat{V}$, where $\hat{V}$ is the unit vector for the velocity of the chief.

The motion of the chaser may also be defined by Relative Orbital Elements (ROEs), analogous to orbital elements, as defined by Lovell [16]. These ROEs allow for the motion to be described in an intuitive way that allows for relative orbit design similar to the way a mission planner would design an inertial orbit using orbital elements. The transformations from relative position and velocity to ROEs can be seen in (6)-(11), where n is the mean orbital motion.

$$a_e = 2\sqrt{\left(\frac{\dot{x}}{n}\right)^2 + \left(3x + 2\frac{\dot{y}}{n}\right)^2} \quad (6)$$

$$x_d = 4x + \frac{2\,\dot{y}}{n} \quad (7)$$

$$y_d = y - 2\frac{\dot{x}}{n} \quad (8)$$

$$\beta = atan2(\dot{x}, 3nx + 2\dot{y}) \quad (9)$$

$$z_{max} = \sqrt{\left(\frac{\dot{z}}{n}\right)^2 + z^2} \quad (10)$$

$$\gamma = atan2(nz, \dot{z}) - \beta \quad (11)$$

Differential Correction Batch Filter

Given the dynamics of the system described above, a filter can be used to estimate the relative state of the chaser spacecraft, given the range and angle estimates provided by the IPAs. A differential correction batch filter using non-linear least squares estimation was used. The basics of the batch filter are described below; the formulation follows [15] with some adaptations.

Mathematical Model

The mathematical model that forms the backbone of the filter is based upon the orbital dynamics described previously. The model uses differential corrections to change the estimate for the state of the spacecraft at a particular time. There are two main components in this model, the state transition matrix (STM) and the mapping matrix.

Since this system has a closed form solution for the state, the STM and the state may be calculated directly. The equations for the STM and the state are shown in (12)-(13).

$$\boldsymbol{\Phi}(t) = \begin{bmatrix} 4 - 3\cos(nt) & 0 & 0 & \frac{\sin(nt)}{n} & \frac{2-2\cos(nt)}{n} & 0 \\ 6(\sin(nt) - nt) & 1 & 0 & \frac{2\cos(nt)-2}{n} & \frac{4\sin(nt)-3nt}{n} & 0 \\ 0 & 0 & \cos(nt) & 0 & 0 & \frac{\sin(nt)}{n} \\ 3n\sin(nt) & 0 & 0 & \cos(nt) & 2\sin(nt) & 0 \\ 6n(\cos(nt) - 1) & 0 & 0 & -2\sin(nt) & -3 + 4\cos(nt) & 0 \\ 0 & 0 & -n\sin(nt) & 0 & 0 & \cos(nt) \end{bmatrix} \quad (12)$$

$$\underline{X}(t - t_0) = \boldsymbol{\Phi}(t - t_0)\underline{X}(t_0) \quad (13)$$

These equations hold for the unforced, or homogeneous, solution. For the thrusting case, a closed-form to the differential equations exists if the thrust is constant in magnitude and direction. For this case, a particular solution is added to the homogeneous solution, as seen in (14)-(15).

$$\underline{X} = \underline{X}_{\boldsymbol{h}} + \underline{X}_{\boldsymbol{p}} = \boldsymbol{\Phi}\underline{X} + \boldsymbol{\Psi}\underline{u} \quad (14)$$

$$\boldsymbol{\Psi}(t) = \begin{bmatrix} \frac{1}{n^2} - \frac{\cos(nt)}{n^2} & \frac{2t}{n} - \frac{2\sin(n*t)}{n^2} & 0 \\ -\frac{2t}{n} + \frac{2\sin(nt)}{n^2} & \frac{4}{n^2} - \frac{3t^2}{2} - \frac{4\cos(nt)}{n^2} & 0 \\ 0 & 0 & \frac{1}{n^2} - \frac{\cos(nt)}{n^2} \\ \frac{\sin(nt)}{n} & \frac{2}{n} - \frac{2\cos(nt)}{n} & 0 \\ -\frac{2}{n} + \frac{2\cos(nt)}{n} & -3t + \frac{4\sin(nt)}{n} & 0 \\ 0 & 0 & \frac{\sin(nt)}{n} \end{bmatrix} \quad (15)$$

Next, the mapping matrix must be defined. The mapping matrix is used within the Normal Equations of least squares estimation. To define this matrix, the measurement model G(X) must be defined first, as seen in (16).

$$\begin{bmatrix} \rho \\ x_{FPF} \\ y_{FPF} \end{bmatrix} = \boldsymbol{G}(\underline{X}) \quad (16)$$

The mapping matrix is now defined as shown in (17)-(18).

$$\widetilde{\boldsymbol{H}}_{\boldsymbol{i}} = \frac{\partial \boldsymbol{G}(\underline{X})}{\partial \underline{X}} \quad (17)$$

$$\boldsymbol{H}_i = \widetilde{\boldsymbol{H}}_i \boldsymbol{\Phi}(t) \quad (18)$$

Filter Algorithm

The batch filter algorithm uses differential correction to modify an initial guess for the state of the spacecraft. The filter works by accumulating portions of the Normal Equations, used for least-squares estimation, for each measurement sample (for n samples) and then solving for the differential change in the estimated state. (19)-(20) show the accumulation of the Normal Equations for the samples, and (21)-(22) show the differential change.

$$\boldsymbol{L} = \sum_{i=1}^{n} \boldsymbol{H_i}^{\boldsymbol{T}} \boldsymbol{W_i} \boldsymbol{H_i} \quad (19)$$

$$\underline{M} = \sum_{i=1}^{n} \boldsymbol{H_i}^{\boldsymbol{T}} \boldsymbol{W_i} \delta \underline{y_i} \quad (20)$$

$$\delta \underline{X} = \boldsymbol{L}^{-1} \underline{M} \quad (21)$$

$$\underline{X_0^*} = \underline{X_0} + \delta \underline{X} \quad (22)$$

This newly estimated state then provides the next initial guess for the filter. A convergence criterion for the RMS of the residual values for each sample can be set, such that estimation is complete when the percent change in residual RMS is less than the criterion. When convergence is met, the best estimate for the state of the spacecraft at the specified time is given. Additionally, the covariance matrix for the

estimate may be calculated by inverting the final **L** matrix. [15]

$$\boldsymbol{P} = \boldsymbol{L}^{-1} \tag{23}$$

This covariance matrix can be used to define the confidence intervals for the state estimate. This covariance matrix can also be propagated forward with the equations of motion. However, since the STM is closed-form and can be computed directly for any time, it is easier to use (24) to determine the covariance matrix at any time after the estimate. [17]

$$\boldsymbol{P_1} = \boldsymbol{\Phi}(t_1, t_0)\, \boldsymbol{P_0}\, \boldsymbol{\Phi^T}(t_1, t_0) \tag{24}$$

D. Maneuver Planning

Once the current state of the spacecraft is estimated, the automated maneuver planning determines the necessary maneuvers to take the spacecraft from this state to the desired state from the mission profile. This section describes the basic maneuver planning strategy.

LQR Introduction

Linear Quadratic Regulators (LQR) provide a method to define the feedback control necessary to minimize a cost function for a linear system. Since the CW formulation of relative motion is a linear time-invariant (LTI) system, LQR control is well-suited. For general LQR theory, see [18]. The specific formulation will follow [19]-[20]. First, the linear dynamics of the system must be defined in a state-space model, as seen in (25)-(29).

$$\begin{aligned} \dot{\underline{X}}(t) &= \boldsymbol{A}\underline{X}(t) + \boldsymbol{B}\underline{u}(t) \\ \underline{Y}(t) &= \boldsymbol{C}\underline{X}(t) + \boldsymbol{D}\underline{u}(t) \end{aligned} \tag{25}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \tag{26}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{27}$$

$$\boldsymbol{C} = I_{6x6} \tag{28}$$

$$\boldsymbol{D} = 0 \tag{29}$$

A quadratic cost function that accounts for positional error and control effort is defined in (30).

$$J = \frac{1}{2}\int_0^\infty \left(\underline{X}_e^T \boldsymbol{Q} \underline{X}_e + \underline{u}^T \boldsymbol{R} \underline{u} + 2\underline{u}^T \boldsymbol{N} \underline{X}_e\right) dt \tag{30}$$

The tracking error $\underline{X}_e$ is defined as the difference in current state and desired state. The weighting matrices **Q** and **R** are now defined; the **N** matrix is set to zero.

$$\boldsymbol{Q} = \begin{bmatrix} \frac{\alpha_{Q_1}}{x_{max}^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\alpha_{Q_2}}{y_{max}^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\alpha_{Q_1}}{z_{max}^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\alpha_{Q_1}}{\dot{x}_{max}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\alpha_{Q_1}}{\dot{y}_{max}^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\alpha_{Q_1}}{\dot{z}_{max}^2} \end{bmatrix} \tag{31}$$

$$\boldsymbol{R} = \begin{bmatrix} \frac{\beta_{R_1}}{u_{x_{max}}^2} & 0 & 0 \\ 0 & \frac{\beta_{R_2}}{u_{y_{max}}^2} & 0 \\ 0 & 0 & \frac{\beta_{R_3}}{u_{z_{max}}^2} \end{bmatrix} \tag{32}$$

$$x_{max} = y_{max} = z_{max} = r_g = Distance\ to\ Goal \tag{33}$$

$$\dot{x}_{max} = \dot{y}_{max} = \dot{z}_{max} = \frac{r_{init}}{r_m} v_m \tag{34}$$

$$\alpha_{Q_1} = \alpha_{Q_2} = \alpha_{Q_3} = \alpha_{Q_4} = \alpha_{Q_5} = \alpha_{Q_6} = r_g \tag{35}$$

$$u_{x_{max}} = u_{y_{max}} = u_{z_{max}} = u_m = \frac{F_t}{m_s} \tag{36}$$

Using the state-space model and Q and R, the algebraic Riccati equation may be solved and used to find the gain matrix K that defines the control. For techniques on solving the algebraic Riccati equation, see [18]; the MATLAB function lqr has been used for this study. The control vector is then defined by (37).

$$\underline{u}(t) = -\boldsymbol{K}_{LQR} \cdot (\underline{X}(t) - \underline{X}_{goal}) \tag{37}$$

This control will be the desired acceleration vector for the spacecraft during the time before the next decision point. Thus, given an acceleration vector and a burn time between maneuver decisions, the proper orientation and thrust control may be determined. A combination of attitude control and Pulse-Width-Modulation (PWM) will be used for thruster control, to compensate for the limited thrusters available.

LQR Operations

The batch filter can only provide an updated state estimate when the target appears within the FOV of the chaser imagers. During active maneuvering it must be assumed that the state estimate will not be corrected. The lack of state update forces the

maneuver plan to operate in a quasi-open loop fashion: the spacecraft will thrust for some period of time before it is able to re-estimate its state and correct the maneuver plan. As the LQR control time grows, the uncertainty in state grows and the chance of re-contact or other problems grows. Because the maneuver time required grows as the distance grows between the initial and desired states, it is necessary to define a waypoint strategy, where the LQR control guides the spacecraft to smaller distance waypoints on the way to the target point. At each waypoint, thrusting is stopped and the target spacecraft is re-acquired in the FOV of the imagers. The estimated state is then updated using the new relative navigation solution, the next waypoint is defined, and control continues. This process occurs iteratively until the final position is within the bounds of waypoint definition and this state is reached. Waypoints are currently limited to a maximum of 15m away in-track. However, more complex waypoint strategies may be defined; see [21] for an automated "glideslope" algorithm.

III. Simulation, Analysis and Testing

The system must now be tested to evaluate its ability to perform the necessary automated proximity operations required for a proposed mission. Similar to hardware testing, the system will be tested on a component-level system first and then in a piece-wise integrated fashion. The following sections will describe several of these tests performed.

A. Open Loop Testing

Each component of the AutoNav system must be tested to analyze its accuracy and efficacy, and to analyze the impact of uncertainty on its performance. The two key components tested in this section are the the Rel-Nav orbit determination filter and the LQR system. For testing of the IPAs, see [12].

Relative Navigation Testing

The relative orbit determination batch filter can be tested using simulated data to evaluate its ability to estimate the state of the chaser spacecraft. In this case a reference relative orbit for the chaser spacecraft is defined, and then the state is taken at discrete periods of time. Using the measurement model previously described, the state is transformed into range and angle estimates. Gaussian noise is then added for each sample to evaluate the filter's ability to handle error and uncertainty. Once the OD filter provides an estimate for the state, the performance of the filter can be examined. In particular, the reduction in RMS residuals from the initial guess to the final solution, and the reduction in error of the state estimate are evaluated. Specifically, the norm of the state estimate error is measured.

Table 2 shows a comparison of six different cases tested with noisy data. In this case, errors were assumed to be 20% for range estimates and 0.05 rad for the two angle estimates; these errors were distributed normally about the true measurement. Three different orbits were tested: a simple trailing orbit at 50m, a static ellipse (a_e = 20, y_d = -75m), and a drifting ellipse (a_e = 20m, x_d = 2m, y_d = -55m). Each orbit was sample for two different durations, an entire orbital period and one-third of a period.

Table 2. Orbit Determination Filter Testing

Relative Orbit	Time	RMS (Initial)	X_{norm} Error (Initial)	RMS (Final)	X_{norm} Error (Final)
Trailing @ 50m	P	25.77	13.70	10.21	5.11
Static Ellipse	P	18.56	7.87	16.10	1.41
Drifting Ellipse	P	16.11	7.58	13.27	3.91
Trailing @ 50m	P/3	12.19	8.28	9.41	16.61
Static Ellipse	P/3	27.23	16.99	25.57	4.33
Drifting Ellipse	P/3	12.45	12.86	9.61	12.21

Several things may be gleaned from these tests. First, the initial orbit determination provides a reasonable estimate to begin the differential correction; on average, the initial X_{norm} error is 11.21m. Second, it is clear that a shorter duration for sampling has negative effects on final solution accuracy. It is clear that increased frequency of sampling is not sufficient; increased accuracy in sampling comes from sampling duration. Third, the batch filter is designed to minimize RMS, not necessarily final state error; therefore, in some cases the initial solution closes to a better state than the filter.

LQR Testing

The LQR guidance algorithm is tested as a controller for accuracy, rise and settle times, and control efficiency. The gains of the LQR system are set based upon these tests. For the purpose of these tests, the state knowledge is assumed to be continuous and perfect. The gains of the LQR were set by examining the rise time of the system and the fuel expended to reach a desired location. Rise time is defined to be the time required to reach 90% of the distance following a step input in the in-track direction. The results of the testing can be seen in Table 3; the gains selected are the ones boxed in. The resulting motion and thrust profile from these gains can be seen in

Figure 4. It is clear that the LQR controller effectively moves the spacecraft to the desired final state in under one orbital period.

Table 3. LQR Test Results

Inputs				Outputs	
r_{init}	r_m	v_m	Beta	Rise Time (Orbits)	Fuel Expended (kg)
100	75	0.1	10,000	0.85	7.90E-04
100	75	0.1	1,000	0.62	1.50E-03
100	75	0.1	100	0.45	4.00E-03
100	75	0.01	100	1.37	5.70E-03
100	75	0.5	100	0.40	3.90E-03
100	75	1	100	0.40	3.90E-03
100	150	0.5	100	0.41	3.90E-03
100	50	0.5	100	0.40	3.90E-03
100	25	0.5	100	0.40	3.90E-03

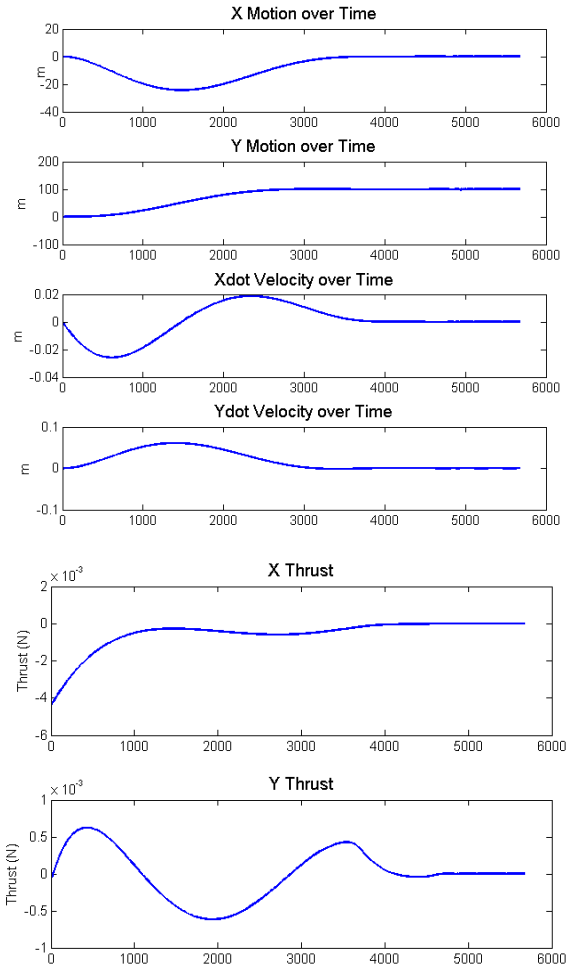

Figure 4. LQR Control Results

Next, the LQR controller must be evaluated for the influence of input estimate errors on final resulting state. Given an initial state estimate with an injected error, the LQR controller was used to control the estimated state to a desired state 100m in-track. Errors in X, Y, $\dot{X}$, and $\dot{Y}$ were examined individually, and then randomized errors were examined. The resulting errors were recorded after 3 orbits. These can be seen in Table 4.

Table 4. Results from Input Estimate Error

Input Estimate Error				Output Error	
X (m)	Y (m)	$\dot{X}$ (m)	$\dot{Y}$ (m)	Final Position (m)	Final Velocity (m/s)
10	0	0	0	377.12	0
-10	0	0	0	377.12	0
0	10	0	0	10.00	0
0	0	1e-3	0	0.00	1e-3
0	0	-1e-3	0	0.00	1e-3
0	0	0	1e-3	17.03	1e-3
0	0	0	-1e-3	17.03	1e-3
4.59	8.97	-1.3e-3	-2.2e-4	126.61	2.5e-3
-3.98	-1.63	-9e-4	7e-4	137.48	1.1e-3
-7.34	1.62	1e-3	-2e-4	281.20	1e-3
2.78	0.64	0	3e-4	108.73	3e-4
1.38	-0.63	2e-4	2e-4	49.75	3e-4
-1.02	-30.73	3e-4	-1e-4	10.24	3e-4
-0.20	4.06	-7e-4	4e-4	17.71	8e-4

There are several important results. First, estimate errors in Y and $\dot{X}$ do not significantly influence the final state. In particular, an error in Y estimates will simply shift the estimated trajectory by the error, without changing the dynamics estimated. This is very beneficial, as the largest errors in estimation will be in the in-track (Y) direction for the stated mission profile. On the other hand, X and $\dot{Y}$ errors significantly change the motion of the spacecraft. An estimate error in the relative radial state (X) of the spacecraft will cause significant control issues if not corrected; this is because a nonzero x_d causes the chaser to drift away from the target. The large final position errors are largely due to this unchecked drift. If the state can be updated quickly before the chaser drifts too far, the effects of the estimation error are minimized. The 100m meter desired move and 3 orbit propagation period are far too long.

B. Closed Loop Simulation

Next, the components are tested when connected with each other to evaluate to propagation of errors and uncertainty throughout the process. In order to provide closed-loop simulation, an image generator was created to simulate images of the target spacecraft at desired range and spherical coordinates. An example image can be seen in Figure 5 with the background of Earth's horizon.

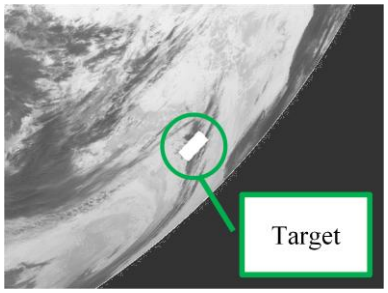

Figure 5. Simulated Image of Target

Using the image generator and an orbit propagator, successive steps in the AutoNav sequence were added and tested together. The specifications and performance of the chaser spacecraft must be defined for the simulation; these parameters can be seen in Table 5.

Table 5. Spacecraft Parameters

Parameter	Value
Spacecraft Wet Mass	50 kg
Thrusters	3 (+X,-X,+Y)
Maximum Thrust (per thruster)	4.4 mN
Specific Impulse	90s
ADCS	3-Axis Stabilized
Imager Array Size (pixels)	640 x 480
Pixel Pitch	25 μm
Focal Length	0.1 m
Imager FOV	9.1° x 6.8°

Orbital Simulation and Relative Navigation

The first step in closed loop testing was to test the batch filter's efficacy using measurement data from the IPAs given generated images. In this process a set of ROEs were generated for the simulated orbit, and the chaser's relative motion propagated forward for one orbital period. At discrete times in the orbit images of the target spacecraft were generated at desired range and spherical coordinates. These images were processed by the IPAs and the measurement estimates were given to the batch filter for state estimation. Figure 6 shows the range estimates from a simulation, along with the true values and a 16% uncertainty band; Figure 7 shows the estimated and real trajectories of the chaser, as well as the individual measurement estimates.

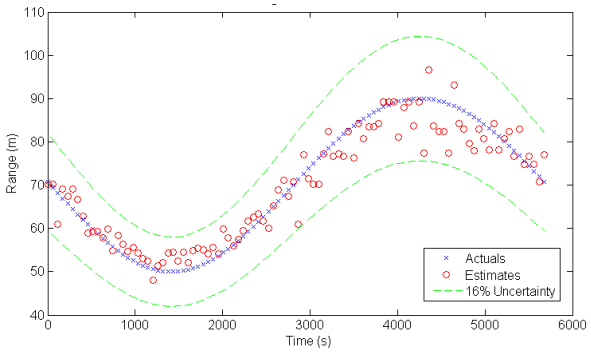

Figure 6. Range Estimates from IPAs for Simulated Orbit

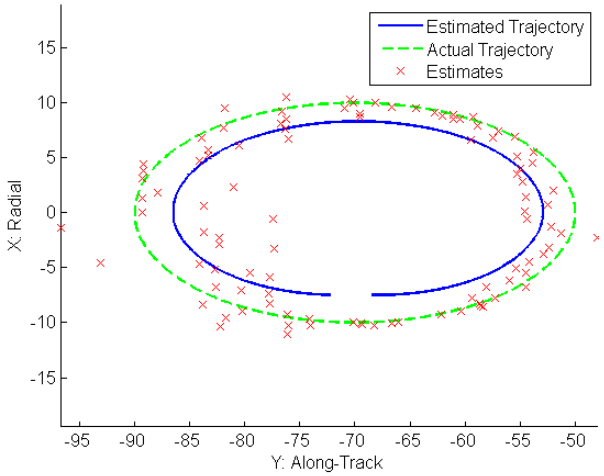

Figure 7. Estimated Relative Trajectory of Chaser

Similar to the open loop orbit determination filter tests, a series of test cases were examined to determined filter performance. The ROE estimates and the range errors are shown in Table 6.

Table 6. Relative OD Results with IPAs

Test Case	a_e (m)		x_d (m)		y_d(m)		Mean Range Error (m)
	Act	Est.	Act	Est	Act	Est	
Trailing 50m	0	0.14	0	0.07	-50	-46.2	3.30
Trailing 125m	0	3.03	0	-0.68	-100	-107.4	20.15
Static Ellipse	20	15.56	0	-0.18	-75	-66.3	9.60
Drifting Ellipse	20	21.74	2	2.60	-55	-59.6	7.47

The orbit determination filter shows a good ability to determine the relevant ROEs after an orbit of

imaging, especially for the cases when the target is within 100m.

LQR Control with Relative Navigation

The next step in the simulation process is to control the spacecraft using state estimates provided by the relative navigation. A block diagram for the simulation is shown in Figure 8. An initial orbit determination is used to provide an initial state estimate for the control. The waypoint logic is used to define the intermediate goals for the LQR control along the way to the ultimate goal, as defined by the mission profile. LQR control is used for one half period to move to the desired waypoint. At this time the spacecraft stops maneuvering and observes the target for one orbital period to update the state estimate. The new state estimate is given to the waypoint decision logic, and the process is repeated.

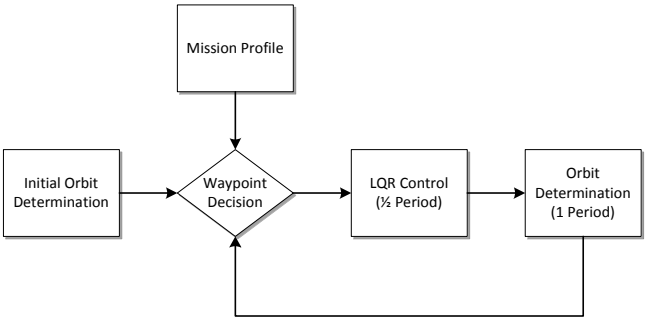

Figure 8. Simulation Block Diagram

For the purposes of testing, the chaser spacecraft was given an initial position of 110 meters behind the target and a final goal of 60 meters behind the target. The simulation was run for 15 waypoints, giving the chaser ample time to close to the goal and maintain its position. The results from one simulation can be seen in Figure 9 and Figure 10.

There are several important aspects of the motion of the spacecraft to note. First, there is an error in the initial state estimate provided by the relative navigation filter before LQR control begins; this error is approximately 12m in the in-track direction. This error, and all other estimation errors throughout the simulation, contributes to the difference in the actual position of the spacecraft and estimated position of the spacecraft. At regular time intervals, 1.5 orbital periods in this case, the estimated state is updated and can be seen by the discontinuities in both figures for the estimated state. These state estimate updates usually bring the estimate closer to the actual position, but this is not always the case, due to the uncertainty bands in the OD filter. The green trajectories in Figure 9 represent the periods of orbit determination where the spacecraft is not maneuvering; thus the end of these trajectories are where the discontinuities occur.

Most importantly, it can be seen that the spacecraft moves to a location close to the desired end goal, and a position well within the success criteria for the proposed mission. The final position of the spacecraft is 7m away from the desired goal. Of some concern is the trajectory movement within 50m, the designated "obstacle avoidance" zone. This is largely due to the filter bias at close ranges: the filter shows a bias to over-estimate the range to the target spacecraft at 50m; however, this bias may actually be related to the image generator, not to the filter itself.

The simulation was run 20 times to examine average performance, accounting for variability in target spacecraft orientation which changes the uncertainty of estimates. The resulting statistics can be seen in Table 7. Overall the simulation shows that the LQR controller with the relative navigation filter provides a very successful and fuel-efficient solution for automated proximity operations. The controller closed to within 7.48m on average after 15 waypoints, and during navigation averaged a range estimation error of 9.17m. Finally, an average of only 0.01 kg of propellant was used; this will be useful in extended life missions or low mass missions with small propellant budgets.

Table 7. Mean Statistics for Simulations

Parameter	Value
Error in Final Position (m)	7.479
Mean Error in Position Estimate (m)	9.165
Max Error in Position Estimate (m)	18.49
Propellant Used (kg)	0.0109

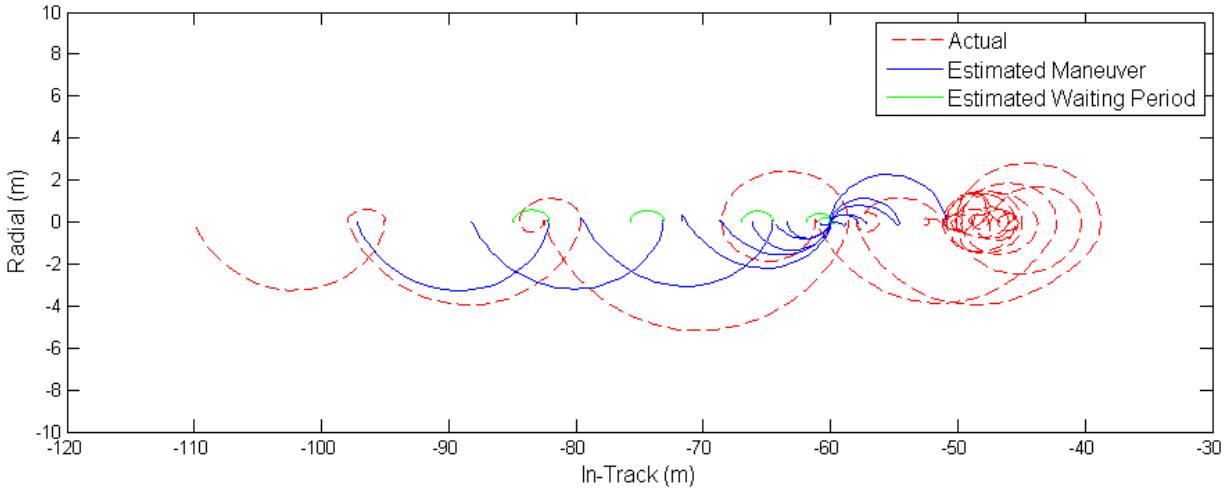

Figure 9. Chaser Trajectory

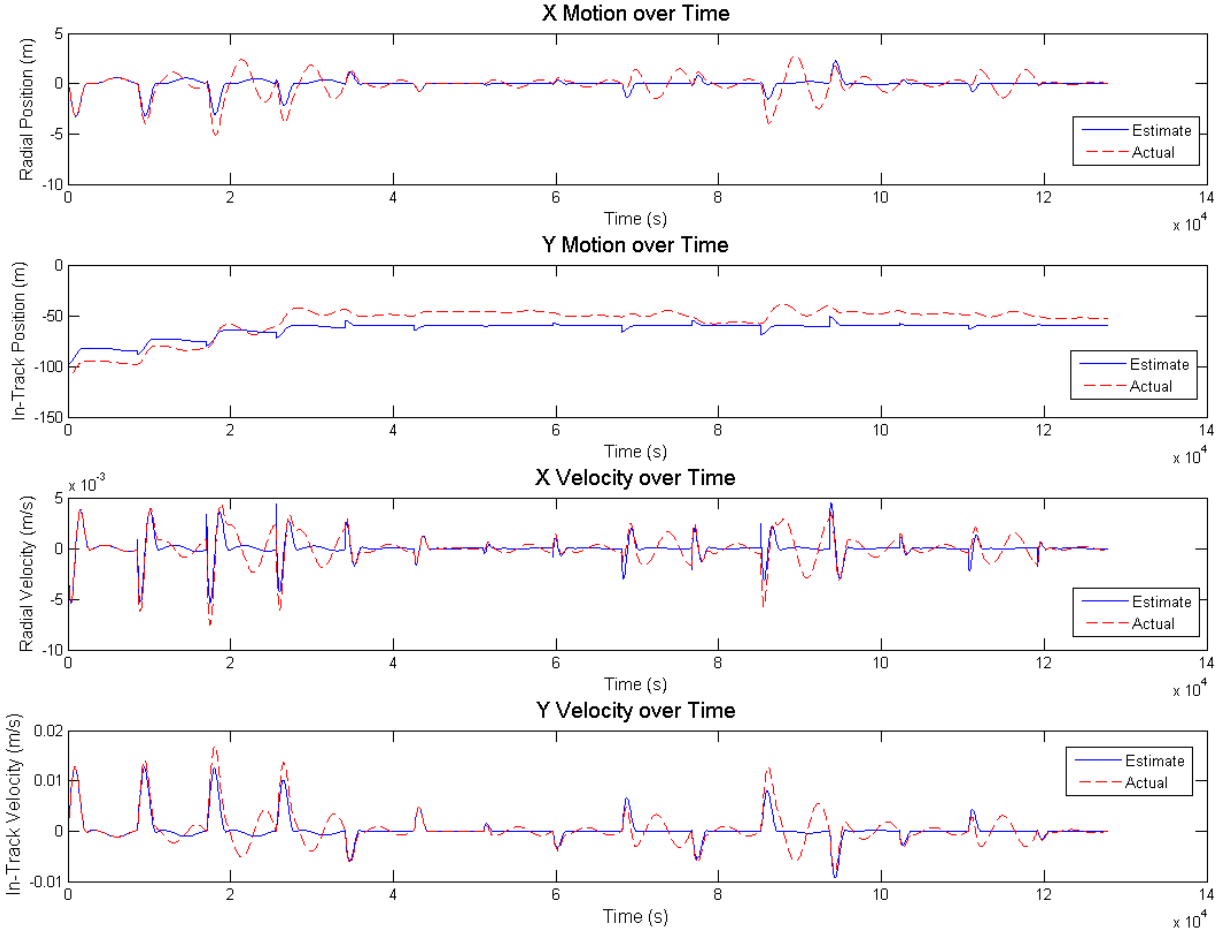

Figure 10. Chaser Motion versus Time

IV. Future Work

There are several key areas that would enable flight-qualified capabilities by reducing operational risk and increasing the fidelity of the estimates or would enhance the value of such a mission. First, a continuous Extended Kalman Filter provides an alternative to the batch filter which would be better able to estimate the uncertainty of an estimate and provide continuous updates and operations. Second, a complete circumnavigation of the target spacecraft using an NMC would present a more ideal mission profile for inspection and spacecraft characterization. It is possible to utilize Artificial Potential Functions (APF) in addition to LQR control to ensure collision avoidance, as seen in [19]-[20]. Third, a number of operational procedures must be defined to implement the AutoNav system for an operational mission related to the loss of sight of the target spacecraft.

Additionally, there are several layers of testing that should be added. For high-fidelity testing, a non-linear force model should be used to compare the simplified propagation of the CW equations to an actual state. Finally, uncertainties and errors from other subsystems, including the ADCS and propulsion subsystem, that will affect the performance of the AutoNav system will be included in the simulation to better understand system performance.

V. Conclusion

The Auto-Navigation System presents a viable solution for automated proximity operations about an uncooperative spacecraft using passive imagers and continuous thrust for small spacecraft. The use of image processing algorithms to estimate range, in addition to spherical angles, provides a valid solution for relative orbit determination necessary for GN&C. The use of an LQR controller for continuous-thrust maneuvering was shown to effectively control the spacecraft to a desired location. Closed loop simulations utilizing the IPAs and LQR control demonstrated the ability to successfully maneuver to a desired location and stay within mission profile bounds. Future work will develop the operational capabilities of the AutoNav system and increase the fidelity of testing modules to improve the accuracy of simulation and tests.

Acknowledgements

The author would like to thank his advisor, David Spencer, for his time and dedication. Additionally, he would like to thank Dr. Frank Chavez, Dr. Alan Lovell, and Dr. Josue Muñoz from AFRL for serving as technical advisors for the project. Finally, he would like to thank the AutoNav team members.

References

[1] NASA Marshall Space Flight Center, "DART Demonstrator to Test Future Autonomous Rendezvous Technologies in Orbit." *NASA Facts*. September 2004.

[2] Howard, R.T. and Bryan, T.C. "DART AVGS Performance." *NASA Marshall Space Flight Center Technical Report*. April 9, 2007.

[3] Davis, T.M. and Melanson, D. "XSS-10 Micro-Satellite Flight Demonstration Program Results." *Proceedings of SPIE Conference on Spacecraft Platforms and Infrastructure*, SPIE Vol. 5419.

[4] Dornheim, M.A. "Rendezvous Trials." *Aviation Week & Space Technology*. Vol. 162 Issue 16, pp. 35-36. April 18, 2005.

[5] Allen, A.C.M., Langley, C., *et al.* "Rendezvous Lidar Sensor System for Terminal Rendezvous, Capture, and Berthing to the International Space Station." *Proceedings of SPIE Conference on Sensors and Systems for Space Applications II*, SPIE Vol. 6958.

[6] Coppinger, Rob. "USAF Releases Images from Orbiting XSS-11." *Flight International*. Nov 15-Nov 21, 2005, p. 41.

[7] Weismuller, T. and Leinz, M. "GN&C Technology Demonstrated by the Orbital Express Autonomous Rendezvous and Capture Sensor System." *29th Annual AAS Guidance and Control Conference*. February 4-8, 2006.

[8] Malik, Tariq. "Prototype Satellites Demonstrate In-Orbit Refueling." *Space.com.* April 4, 2007.

[9] Clark, Stephen. "Satellite In-Space Servicing Demo Mission a Success." *Spaceflight Now*. July 23, 2007.

[10] Kelso, T.S. "Supplemental Two-Line Element Sets." *Today from the Center for Space Standards & Innovation.* December 16, 2000.

[11] Patel, H., Lovell, T.A., *et al.* "Relative Navigation for Satellites in Close Proximity Using Angles-Only Observations." *AAS/AIAA Space Flight Mechanics Meeting*, January 29 – February 2, 2012. AAS 12-202.

[12] Bellet, Edouard. "Detection and Localization of a Target on a Thermal Image Using a 'Blobber' Algorithm." *Georgia Tech Center for Space Systems*, 2011.

[13] Hill, G.W. "Researches in the Lunar Theory." *American Journal of Mathematics*, Volume 1, 1878, pp. 5-26.

[14] Clohessy, W.H. and R.S. Wiltshire. "Terminal Guidance System for Satellite Rendezvous", *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653-658.

[15] Vallado, D. A. *Fundamentals of Astrodynamics and Applications.* Hawthorne, CA: Microcosm Press. 2007.

[16] Lovell, T. A., and Tragesser, S. G., "Guidance for Relative Motion of Low Earth Orbit Spacecraft Based on Relative Orbit Elements," AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Province, RI, 2004.

[17] Tapley, B.D., Schutz, B.E., Born, G.H. *Statistical Orbit Determination*. Burlington, MA. Elsevier Inc., 2004.

[18] Bryson, A.E., Jr., and Ho, Y.-C., *Applied Optimal Control*. Washington DC: Hemisphere. 1975.

[19] Bevilacqua, R., Lehmann, M., Romano, M., "Development and experimentation of LQR/APF guidance and control for autonomous proximity maneuvers of multiple spacecraft." *Acta Astronautica*, Vol. 68. 2011.

[20] McCamish, S.B., Romano, M., Yun, X., "Autonomous Distributed Control of Simultaneous Multiple Spacecraft Proximity Maneuvers." *IEEE Transactions on Automation Science and Engineering*, Vol. 7, No. 3. July 2010.

[21] Hablani, H.R., Tapper, M.L., and Dana-Bashian, D.J. "Guidance and Relative Navigation for Autonomous Rendezvous in a Circular Orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 3, May-June 2002.