FLASH LIDAR based Relative Navigation

Jack Brazzel
NASA Johnson Space Center
2101 NASA Parkway
Houston, TX 77058
jack.p.brazzel @nasa.gov

Fred Clark
Draper Laboratory
17629 El Camino Real
Houston, TX 77058
clark@draper.com

Zoran Milenkovic
Draper Laboratory
17629 El Camino Real
Houston, TX 77058
zmilenkovic@draper.com

***Abstract*—Relative navigation remains the most challenging part of spacecraft rendezvous and docking. In recent years, flash LIDARs, have been increasingly selected as the go-to sensors for proximity operations and docking. Flash LIDARS are generally lighter and require less power that scanning Lidars. Flash LIDARs do not have moving parts, and they are capable of tracking multiple targets as well as generating a 3D map of a given target. However, there are some significant drawbacks of Flash Lidars that must be resolved if their use is to be of long-term significance. Overcoming the challenges of Flash LIDARs for navigation—namely, low technology readiness level, lack of historical performance data, target identification, existence of false positives, and performance of vision processing algorithms as intermediaries between the raw sensor data and the Kalman filter—requires a world-class testing facility, such as the Lockheed Martin Space Operations Simulation Center (SOSC). Ground-based testing is a critical step for maturing the next-generation flash LIDAR-based spacecraft relative navigation. This paper will focus on the tests of an integrated relative navigation system conducted at the SOSC in January 2014. The intent of the tests was to characterize and then improve the performance of relative navigation, while addressing many of the flash LIDAR challenges mentioned above. A section on navigation performance and future recommendation completes the discussion.**

Table of Contents

1. INTRODUCTION

Development of relative navigation systems with Flash Lidars was the prime motivator for the project that yielded the results and conclusions discussed herein. The work performed is an extension of the Orion/MPCV program's efforts to ensure that capabilities for Rendezvous, Proximity Operations, and Docking (RPOD) are ready for future use. As such, much of the discussion will be focused on a generic Orion/MPCV mission to the International Space Station (ISS). There have been many successful spacecraft-to-ISS docking, therefore using the ISS as the development platform removes many unknowns on the target side of relative navigation. The Orion relative navigation architecture for integrated testing was documented in detail in reference [10].

Building a relative navigation system begins with hardware selection, closely followed by hardware characterization. The discussion in this paper begins with the basics of lidar hardware, followed by methods for characterizing the performance of that hardware. Once sensors are characterized, various software algorithms must be evaluated. We discuss some of the specialized algorithms that have been incorporated in testing. Finally, the integrated product, one with hardware, specialized algorithms, and some form of a Kalman filter all come together to make the relative navigation system work.

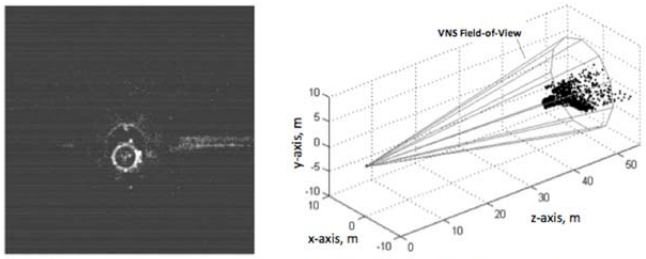
2. LIDAR

With the increasing availability of Lidars for space rendezvous they have already become the de-facto standard in relative navigation for proximity operations. The Space Shuttle, for example, executed over 60 successful dockings to the ISS using a Lidar. Similarly European Space Agency's (ESA) Automated Transfer Vehicle (ATV) used a lidar for five automated dockings to the ISS. To date, Orbital's Cygnus, and SpaceX's Dragon resupply vehicles have also used Lidar for proximity operations sensors. Lidar is even going to be employed at an upcoming OSIRIS-Rex asteroid rendezvous mission. The question then is not "if lidar" but "which lidar"? But first, let's look at "what is lidar"?

A lidar is a laser device consisting of an emitter and a detector. The emitter sends out a pulse of light, and the detector collects the return signal. Each return comes from a direction, representing angular measurements with respect to the detector's boresight. Lidar devices also contain timing circuitry that measures the time from emittance to detection. Multiplying the time by half the speed of light yields a range measurement.

978-1-4577-0557-1/12/$26.00 ©2014 IEEE

A scanning lidar typically uses a very narrow laser beam, the direction of which is controlled by gimbaling mirrors, in combination with a small detector, such as an avalanche photo diode (APD). The device scans the field of view (FOV) by moving the mirrors. Typically, the scanner is looking for a strong return to lock onto. Once locked, it continues to track the return until the signal becomes too weak or moves out of the FOV. A flash lidar, by contrast, has a diffuser that spreads the outgoing beam, coupled with a large detector array. The strength of the outgoing beam, the sensitivity and size of the detector array determine the performance boundaries of the lidar. The flash lidar does not acquire lock, rather it operates akin to a camera by taking a "laser picture" of the FOV. The output image is an array of intensities and ranges, whereby each cell contains an intensity value and a range value. This information can be used to compute not only relative position between two spacecraft, but relative attitude as well. This kind of computation is called pose. Figure 1 shows the prototypical lidar measurements of the ISS from a approaching lidar. The 1a) represents the intensity image, while 1b) shows the range mapped into a 3-D point cloud.



a) Intensity image. *b) Range image converted to 3D point cloud.*

Figure 1: Lidar intensity image and range point cloud of the ISS.

A discussion on the basics of lidars deserves a note about targets. For the rendezvous in space problem "the target" is a term most frequently used to describe the spacecraft that is not executing any translational maneuvers, while "the chaser" is the actively translating vehicle. For lidars a target is anything that is capable of reflecting laser signals back into the detector. A uncooperative target does not have any known or specially designated reflective features. A well-known example of uncooperative targets is the Google-car's lidar-based relative navigation system, which must contend with pedestrians, other vehicles, infrastructure, plants, animals etcetera, none of which have features built specifically to support lidar returns.

Fortunately, for the ISS rendezvous problem, the target is a cooperative one with carefully positioned reflectors to enable visibility for incoming vehicles. Unique geometry of reflector placement enables matching centroids to known reflector locations.

3. LIDAR TESTING

Space-going lidar must withstand extreme vibration and environmental conditions, often without active thermal control, with days, months, or even years between uses. Lidar performance must be well understood prior to launch, and it must show consistent performance across a variety of conditions. The performance and reliability of sensors is the number one risk for RPOD. A solid testing plan must be implemented to better characterize sensor behavior. The testing should, of course, include ground testing. Preferably, testing in the space environment is desirable as well. The Neptec TriDAR, Advanced Scientific Concepts (ASC) DragonEye, and Ball VNS were all tested on the Space Shuttle prior to their deployment. Repeated flight tests are the ultimate risk mitigation techniques. However, with the unfortunate retirement of the Space Shuttle, not only are tests prohibitively expensive, the availability of test flights has all but disappeared.

An acceptable starting point for ground testing is outdoor testing. The advantage of outdoor testing is lower cost and very large range. Test can be conducted by mounting the sensors on a static or moving platform such as a car or a helicopter. Preferably, the target is fixed. One major disadvantage of outdoor testing is the impact of unknown disturbances in terms of atmospheric conditions, vibrations. Another major disadvantage is generating truth data; i.e. the true relative position and relative attitude between the sensor and the target. Therefore, the basis for data analysis is unclear.

A better option for ground testing is a controlled environment in the form of a first-class indoor facility. In lieu of space-based testing, ground-tests in a controlled environment can build a good picture of the expected sensor.

4. TESTING FACILITY

The Space Operations Simulation Center (SOSC) at the Lockheed Martin Corporation (LMCO) in Colorado is a dynamic test environment focused on Autonomous Rendezvous and Docking (AR&D) development testing and risk reduction activities. The SOSC supports multiple program pursuits and accommodates testing GN&C algorithms, hardware testing and characterization, as well as software and test process development.

The main manipulator of sensors at the SOSC is the rail-mounted 6DOF robot that has a range of linear motion of 60 meters longitudinally, and 15.2 meters in the lateral and vertical directions each. The robot can translate, and rotate about all three axes via commands from the robot control station. The commands to the robot mechanism can come from a variety of ways such as a pre-recorded file, real-time console commands or a simulation. Cameras, lidars and any other sensors are mounted on the robot, thus providing the capability to test the sensors while in motion.

For Orion-to-ISS RPOD testing, the facility supports a full-size ISS mock-up to aid in simulating real-world rendezvous and docking scenarios. The ISS mock-up is a static target that has a fixed location within the SOSC high bay. Figure 2 shows a photograph of the ISS mock-up with the robot in front of it [1]. The photograph was taken from ground-level perspective. The key to the mock-up is a full-size docking target, complete with cooperative targets, allowing lidars to effectively track the docking location. A detailed description of SOSC capabilities was documented by Ahlbrandt [2].

Even though this facility provides unique testing opportunities limitations still exist. The limits of the robotic motion are the primary restrictions. Of course, the robot cannot move outside the confines of the building, it cannot pitch over 360 degrees, and it has maximum acceleration limits in all three axes. Still, very valuable results have been obtained from the SOSC testing, some of which are discussed in later chapters.

Figure 2: Full-size ISS mock-up (left) with sensor enclosure (center) mounted on the 6DOF robot (right), as seen from a ground perspective [1].

5. TECHNIQUES FOR LIDAR DATA ANALYSIS

While there are numerous techniques for analyzing sensor data, the discussion in this section will focus on a few that have been used recently.

5.1 LOESS

The loess method, developed by Cleveland [12] is a nonparametric method that obtains a smoothed curve by fitting successive linear regressions in local neighborhoods. The method is similar to a moving average or running median methods in that it uses a neighborhood around each *X* value to obtain a smoothed *Y* value corresponding to that *X* value [13].

5.2 DOUBLE DIFFERENCING

A standard technique for the "detrending" of time series is to compute differences among consecutive points [3].

Let z_k denote one of the four types of measurements at time step *k*. The first difference is defined as

$$\Delta z = z_k - z_{k-1}$$

The second difference is defined as

$$\Delta^{(2)} z = \Delta(\Delta z) = (z_k - z_{k-1}) - (z_{k-1} - z_{k-2}) = z_k - 2z_{k-1} + z_{k-2}$$

To estimate the standard deviation of the noise of the lidar measurements, we must simply calculate the standard deviation of the second differences of the measurements and divide by $\sqrt{6}$.

Let us clearly delineate the assumptions inherent is this method. We assume the following:

1. The data are equally spaced in time.
2. Consecutive measurement errors are uncorrelated.
3. The data are stationary.

Double differencing is an extremely useful technique appropriate for many occasions. For example, if another source of data is not available for comparison, the double-differencing method will provide a good estimate of the measurement variance.

5.3 BEST ESTIMATED TRAJECTORY

A best estimated trajectory (B.E.T) is a very useful basis for analysis and comparison of sensor data. A B.E.T is a product of optimally smoothing data from a Kalman filter. It is best to filter data from a separate and already characterized sensor, compute a B.E.T, and compare the answers with the sensor in question. For lidar test flights on the Space Shuttle, external data was available via the Trajectory Control Sensor (TCS is a scanning lidar).

The B.E.T. development process for as applied to a Space Shuttle mission is outlined in figure 1. In this figure, $\mathbf{\Phi}$ denotes the state transition matrix, $\mathbf{P}$ denotes the state covariance matrix, and $\mathbf{x}$ denotes the 6 x 1 state vector. The equations of the smoother are given below.

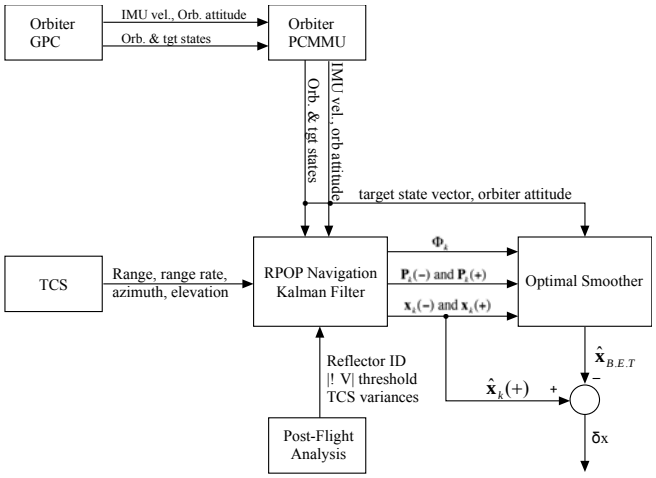


Figure 3: B.E.T. development process

Let the final step be denoted by the subscript *N*. The smoothing algorithm begins at the last state and moves backward in time. The smoothed state vector at step *k* is estimated from

$$\hat{\mathbf{x}}_{k|N} = \hat{\mathbf{x}}_k(+) + \mathbf{A}_k[\hat{\mathbf{x}}_{k+1|N} - \hat{\mathbf{x}}_{k+1}(-)]$$

$$\text{where } \mathbf{A}_k = \mathbf{P}_k(+)\Phi_k^T\mathbf{P}_{k+1}^{-1}(-)$$
$$\text{and } \hat{\mathbf{x}}_{N|N} = \hat{\mathbf{x}}_N(+) \text{ for } k = N-1$$

The smoothed covariance matrix is computed from

$$\mathbf{P}_{k|N} = \mathbf{P}_k(+) + \mathbf{A}_k[\mathbf{P}_{k+1|N} - \mathbf{P}_{k+1}(-)]\mathbf{A}_k^T$$

$$\text{where } \mathbf{P}_{N|N} = \mathbf{P}_N(+) \text{ for } k = N-1$$

Once the B.E.T is derived it can be used for comparing raw sensor data to derived B.E.T. measurements. The derived B.E.T. measurement is computed via

$$\boldsymbol{z_k} = h(\hat{\boldsymbol{x}}_{\boldsymbol{k}})$$

This kind of comparison can be done with indoor or outdoor tests, so long as a Kalman filter is executed while collecting sensor measurements. The disadvantages of this method are the need for significantly more software in the loop. The B.E.T can be constructed from self-data; i.e. filtered data from a sensor can be used to create a B.E.T. to then assess the noise of the sensor. However, it is better to construct the B.E.T. using data from a different sensor if at all possible.

5.4 SIMULATION TRUTH DATA

If a 6DoF simulation is included and used as truth in the testing, then the approach to analysis is to simply derive a measurement from the simulation truth and directly compare. The bulk of the analysis at the SOSC was done using this approach. The main disadvantage of this approach is that a simulation to control the testing must be built. Furthermore, the simulation must be configured correctly to match both the spacecraft and the facility parameters.

6. NAVIGATION ALGORITHMS

Flowing down flash lidar data to the navigation software is more involved than passing scanning lidar data. For a scanner, the measurement is turned from hardware units to engineering units at the interface, resulting in anglular and range measurements that are (usually) fed to an extended Kalman Filter (EKF). Flash lidar output does not contain any measurements. Instead, the array of return intensities and ranges is passed into image processing algorithms. The image processing algorithm's role is to identify centroids.

The goal of the centroiding algorithm is to process the raw image from the lidar and determine whether strong returns in the pixel map represent noise or actual ISS reflectors. By identifying the reflectors in the pixel map, the information can further be processed to generate pose measurements for processing by the Kalman filter. In the testing facility, the centroiding algorithm ran on a stand-alone Linux machine. The inputs to the program were Lidar images, while outputs were centroid locations. It should be noted that the centroiding algorithm used for this test was developed by NASA and is documented by Christian [1].

If three or more centroids can be identified consistently, then pose solution can be computed based on an algorithm by Haralick [9]. This method was modified by Christian to meet particular needs of our system [5]. The pose solution outputs two components: 1) a relative position from the sensor to the center of the target (basis of pose frame), and 2) a relative attitude quaternion between the sensor frame and the pose frame.

6.1 REFLECTOR IDENTIFICATION IN THE PRESENCE OF SPARSE DATA

All questions have not been answered however. Prior to sending an angle and range measurement, an algorithm must be employed to match centroids to reflectors. While consistently centroiding three locations can produce a high-confidence pose solution, intermittent three centroids degrade the confidence in the solution. It follows to ask questions about how we identify which reflector is tracked if only 1 or 2 centroids are identified. To address this issue, a special reflector identification algorithm was designed. The ideal reflector identification algorithm will seek to have the following characteristics:

1. Be able to identify at least one reflector in the field of view with high reliability
2. Maintain independence from the navigation filter
3. Have the capability of operating with arbitrary reflector placement on the target
4. Be independent of the proximity operations profile
5. Have acceptable CPU requirements

6. Be able to identify without long runs of data
7. Be able to accommodate a reflector going out of the field of view
8. Be able to accommodate a reflector visibility model
9. Be able to tolerate small errors in knowledge of the target attitude

If the reflector identification does its job correctly, then one or more measurements will be available for processing by the Kalman filter. The measurements consist of range, vertical and horizontal angles to identified reflectors. A pair-matching method based on Mahalanobis distance was selected for reflector identification. The definition for the Mahalanobis distance for this particular problem is:

$$D_{ij} = \sqrt{(\delta\hat{\mathbf{r}}^{(meas)}_{tgt\ cg\ to\ refl\ i-j} - \delta\mathbf{r}^{(known)}_{tgt\ cg\ to\ refl\ i-j})^T \Sigma^{-1} (\delta\hat{\mathbf{r}}^{(meas)}_{tgt\ cg\ to\ refl\ i-j} - \delta\mathbf{r}^{(known)}_{tgt\ cg\ to\ refl\ i-j})}$$

In this case, $\boldsymbol{\Sigma}$ denotes the covariance matrix of the measured difference vector $\delta\hat{\mathbf{r}}^{(meas)}_{tgt\ cg\ to\ refl\ i-j}$. The measured difference vector is the difference between known locations of the reflectors on the ISS and estimated locations based on the latest measurements. Selecting the smallest D_{ij} will result in the minimal predicted error and the best-suited pair for processing by the navigational filter.

6.2 NAVIGATION FILTERING

The primary relative navigation extended Kalman filter estimates the inertial states of the chaser and target. The states include position, velocity, and attitude of the body with respect to the Earth-centered inertial frame. The states are augmented with a range bias and misalignment. The filter is updated with the lidar measurements in terms of range and angle measurements. A separate multiplicative extended Kalman filter (MEKF), was designed and implemented in order to process the relative attitude quaternion coming from pose and produce the best estimated relative attitude. The relative attitude filter is not the focus of this paper. Note that both filters utilize *simulated* IMU measurements in this loop.

7. COMPONENT INTEGRATION

Thus far, we have discussed the general principals behind flash lidar as applied to spacecraft rendezvous. We've mentioned how lidar works and collects data, testing options and analysis methods. We have also discussed components required to complete a relative navigation system utilizing a flash lidar. The list includes centroiding, reflector identification, pose determination, and state estimation via extended Kalman filtering.

The most effective testing is best done in a closed-loop manner. Closed-loop testing provides the opportunity to test the system with simulated dynamics while moving the sensor, allowing the Kalman filter to be tested as it would in flight. The figure 4 shows the components as connected in the test. From the top, the figure shows the lidar physically mounted on the robot. Lidar images are sent from the lidar to the relative navigation software. In particular, the centroiding is the first algorithm to grab the image. If three or more centroids can be identified with confidence, then computation of pose takes place. The second criterion to begin pose processing is being below maximum range. A maximum range is necessary to allow for sufficient geometric separation within the image. The resulting pose measurement is first converted into an equivalent range and angles measurement, then passed to the EKF to process and update the state. If three centroids are not identified with confidence, or the ranges are above the pose threshold, then ranges and angles to the centroids are passed to reflector identification. Reflector identification then pinpoints which reflectors are most likely to be matched with the centroid. These reflectors now place the return on the target, representing relative measurements that can be processed by the EKF. The outputs of the filter represent inertial states, in terms of position, velocity and attitude, of the chaser and the target. Notice that even though the interface to the EKF is equivalent for both pose or centroid measurements, the measurement weighting changes based on the source of the data.

However, there are still three components that are necessary to close the loop around the sensor and the filter.

First, control software must be active to effect thrusters. Controlling the dynamic trajectory of the robot is critical to maintaining reflectors in the FOV. Commanding is done via the prototype flight software control system. The control system issues thruster firing commands using a simplex-based optimization technique derived by Glandorf [8].

The thruster firing commands move from control software to the GN&C simulation. These thruster commands, together with forces from several other models, are integrated by the simulation,. The most prominent models are orbital and attitude dynamics of the two objects, gravity modeling, effectors, inertial measurement units (IMUs), and interfaces between hardware components. The simulation outputs the inertial position and attitudes of the two vehicles to the robot controller.

At the end of the loop, the robot controller converts the difference between inertial states to a new robot state in facility coordinates. The robot then moves and provides a dynamic platform for the sensor.

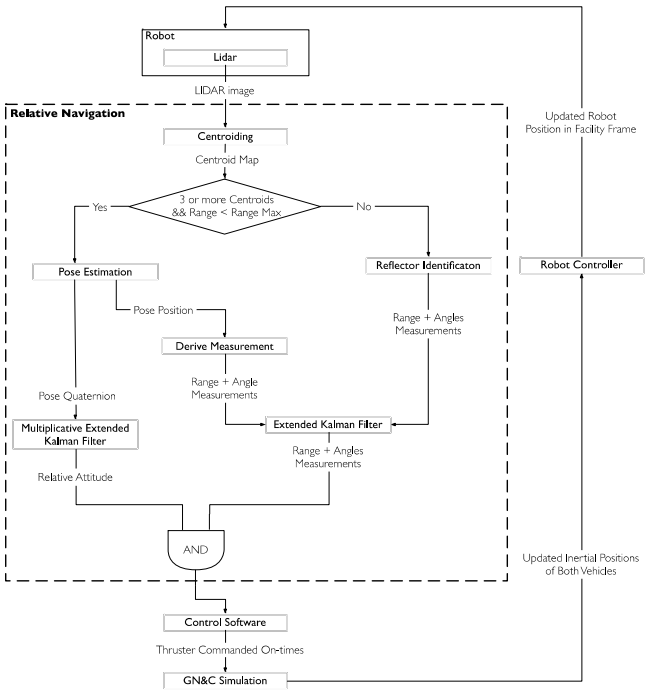


Figure 4: Integrated testing components and key data flow.

8. TEST RESULTS

Testing with the aforementioned setup has been done at four different cycles in the last three years. At each successive test cycle more capability is added and tested. Figure 4 shows the current test setup. Numerous test of both static and dynamic nature have been executed. It would do a disservice to the volumes of work done and data generated to attempt to put it in this document. Therefore, this section will focus on one of the latest sets of tests performed with the configuration presented.

8.1 TEST DESCRIPTION

The SOSC testing comprised of a set of 10 tests beginning at 55 meters sensor-to-target distance and stopping at 2 meters. The main objective of the test was to better-characterize the sensor and the relative navigation system performance. The test trajectory is given in reference 10. It begins with a glideslope approach at the initial conditions (dispersed around 55 meters). The glideslope continues until 15 m, where a station-keeping hold is executed. The station-keeping mode holds a particular location from the docking port of the target within control errors. After approximately 200 seconds of station-keeping, the final approach begins. The final approach part of the trajectory aims for a closure rate of 0.03 m/s, while controlling the attitude and lateral velocity within docking tolerances.

Figure 5 shows the details of the position during the test. The top plot shows each component of the simulation truth position (x, y, z) as given in the target docking port (TDP) frame. The TDP X direction begins at the desired docking location on the target and extends normally outward. The Z component is in the direction of the center of the Earth while the axis completes the right-hand coordinate system. Note that the system is fixed to the target structure and moves accordingly when the target attitude changes. The X component in the top line is shown is blue, as it steadily decreases for the first third of the trajectory. At station-keeping X remains constant (within control errors) for over 200 seconds. Final approach continues until 2 meters of relative range is reached.

The bottom plot in figure 5 shows the inplane view of the trajectory, in terms of X-Z TDP. This plot shows the simulation output state (Truth) and the estimated state from the EKF (Nav). For the rest of the data analysis we employed the comparison to simulation technique as described in section 5.3. The bottom plot shows notionally the kind of position errors that can be expected out of the test.

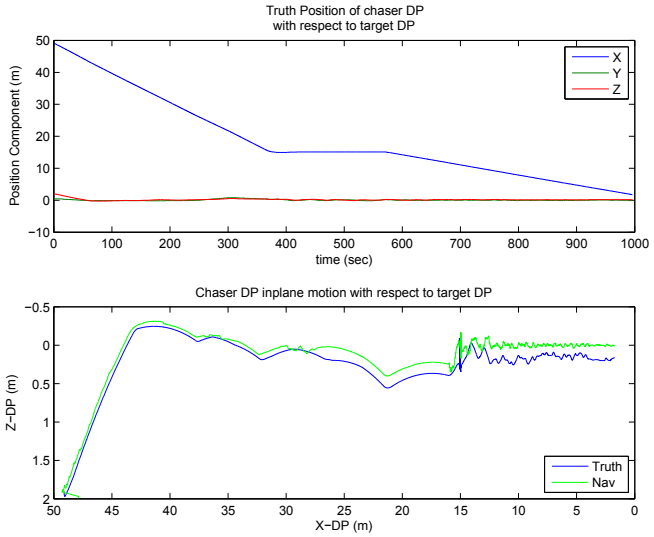


Figure 5: Trajectory plots of the chaser docking port position in the target docking port frame.

Measurements are constructed matching centroids to reflectors. The results of the centroiding algorithms are passed in as a list to the reflector identification algorithm. For the testing at hand, there could be up to 10 centroids passed into reflector identification. The purpose of reflector identification is to match these centroids to known reflectors on the target structure. Figure 6 shows the results of the reflector identification algorithms. The top plot represents which one of the available lidar centroids was matched to a reflector identified in the bottom plot. These measurements are fed to the EKF from the beginning of the test until the station keeping. Note that if centroid number and reflector identified are set to zero, then the entire centroid-to-identified reflector code is bypassed and a pose-derived measurement is used instead.

In figure 7 we plot the range, and angular measurements derived from pose. The pose-derived quaternion is not plotted. For pose-derived measurements to be available there have to be three or more visible and identifiable centroids in the lidar image. Therefore, at the time instances given in figure 7, there must be three or more identifiable centroids. Even though some pose measurements are available prior to the station-keeping phase, the navigation system does not incorporate pose measurements until station-keeping.

What measurements were actually available to the EKF for updating the state? Figure 8 shows the range, horizontal, and vertical bearing angles that were sent to the EKF for processing. The measurements are a mash-up of measurements constructed from reflector identification prior to station keeping, and pose-derived measurements once station keeping starts. As expected, the measurements vary with the motion of the spacecraft. These plots also show several measurements that are far away from any reflector on the target. We can safely conclude that these are false positives. The angle plots give an indication of the discrete nature of reflector identification as the measurements jump between discrete intervals as different reflectors are identified.

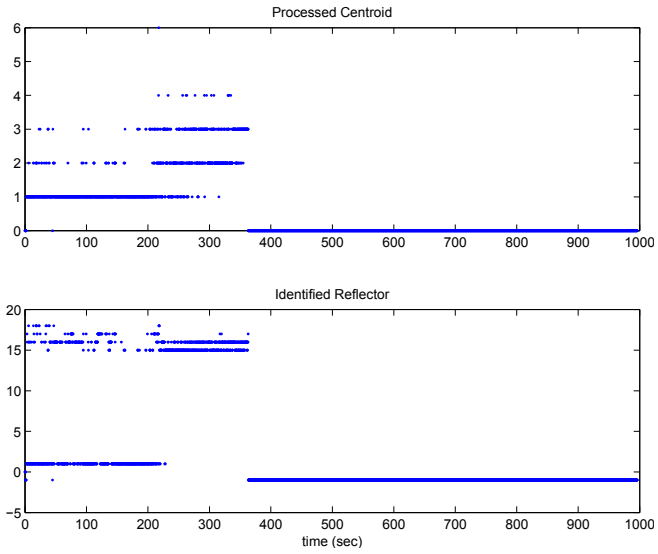


Figure 6: Reflector identification matching to centroids.

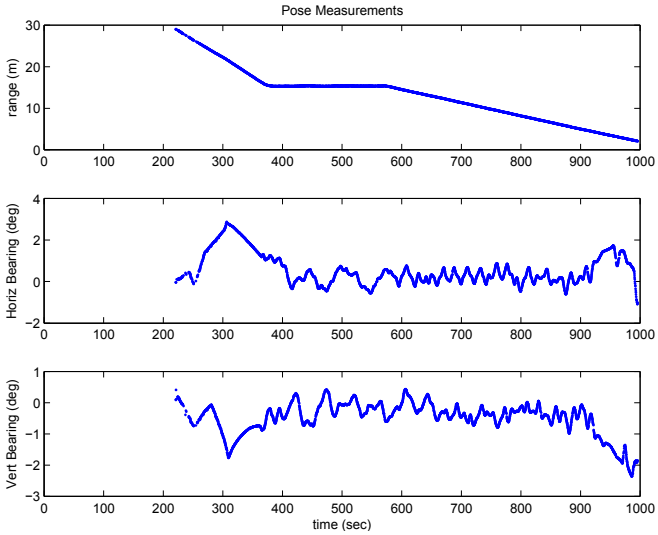


Figure 7: Pose-derived measurements as available when three or more centroids are available.

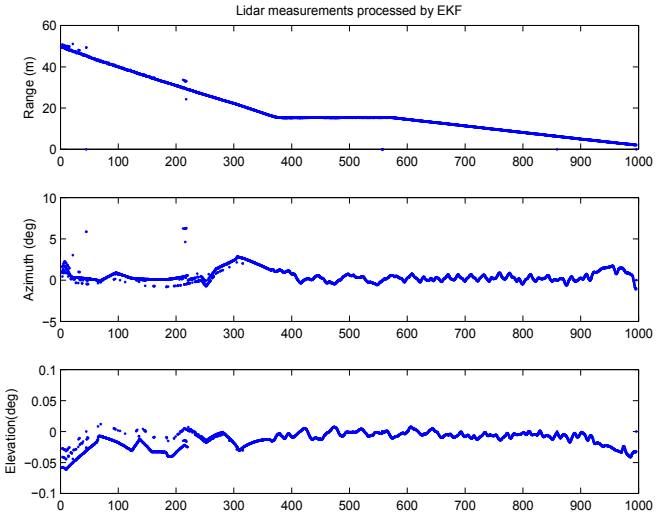


Figure 8: Measurements passed into the EKF prior to residual edit.

The available measurements are used by the EKF to update the state and the covariance matrix if and only if they pass a residual edit test. The residual edit consists of first constructing a measurement residual, followed by computing the residual ratio, and checking this ratio against a threshold. First, computing the residual

$$\delta \boldsymbol{z} = \boldsymbol{z} - \boldsymbol{h}(\widehat{\boldsymbol{x}})$$

where $\boldsymbol{z}$ is the measurement, and $\boldsymbol{h}(\widehat{\boldsymbol{x}})$ represents the linearized expected measurement from the current state estimate. Even though the residual is a vector, there are three components of the measurement processed individually. Figure 9 shows the residuals for each measurement. Figure 10 shows these residuals plotted in histogram format. As expected, the residuals center about zero, but false positive can be seen in the plots as well.

Additional data that can be gleaned from the residual plots is the patterns of potential misidentifications by the reflector identification algorithm. Well-behaved residuals will be randomly distributed about a mean of (close to) zero. However, a systematic error is present as the residuals are grouped in discernable lines in the angle plots of figure 9.

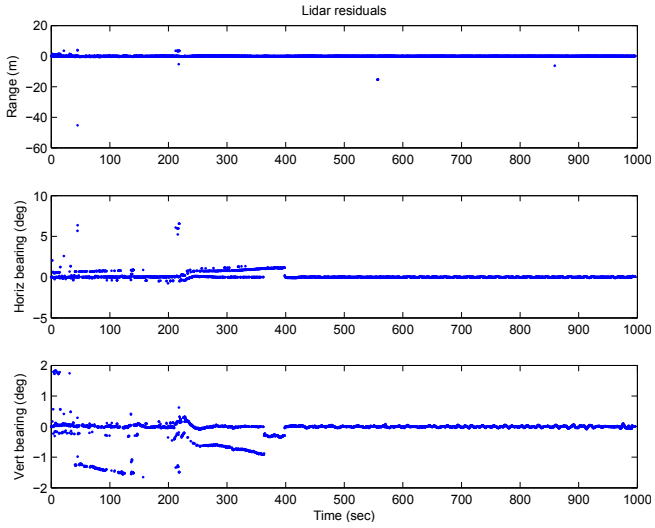


Figure 9: Residuals

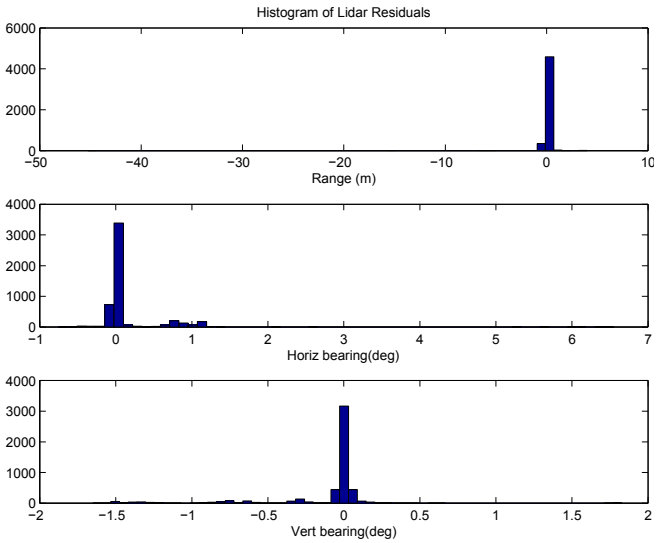


Figure 10: Histogram of Residuals

The residual ratio is computed by using the residual as follows:

$$ratio = abs(\frac{\delta \boldsymbol{z}}{\boldsymbol{HPH^T + R}})$$

Where **H** is the measurement partials matrix, **P** is the covariance matrix, and **R** is the measurement-weighting matrix. Now, if the ratio is greater than 3 the measurement is rejected. Any measurement with a ratio of less than 3 is processed by the EKF. Figure 11 shows the residual ratios for the measurements, while Figure 12 shows the number of accepts (blue) or rejects (red) based on the aforementioned criteria. From a performance standpoint, we expect only 7 out of 1000 measurements to be rejected. However, such is not the case here. There are too many rejections for this system.

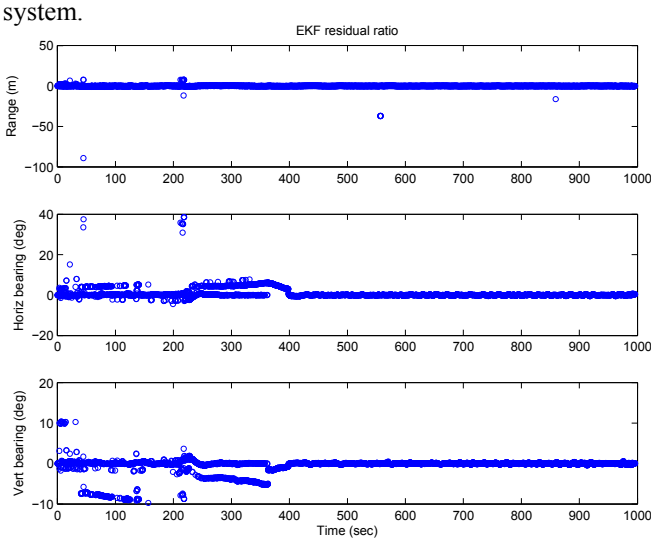


Figure 11: Residual Ratios

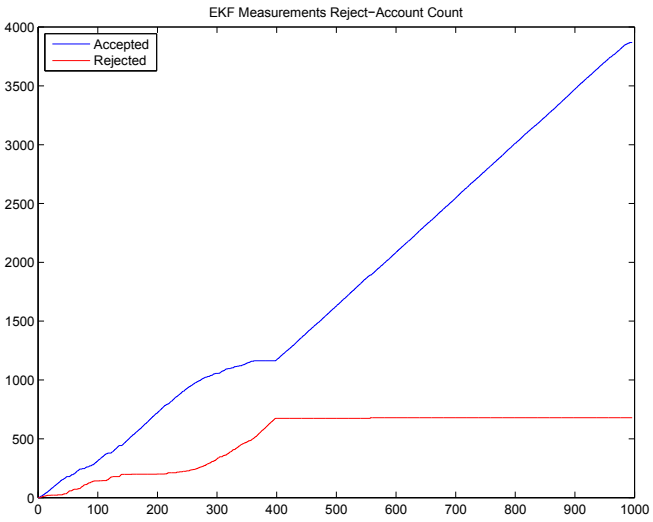


Figure 12: Accept-Reject Count

Finally, the accepted measurements are processed by the EKF in order to update the state estimate and the error covariance. Figures 13 and 14 present the position and velocity estimation errors for the test, co-plotted with the covariance bounds (1σ). Each figure has the three vector components plotted separately. Error components were computed by differencing the relative state from the GN&C simulation and the EKF state estimate. The plots underscore the need for improvement in the system. For example the X-position develops a 0.30 m bias as the run

progresses. The lateral error (Y) shows a bump in the middle of the run, and then settles at approximately 0.05 m error. Both of these components are out of the covariance lines, but with little scatter, indicating that the bias components dominate the error. Velocity-wise, the errors are well behaved with a slight bias (0.002 m/s) in the Z direction. Despite the large number of rejections, the navigation solution holds together in terms of error and covariance. This is welcome news, as it shows that the EKF is actually quite robust to rejections, false positives and mis-identified reflectors.

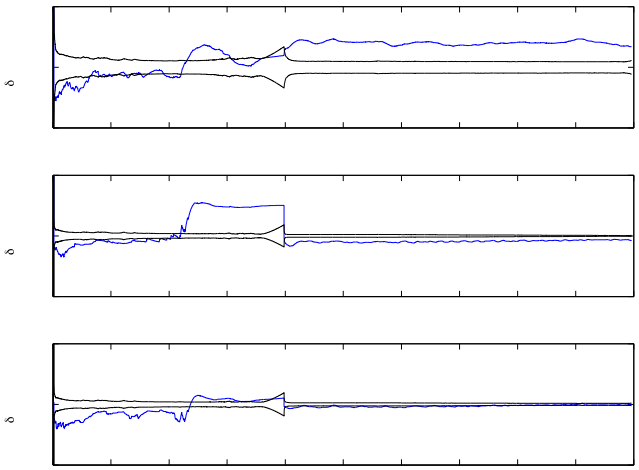


Figure 13: Relative Position Navigation Performance

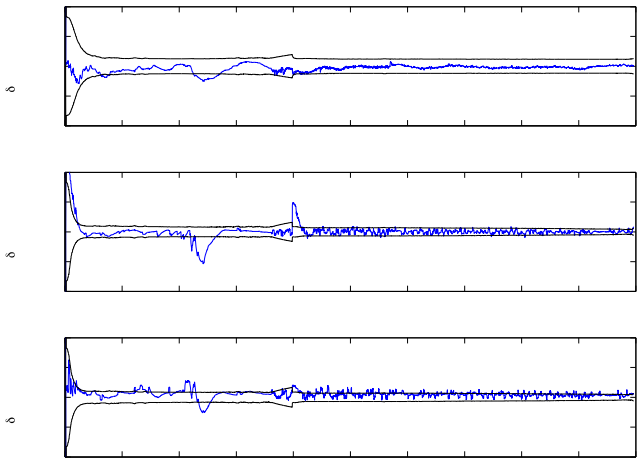


Figure 14: Relative Velocity Navigation Performance

The final set of plots relate to the estimation of the inertial attitude by the translational filter. Figure 15 shows errors in the three rotational axis as described by estimated Euler angles and associated 1-sigma error covariance bounds. The estimation error is well-behaved with respect to the covariance, but there is an obvious influence of the dynamics on the estimation. Tighter control deadbands take effect at station-keeping and continue through the end of the run. An increase in the error noise in figure 15 corresponds to the change in deadbands.

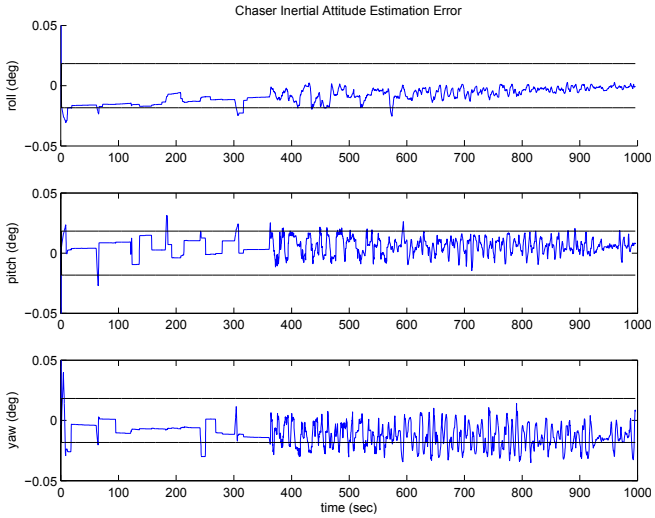


Figure 15: Inertial Attitude Navigation Performance

9. CONCLUSIONS

Lidar hardware options and configurations, as pertaining to spacecraft relative navigation, have been discussed. It is important to select to hardware with the performance specifications that will meet the requirements of the mission at hand. These performance specs will, in turn, drive the costs of lidar hardware. Methodologies for testing lidars and subsequent data analysis have been discussed. There are many options for testing, but usage of first-class facilities such as the SOSC drive down risk and provide valuable hands-on experience for engineers. Capabilities such as closed-loop testing are critical to understanding the behavior of the sensor at hand, as well as the navigation system as a whole. Vision and identification algorithms were presented. These algorithms are necessary for extracting all of the data from flash lidars. The vision algorithms are on the critical path and must be thoroughly explored and well-designed if the system is to work correctly. Much work remains to be done in this area. The rest of GN&C, including the EKF, are mature and well-understood components. Despite utilization of mature algorithm, the implementation details make all of the difference in the overall performance of the navigation system. The data analysis shows that even if every detail must function exactly as intended, subtle error sources can induce biases that can result in violating requirements.

ACKNOWLEDGMENTS

The effort required a large team of engineers over extended periods of time. The authors would like to acknowledge the Johnson Space Center team of Christopher D'Souza, Rebecca Johanning, and Renato Zanetti; the team at

Lockheed Martin SOSC led by Frank Moore, Sherri Ahlbrandt, Cory Burr, Reid Hamilton, Zachary Wilson, and David Huish. Finally, a special thanks to Chris Rossi for building offline replay tools for test data that enabled much analysis, and professor John Christian at West Virginia University for design of vision processing algorithms.

REFERENCES

[1] Christian, John A., D'Souza, Christopher N., Milenkovic, Zoran, Johanning, Rebecca "Closed-Loop Testing of the Orion Rendezvous GNC Algorithms in the Space Operations Simulation Center", 36th Annual AAS Guidance and Control Conference, February 1-February 6, 2013, Breckenridge, Colorado, AAS-13-036.

[2] Ahlbrandt, Sherri, Moore, Frank, Huish, David, Burr, Cory, Hamilton, Reid, "The Space Operations Simulation Center: A 6DOF Laboratory for Testing Relative Navigation Systems". 37th Annual AAS Guidance and Control Conference, January 31 – February 5, 2014 Breckenridge, Colorado.

[3] Vandaele, W., *Applied Time Series and Box-Jenkins Models*, Academic Press, Inc., 1983.

[4] Gelb, A., *Applied Optimal Estimation,* M.I.T. Press, 1974.

[5] Christian, J. A., Robinson, S. B., D'Souza, C. N., and Ruiz, J.P., "Cooperative Relative Navigation of Spacecraft at Close Range using Flash LIDARs, " accepted for publication in Journal of Guidance, Control, and Dynamics, 2013.

[7] Lefferts, E., Markley, F., and Shuster, M., "Kalman Filtering for Spacecraft Attitude Estimation," Journal of Guidance, Control, and Dynamics, Vol. 5, No. 5, Sept.–Oct. 1982, pp. 417–429.

[8] Glandorf, D.R, "An innovative approach to the solution of a family of linear programming problems." Jan 1, 1987. NASA. NASA-CR-182987.

[9] Haralick, R., Joo, H., Lee, C., Zhuang, X., Vaidya, V., and Kim, M., "Pose Estimation from Corresponding Point Data," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 6, 1989, pp. 1426-1446.

[10] Milenkovic, Zoran and D'Souza Christopher, "The Space Operations Simulation Center (SOSC) and Closed-Loop Hardware Testing for Orion Rendezvous System Design", JSC-CN-26684, AIAA conference, Minneapolis, MN, 13-16 August 2012.

[11] Morrison, D. F., *Multivariate Statistical Methods*, 2nd edition, McGraw-Hill, Inc., 1976.

[12] Cleveland, W.S *Journal of the American Statistical Association* Vol. 74, No. 368 (Dec., 1979), pp. 829-836

[13] Kutner, M. H., *Applied Linear Statistical Models*, 5th edition, McGraw-Hill, Inc., 2005.

BIOGRAPHIES

Zoran Milenkovic *received a B.S. in Aerospace Engineering from Iowa State University in 2004 and an M.S. in Aerospace Engineering from the University of Houston in 2010. His work at the Draper Laboratory has been focused on the area of guidance and navigation for rendezvous and proximity operations for the Space Shuttle, the Orion MPCV, and the Sierra Nevada Dream Chaser amongst others.*

Jack Brazzel *currently serves as the branch chief for Johnson Space Center's (JSC) Autonomous GN&C Systems Branch in the Engineering Directorate. He earned a B.S. in Aerospace Engineering from the University of Texas at Austin in 1989 and M.S. in Space Science from University of Houston – Clear Lake in 2002. He worked at Boeing in Houston for 14 years before joining NASA JSC in 2003. He has worked many aspects of GN&C and rendezvous, proximity operations, and docking for many different programs including Orbital Express, Space Shuttle, International Space Station (ISS), Hubble Rescue Vehicle, Orion Crew Exploration Vehicle, Commercial Crew, and various visiting vehicles to ISS. Primary focus has been GN&C algorithm development and integration, sensor testing, trajectory design and planning, and manual piloting and monitoring.*

Fred Clark *has spent more than 30 years as a navigation specialist at Lockheed Martin and Draper Laboratories in Houston, Texas. After working as a statistician for the state of Michigan, Mr. Clark relocated to Houston to work on Space Shuttle rendezvous and proximity operations, where he was one of the original authors of the Rendezvous and Proximity Operations Program (RPOP). He involvement in human spaceflight has transitioned to the Orion program.*