Simulation Results of ISS AR&D Using Only Range Images

Reuben R. Rohrschneider[1], William Tandy[2], Jeff Bladt[3], and Ian J. Gravseth[4]

Ball Aerospace & Technologies Corp., Boulder, CO, 80301

NASA's future plans for space vehicles call for the ability to automatically rendezvous and dock (AR&D) with the International Space Station (ISS) and other targets. This requires sensors and algorithms capable of determining the relative position and orientation (pose) between the target and chase vehicles under the drastically varying lighting conditions of low Earth orbit and beyond. To this end, Ball Aerospace has developed algorithms to produce six degree-of-freedom navigation data from 3D point clouds. The algorithms require a-priori knowledge of the target vehicle geometry and a range image of the target vehicle for in-flight pose determination (no visible or reflective targets are needed). The algorithms have been incorporated into a simulation that includes a flash LIDAR model, orbital dynamics, vehicle thrust control, and a three-dimensional model of the ISS. The flash LIDAR is used as the only relative navigation sensor during AR&D. In this paper we present the results of the docking simulation, including the accuracy of the pose determination algorithms during a successful approach and docking with ISS.

Nomenclature

AR&D	=	autonomous rendezvous and docking
CCD	=	charge coupled device
ICP	=	iterative closest point
ISS	=	International Space Station
LIDAR	=	LIght Detection And Ranging
PID	=	Proportional Integral Differential (controller)
OSG	=	Open Scene Graph
pose	=	relative position and orientation
STORRM	=	Sensor Test for Orion RelNav Risk Mitigation

I. Introduction

RENDEZVOUS and docking are currently performed either manually, or in a highly cooperative manner that requires active hardware on both target and chase vehicles. The Sensor Test for Orion RelNav Risk Mitigation (STORRM) relative navigation sensor suite, flown in 2011 on STS-134, advanced the current state of the art in cooperative rendezvous and docking technology [1]. To enable autonomous rendezvous and docking (AR&D) with cooperative targets the STORRM sensor suite requires an active flash LIDAR unit on the chase vehicle and only passive reflectors on the target vehicle. The flash LIDAR used on the STORRM instrument produces range and

[1] GN&C Technology Initiative Lead, Civil and Operational Space, 1600 Commerce St., AIAA Member.

[2] Senior Engineer, Structural Engineering, 1600 Commerce St.

[3] Principal Engineer, Spacecraft Systems, 1600 Commerce St., Senior AIAA Member.

[4] Principal Engineer, Spacecraft Systems, 1600 Commerce St.

intensity images of the target. In planning a sensor for semi- and non-cooperative AR&D, it is easy to see that these two sets of images are ideal data for these missions.

When discussing AR&D it is important to clarify the differences between cooperative, semi-cooperative and non-cooperative AR&D. Cooperative AR&D needs both vehicles to work together with one vehicle at least providing easy to decipher pose indications. Semi-cooperative AR&D assumes that there is some a-priori knowledge of the target vehicle geometry and that the target vehicle is in a constant torque and force state. This means that the vehicle may be spinning, or a torque rod may be powered in a steady-state condition, but the vehicle is not changing directions or actively avoiding the chase vehicle. Non-cooperative AR&D makes none of these assumptions and is the most general case. This paper covers the semi-cooperative case, though much of the work presented also applies to the non-cooperative case.

In choosing a data source, the current sensor options for AR&D are the previously described LIDAR (which provides lighting-independent data) or visible cameras. Data from either of these sensors can be used to determine relative pose. However, the primary drawbacks of visible cameras are lighting dependence and poor range determination performance at large distances. The weaknesses of the visible cameras can be avoided by using only the range data from the LIDAR. An ideal solution might combine the visible and LIDAR sensors, but while sensor fusion is a current trend in Earth-based applications such as robotics, power and weight requirements for space missions seek to minimize sensor counts. If a LIDAR only solution is achievable it would be strongly preferred. In terms of future mission planning, a working LIDAR-only pose determination solution validated for the ISS can also be extended to other problems such as servicing of existing space assets without reflective targets or rendezvous and landing on an asteroid.

In this paper we present the results of an end-to-end study performed by Ball Aerospace to automatically determine relative pose based purely on range images. First, a flash LIDAR model was built to aid in algorithm testing. This was tied into closed-loop simulations of AR&D. Then, two algorithms were written to meet the pose determination objectives: one for medium ranges to the target and one for short ranges. A brief overview of the methods and models is presented, followed by performance results for an ISS approach and docking.

II. Methodology

Approach analysis to the ISS found the pose determination solution spaces diverged into two ranges: a long to medium range and a short range. The division was found to occur naturally at the point where the target vehicle just filled the field-of-view (FOV). After that point clipping of the target vehicle occurs and, while some of the medium range algorithms can tolerate a small to moderate amount of clipping, they generally perform better when the entire target vehicle is visible.

To test these algorithms a flash LIDAR model was built to generate range images. This same flash LIDAR model was also incorporated into a closed loop simulation to determine if the pose performance was sufficient for AR&D. The pose algorithms, flash LIDAR model, and simulation are described briefly below.

A. Medium-Range Algorithm

Several medium-range algorithms were tested, and the iterative closest point [2,3] tracking method is presented here. The method relies on matching the current range image to a model of the target vehicle so that an absolute pose can be determined to a known point on the vehicle. At long ranges the base algorithm works well, but it was found to produce incorrect pose at closer ranges. The cutoff was determined experimentally using the simulation described below, and occurred when approximately 20% of ISS was outside the FOV. Several modifications were made to the algorithm to enable operation at closer ranges, including cropping the reference model based on the last known relative pose. This improved the ability of the algorithm to converge on the correct solution by eliminating spurious points that could incorrectly influence the optimization process.

While these results use a reference model, the method could be adapted to unknown targets by first scanning a target vehicle by circumnavigation (using a stand-off distance), followed by continued tracking of the object relative to the model built during the circumnavigation. Target vehicles could then be a damaged spacecraft, space debris, or an asteroid. During the initial scan of an unknown object, the ICP algorithm has been successfully used to refine the

relative position of successive range images. The refined relative pose can then be used to correct the inertial navigation state.

B. Short-Range Algorithm

The determination of target pose at medium to long range distances is difficult, but is helped considerably by being able to see at least large parts of the target. Computation time at distance is also reduced because only part of the CCD contains data so analysis can be done on reduced pixel counts. However, as the docking vehicle approaches the target the view will begin to become clipped. The beginning of significant view clipping marks the start of short range algorithms.

Difficulties at short ranges include a full CCD of pixel returns (which reduces computational shortcut options), more rapidly changing features, and material textures that can add noise at close range. However, the most difficult problem is that instead of navigating around an object we are now navigating along an object. This change requires intra-object landmarks to be detected and used for close-in operations. The databases and detection algorithms to do this take a noticeable jump in complexity.

Initial efforts looked at using RANSAC [4] to find shapes in the 3D data. The idea has merit as most manmade space structures are cylinders and planes. Unfortunately, while moderately successful without much effort, the computation time was unacceptable at tens of seconds per analyzed frame.

The next iteration looked to create one histogram per view, such as seen in the Fast Point Feature Histogram [5]. Each view was processed into a single histogram and these were collected into a database of all the views deemed likely in a docking scenario. In testing this was both more reliable than RANSAC and much faster to compute coming in at just a couple of seconds per frame. However, this created a need for multiple, large databases that had to be swapped in and out of memory depending on the range to the target. Indeed, as testing progressed it became clear that the number of databases and the reliance on model accuracy was becoming prohibitive for current space hardware limits.

After a number of algorithm iterations it was seen that something akin to a Pyramid of Images [6] approach was efficient. By mixing in a combination of global and local features the pyramid could be guided to each successive level, making it both an efficient and reliable method for our purposes. Specifically, each range image that is collected is segmented into a 3x3 grid. Within the 9 grids a number of features are calculated. These features range in complexity from simple pixel counts to moderately higher order features using slopes and normals. At the coarse 3x3 grid level the features are not required to be accurate. Instead, they must merely be indicative. Grid regions that meet assigned criteria are then further segmented into their own 3x3 grid and additional features that tend to be more discriminative are calculated for these smaller regions. Features in the smaller region that meet their criteria are then used in the final pose calculation. The criteria range from simple value thresholds to spatial relationships, but all are straightforward to calculate since processing power is at a premium.

By starting with a coarse grid, then calculating higher order (and more discriminative) features only on subsets of these larger grids, processing time can be compressed to 25+ Hz even on the limited processing power of space qualified hardware. Figure 1 outlines the general steps taken to narrow down the segments of the image that the processing time is spent on. Pose estimation results for out-of-plane angles typically seen in testing are show in Figure 2. The errors are low and the results are typically robust, but there are occasional poses that produce moderately larger errors for the algorithms as seen by the red squares in Figure 2. While these errors are much higher than the mean error they are still relatively low.

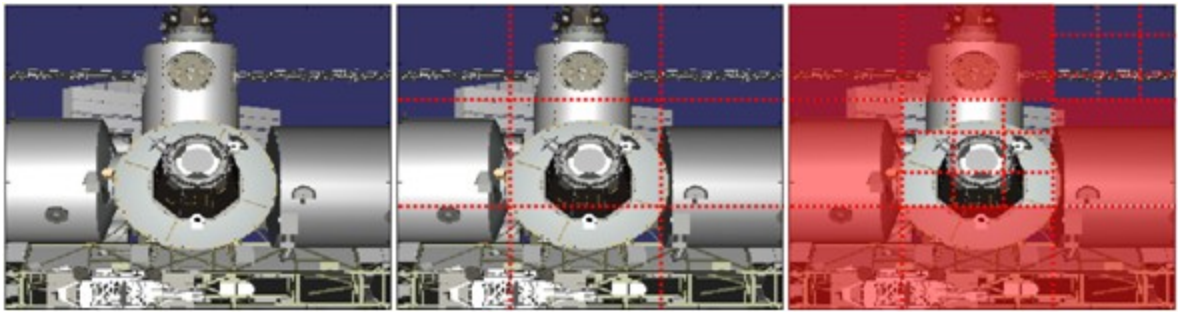

Figure 1: Left: The raw range image is received. Middle: The image is broken into a 3x3 grid and features are calculated for each region. Right: Regions that are indicative of key areas are further segmented into 3x3 regions and higher order features are calculated. Small regions that pass their criteria are used to estimate pose.

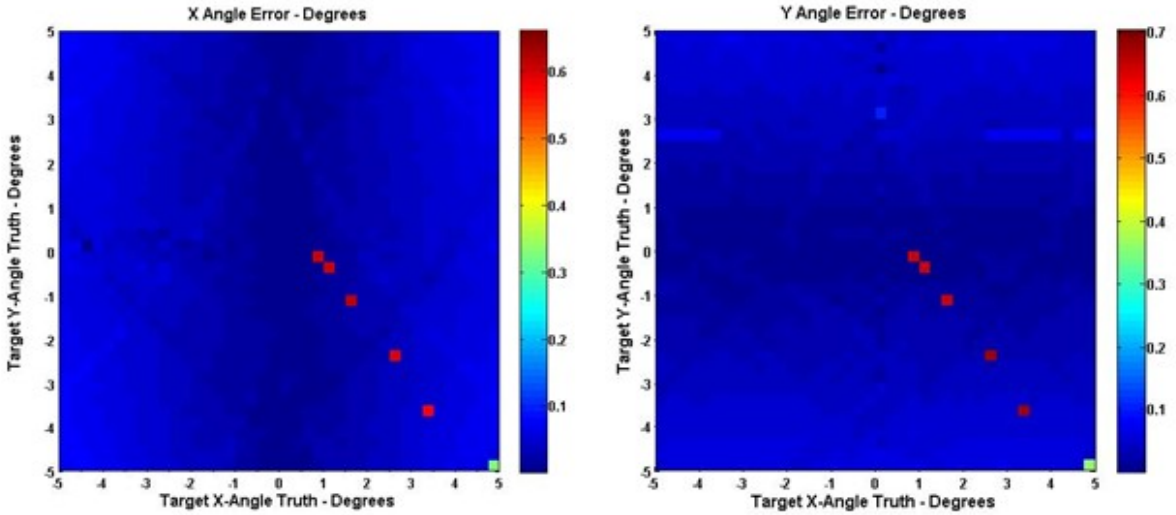

Figure 2: Pose estimation errors for the X and Y angles. Mean errors are 0.037 degrees with standard deviations of 0.045 for both angles.

C. Flash LIDAR Model Description

A way to simulate a flash LIDAR unit was necessary in order to generate arbitrary views of target spacecraft for algorithm testing and to generate target spacecraft databases. To fulfill this need a model was built based on the OpenGL language due to its availability and the pervasiveness of hardware support on graphics cards in nearly all modern computers. An interface was then written between Matlab and OpenGL so that the algorithms could navigate around the structure by inputting a pose and receiving the resulting model view. The readily available optimized software and hardware acceleration make for an adaptable model that runs at greater than 30Hz on a modern desktop (3.2 GHz Intel Xeon with a Nvidia Quadro FX 1800 graphics card) when producing 256 by 256 range images. The infrastructure is in place to provide realistic noise models, but these have not yet been used for this effort. Perfect range images have been used for the simulations shown in this paper.

D. AR&D Simulation

Flight control algorithms for semi-cooperative autonomous rendezvous and docking (AR&D) are tested using a MATLAB-based, non-real-time, closed-loop simulation that features an OSG-based LIDAR camera model and includes standard attitude and orbit dynamics models for both the chaser and target vehicles. Figure 3 illustrates the functional flow of the closed-loop AR&D simulation.

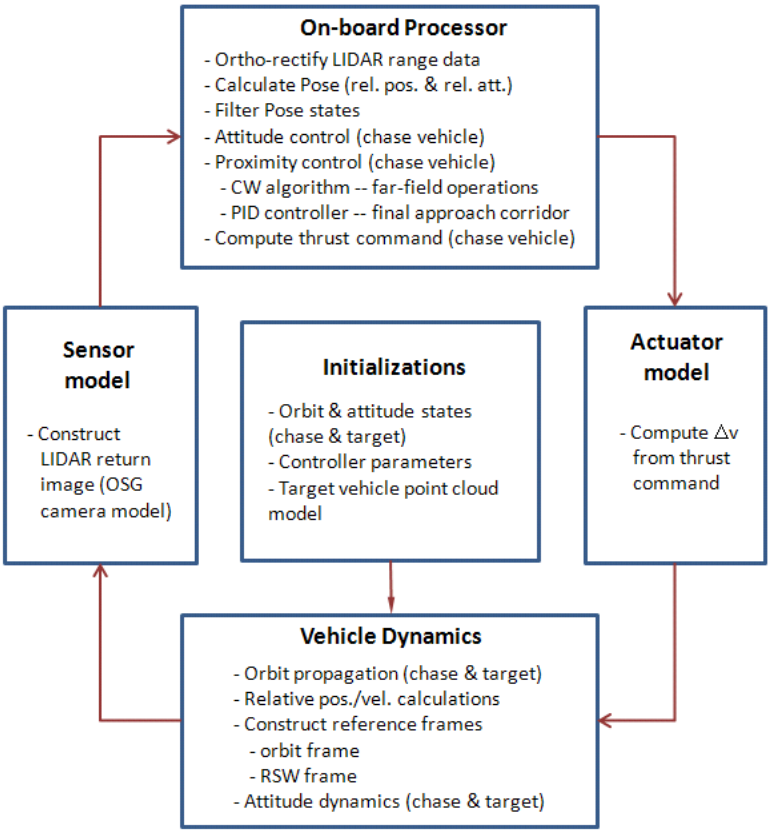

Figure 3. Semi-cooperative AR&D non-real-time, closed-loop simulation featuring an OpenGL-based LIDAR model and 6-DOF dynamics and control.

Vehicle dynamics (truth models) for the chaser and target vehicles are comprised of a 13 x 13 gravity geopotential model, atmospheric drag model assuming constant density, and rigid-body attitude dynamics using Euler equations. The LIDAR sensor is simulated using an OSG-based camera model and point cloud target vehicle data bases. The simulation user may select either ISS or Hubble Space Telescope point cloud models. The OSG camera model forms LIDAR return signals (i.e., range data) based on (truth model) relative position and attitude between the chase and target vehicles along with the target vehicle point cloud model.

LIDAR return signals pass from the sensor model into the on-board processor model, where an ortho-rectification algorithm generates 3D relative position measurements for each pixel in the camera focal plane. Next, the ICP algorithm processes the relative position measurements into pose estimates (i.e., relative position and orientation between the two vehicles), which are passed through a median filter to reduce signal noise. Chase vehicle attitude control maintains its local vertical local horizontal (LVLH) orientation as it closes on the target vehicle. Chase vehicle position relative to the target vehicle is controlled by a Clohessy-Wiltshire algorithm during far-field operations and by a standard proportional-integral-derivative (PID) controller for near-field operations, especially once the chase vehicle enters the final approach corridor. The simulation user may select the range at which the transition between far-field and near-field operations occurs. Lastly, desired thrust is computed and passed into the actuator model, which computes the resulting thrust-induced change in velocity and closes the loop by updating the chase vehicle's velocity in the vehicle dynamics model.

III. Results

The above algorithms and simulator were combined, and analysis was run starting at 100m to the docking ring until only the inside of the docking ring is visible (approx. 1.5m). At this point insufficient features exist in our current model to determine pose. The simulation is initialized in a perfect state and the chase vehicle performs a maneuver to intercept the target vehicle based on the initial perfect knowledge. The measured relative pose is then used with the PID controller to guide the chase vehicle to the docking ring of the target vehicle. The relative position of the chase and target vehicles are shown in Figure 4. The relative position calculation is quite accurate, with typical errors less than 0.2 meters. Error plots for both position and orientation are shown in Figure 5.

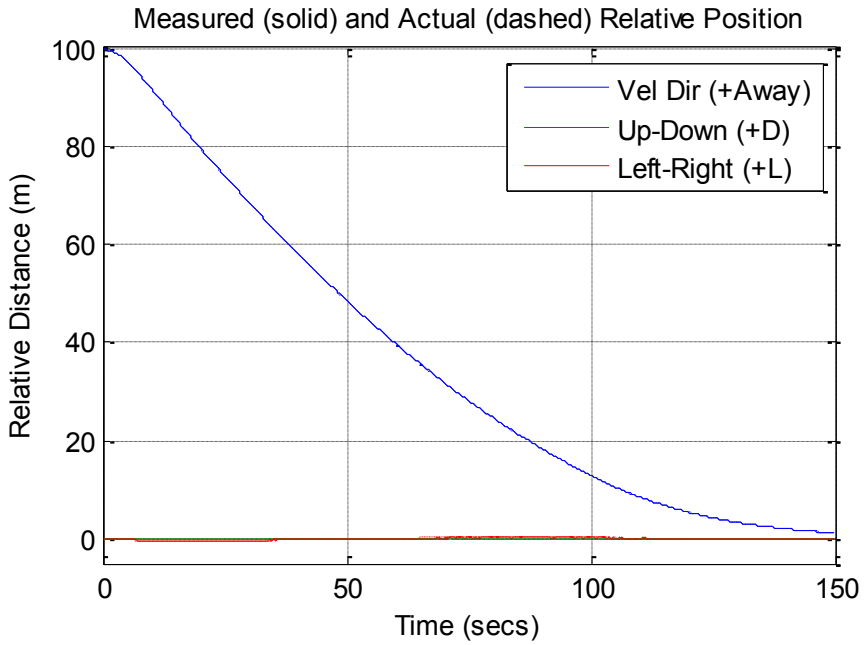

Figure 4. Relative position of chase and target vehicles.

The computed pose is very good except for a region between 65 and 115 seconds, where the ICP algorithm begins to break down as the ISS gets closer to filling the field of view. The region from 0 to 65 seconds where noise is very small is the initial ICP algorithm used when ISS is fully contained within the FOV or minimal clipping of the solar panels is present. The algorithm accuracy is reduced once significant portions of ISS are outside the FOV, and minor modifications were made to the algorithm to produce a working solution. This modified algorithm produces more noise between 65 and 115 seconds. The modified algorithm fails to produce usable results at ranges closer than approximately 7 meters, at which point the close range algorithm is used. As expected, this algorithm produces very good estimates of the range to the docking ring. The accuracy in the image plane is slightly worse due in part to the lack of features on the current ISS model within the docking ring (the docking cross is missing in our model). The point at which the in-plane position begins to drift (approx. 150s) is where only the edges of the docking ring petals are visible, and the FOV is dominated by a featureless pressure bulkhead. The addition of features within the docking ring to our model is a topic of future work that should improve the accuracy of the relative position computation. Despite the errors observed, the docking simulation shows good tracking to the center of the docking ring until all three petals are not visible (approximately 155s in the simulation) and the separation distance is approximately 2 meters.

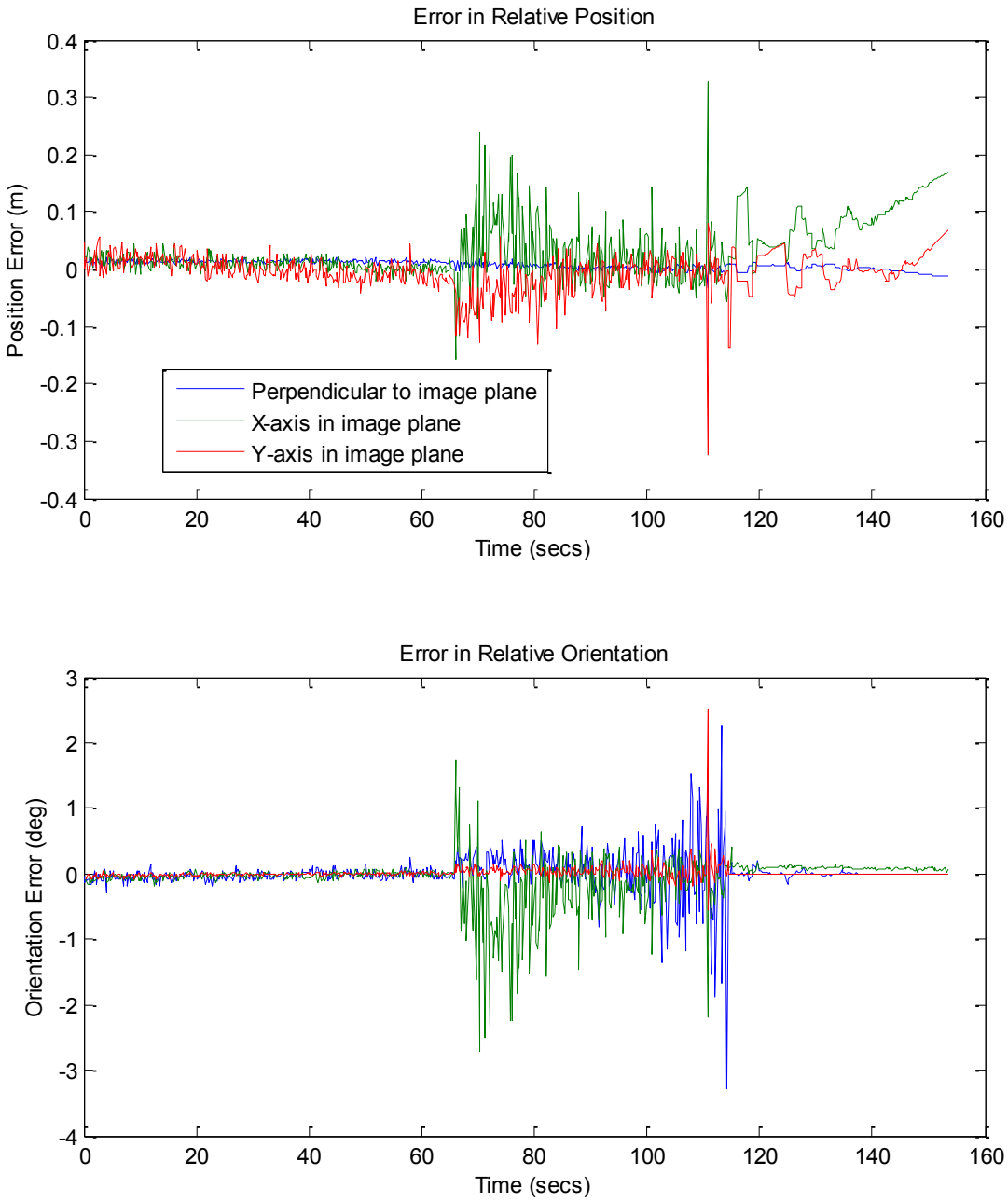

Figure 5. Relative orientation of chase and target vehicles.

The computed relative orientation exhibits low noise with the exception of the region where the modified ICP algorithm is necessary. Even at distances of 100m to the docking ring the relative orientation is predicted to within a small fraction of a degree in all axes. During the final 7m of approach (115s to 160s in the simulation) the noise is very low as the close range algorithms are capable of determining feature positions more accurately as features become better resolved at shorter ranges. The rotation about the flash LIDAR viewing vector is not determined for the final 7m since insufficient features are visible to determine this rotation (the docking ring has 120 deg symmetry). Instead, the algorithm maintains the current rotation about the viewing vector and shows no rotation error in Figure 5 (red curve) during this portion of flight since this rotation is not computed.

Computed relative position is filtered as described in the AR&D Simulation section above, and relative velocity is calculated from the filtered relative position estimate. Relative velocity is an important part of the PID controller and the small noise in the relative velocity is amplified in the thrust control. Figure 6 shows the computed relative velocity and the resulting thrust commands. This is a simple simulation intended only to prove the effectiveness of the pose algorithms (very simple control method and very simple navigation filtering). Showing a successful docking demonstrates that the pose algorithms are effective, even in the absence of knowledge of target vehicle motion and realistic flight filters.

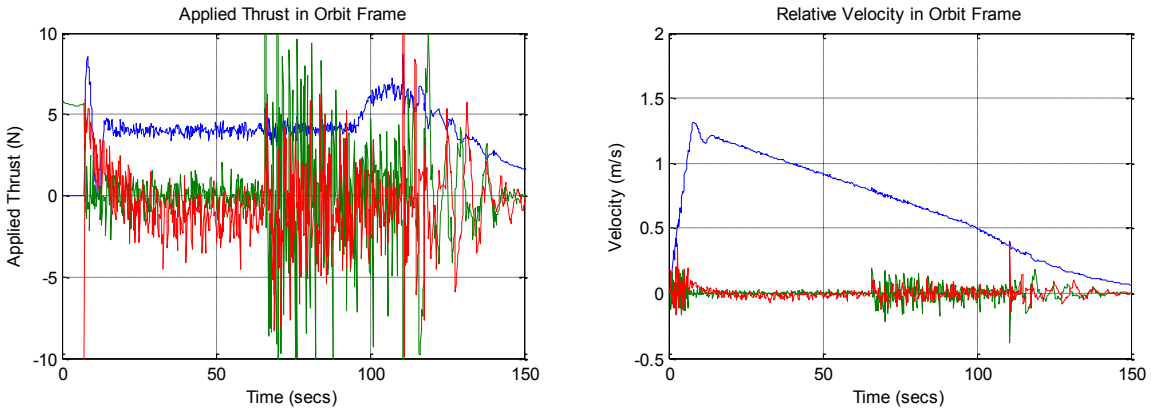

Figure 6. Computed relative velocity and the resulting thrust commands.

IV. Improved Close Range Pose Algorithm

In the above analysis it was discovered that an improved close range pose algorithm was needed to refine the docking performance. Analysis of the data obtained from the STORRM mission also informed the development of the improved close range pose algorithms. The important lessons from analysis of the STORRM data were that some surfaces were not visible due to shallow incidence angles, there is a moderate level of background noise and bad pixels, and that the standoff cross on the docking target is not clearly visible. These observations drove us to create a computationally efficient noise model that could be used for AR&D simulations. The model developed is not radiometrically correct, but attempts to mimic some of the behavior observed in the STORRM data. Figure 7 compares a range image from STORRM to one produced by our computionally efficient model.

Using both the STORRM data and the computational noise model, improvements were made to the close range pose algorithm to ensure accurate results in the presence of noise, but also to ensure improved position determination during the final meters of docking. The resulting algorithm is still computationally efficient, and does not use the docking cross, but rather point clustering and multiple metrics to identify the docking ring using only range data. The performance of the improved close range algorithm are shown in Figure 8 for a number of ranges to the docking ring and relative orientations. The center of the range of angles shown (+/- 4 deg) in Figure 8 are those most common in the final meters of docking, and show errors between +/-10cm (compared to +/-20cm in the algorithms presented in Section IIB). The rotation about the viewing vector is still not determined, and errors for range to target and orientation angles are comparable to those for the algorithm presented above.

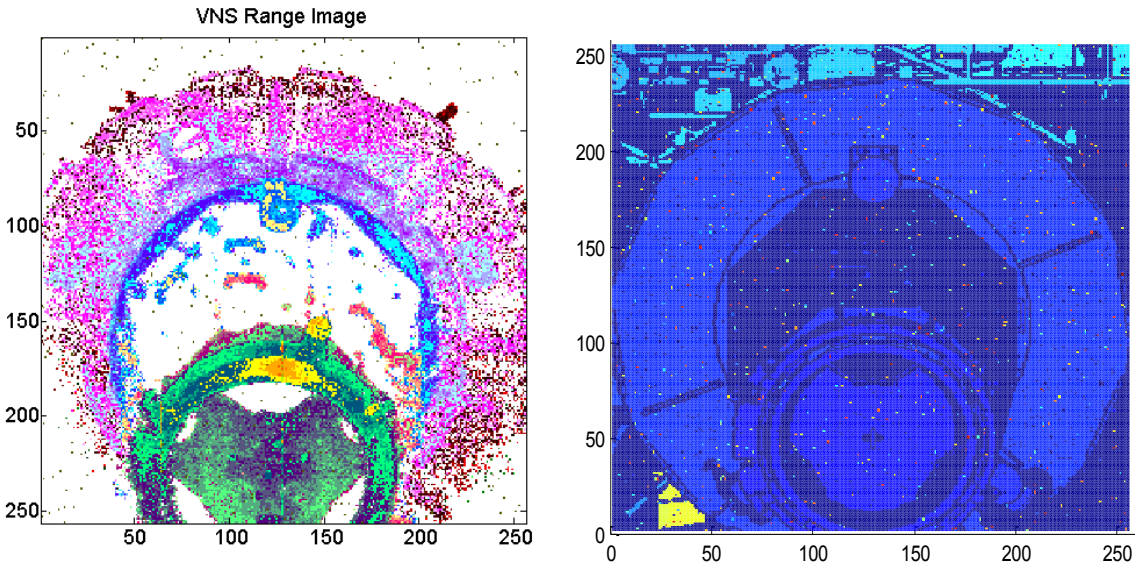

Image courtesy NASA.

Figure 7. Comparison of a STORRM range image to one from our computationally efficient noise model. In the left image white indicates no return signal, and corresponds to dark blue in the right image.

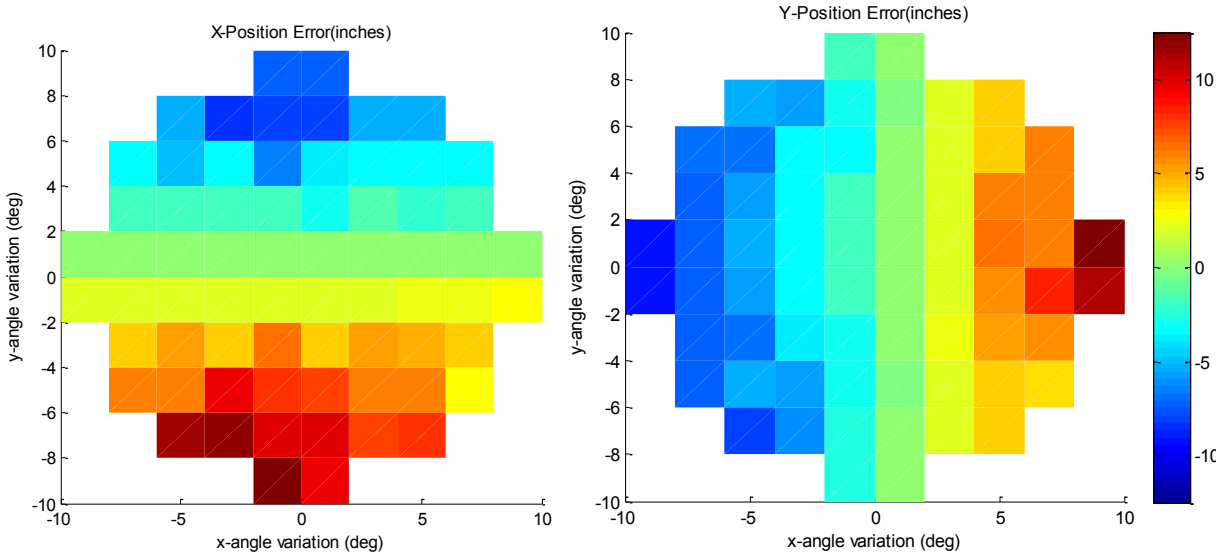

Figure 8. Performance of the improved close range pose algorithm shows significant improvement in position error compared to the old algorithm.

V. Conclusions and Future Work

By building an end-to-end simulation platform the team was able to test and compare a large variety of pose determination algorithms. The platform is perhaps the best achievement of this effort as it builds a strong foundation for future efforts. For this paper it was seen that through testing AR&D algorithms there was not a singular solution that worked at all ranges. Instead, multiple approaches had to be used based on the range to target.

The ICP algorithms created the core approach for much of the approach-to-target range, but for near-field operations a form of pyramid of images approach was created. For the close operations the pyramid approach was tailored so that the coarse regions that were indicative of relevant data were further processed. An improved close range pose algorithm was presented that makes use of the spacial efficiency and offers better position estimates during the final meters of docking, and is still computationally efficient. The tailoring process enabled good pose estimation while operating at a high enough frequency to fit onto existing space hardware.

References

1. I.J. Gravseth, R.R. Rohrschneider, and J. Masciarelli, "Vision Navigation Sensor (VNS) Results from the STORRM Mission," 35th Annual AAS Guidance and Control Conference, Breckenridge, CO, Feb. 3-8, 2012.
2. P. Besl, and N. McKay, "A Method for Registration of 3-D Shapes," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, 1992, pp. 239-256.
3. H.M. Kjer, and J. Wilm, "Evaluation of Surface Registration Algorithms for PET Motion Correction," Bachelor Thesis, Technical University of Denmark, June 2010.
4. R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection" *Computer Graphics Forum*, Vol. 26, No. 2, June 2007, pp. 214-226.
5. R.B. Rusu, N. Blodow, and M. Beetz , "Fast Point Feature Histograms (FPFH) for 3D registration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17, 2009, pp.3212-3217.
6. J.L. Crowley, and A.C. Sanderson, "Multiple Resolution Representation and Probabilistic Matching of 2-D Gray-Scale Shape", *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 1, January 1987, pp. 113-121.