Assessment of Sensor Data Accuracy within Gazebo/ROS for High-Precision Autonomous In-Space Robotic Operations

Jessica S. Friz[1], John Cooper[2], and Cameron Miksch[3]
NASA Langley Research Center, Hampton, VA, 23666, USA

Shashank Kalluri[4]
University of Michigan, Ann Arbor, MI, 48109, USA

Modeling high-precision in-space servicing, assembly, and manufacturing operations in a simulated environment is a critical step in the development of robotic systems that will be used to autonomously assemble large-scale structures in space. Limited facility size and high costs for manufacturing prototypes make it challenging to conduct full-scale operational testing under appropriate environmental conditions; therefore, testing in a modular, high-fidelity simulation environment is necessary for verification and validation of technology and architecture designs prior to launch. Several modeling and simulation environments exist both within NASA and industry that can be used to test robotic system design and operations, including the widely used commercial tool Gazebo integrated with Robotic Operating System software. Because the performance of autonomous robotic systems relies heavily on the quality of sensor input data, this paper focuses on assessing the accuracy of pose data from an optical sensor model in the Gazebo environment against the behavior of real hardware. The results of the tests will help developers using Gazebo for large-scale, high-precision simulation to account for modeling inaccuracies within their robotic control system algorithms.

I. Background

Large extraterrestrial structures like the In-Space Assembled Telescope (iSAT) [1], the Lunar Gateway [2], and the Lunar Safe Haven [3] (See Figure 1) will enable humanity to study and explore the universe farther than ever before; however, due to the size and complexity of the construction of these structures, deploying them via traditional methods from volume-constrained launch vehicles will no longer be a viable option. In-space servicing, assembly, and manufacturing (ISAM) operations will be necessary to autonomously assemble these structures when human supervision and intervention is not possible. ISAM technology developers must be able to trust that their autonomous systems can successfully complete high-precision tasks, such as mirror placement on the iSAT, given a variety of situational factors. High precision, as defined by the NASA Science & Technology Mission Directorate (STMD), indicates placement and alignment errors of less than 0.005 m translationally and 0.0873 rad rotationally. Limited facility size and high costs for manufacturing prototypes make it challenging to conduct full-scale operational testing under relevant gravitational and environmental effects. Modeling high-precision ISAM operations in a modular, high-fidelity simulation environment will be a critical step in the verification and validation (V&V) of ISAM mission architectures prior to launch.

[1] Aerospace Engineer, Simulation Development and Analysis Branch
[2] Research Aerospace Engineer, Autonomous Integrated Systems Research Branch
[3] Systems Engineer, Simulation Development and Analysis Branch
[4] Student, Department of Aerospace Engineering

Figure 1 In-Space Structures

Several high-fidelity modeling and simulation environments exist both within NASA and industry that have been used for V&V of robotic system design and operations. Engineers within the NASA Langley Research Center (LaRC) ISAM community use some of these environments for their own technology development, including the Langley Standard Real-time Simulation in C++ (LaSRS++) [4] and the Baseline Environment for Autonomous Modeling (BEAM) [5]; however, commercially available tools, such as Gazebo [6], are more widely used across the greater ISAM community and provide developers with the flexibility to take advantage of open-source packages, interfaces, and integrations that have not been available in the past. While these simulation environments are powerful tools, more testing is needed to assess their applicability for modeling short-range, high-precision ISAM operations. Different ISAM concepts will require different levels of modeling accuracy, but for operations like those that will be performed on the iSAT, sub-millimeter accuracy will be required for certain parts of the structure. Trust in the simulation environment must be established to perform reliable V&V.

Generic ISAM modeling and simulation capabilities were assessed within LaSRS++ during the On-orbit/On-surface Servicing, Assembly and Manufacturing (OSAM) Architecture Simulation System (OASiS) project [7]. An initial assessment of the fidelity of the sensor models in the simulation environment compared to real hardware was completed. The pose data from the sensors in the simulation environment was only accurate translationally to within 20 cm of the hardware results and the rotational data was too inaccurate for comparison. While there are many elements of the development and testing process that could have been improved, these initial results indicate that the simulation, in its current state, does not achieve the accuracy necessary for modeling high-precision operations on the iSAT.

Gazebo has been identified as a tool used commonly across the ISAM technology community, including ISAM technology developers at NASA LaRC. Many studies have been conducted to assess Gazebo's fidelity for in-space robotic operations [8] [9] [10] [11], including an assessment of LiDAR sensor modeling accuracy and scalability given different GPU limitations [12]. Gazebo is an open-source robotic modeling software with many downloadable resources, a strong online support community, and streamlined integration of self-developed models and autonomous control frameworks such as the Robotic Operating System (ROS) [13]. All these features make it an attractive tool for ISAM modeling and simulation; however, because the performance of autonomous robotic systems relies heavily on the quality of sensor input data, assessments must be conducted to measure the sensor modeling accuracy of the Gazebo simulation environment against the behavior of real hardware. ISAM technology developers using Gazebo for simulation must be able to account for modeling inaccuracies within their control system algorithms. This will ultimately lead to more robust and trusted technology designs that can achieve higher levels of operational accuracy.

II. Modeling Approach

A. Problem Definition and Task Breakdown

While there are a variety of sensors available on the market that may be used for in-space robotic operations, tests will need to be conducted to assess which types of sensors (e.g., optical, LiDAR, etc.) will be better suited for high-precision operations. Optical sensors may prove to be more optimal for this application; therefore, the team chose to evaluate the accuracy of pose data returned from an optical sensor model simulated within Gazebo against analog hardware testing of a relevant ISAM operation. Two key operations were identified as candidates for modeling and testing, including:

- Close-proximity Grapple Tool assembly operations to support the Precision Assembled Space Structures (PASS) project [14]
- Soft robotic end effector control for the Heat Integrated Compliant Lightweight Tool End-effector (HI-C LITE) project [15]

The PASS team will be conducting full-scale hardware demonstrations of the autonomous assembly of parts of the iSAT, including high-precision grappling and placement of tri-truss structures for the primary mirror backbone.

Some of the elements involved in this demonstration can be seen in Figure 2 including tri-trusses, commercial-off-the-shelf robotic arms, and sensors. This paper covers the procedures and testing conducted to assess the fidelity of Gazebo for modeling close-proximity operations to support the PASS project. In support of the HI-C LITE project, an assessment was conducted to understand Gazebo's capabilities for modeling soft body dynamics which is reviewed in a separate paper [15].

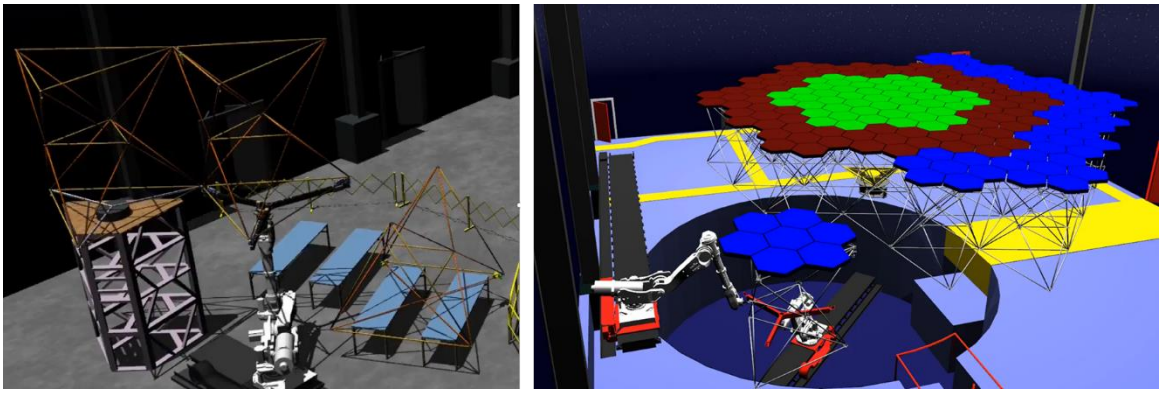

Figure 2 PASS Hardware Demonstration Elements

The following tasks were identified to evaluate Gazebo for modeling short-range, high-precision operations:

- Researching Gazebo and assessing its sensor modeling capabilities and limitations
- Defining models and test cases to build within the simulation environment and with hardware
- Selecting a sensor system for the tests, calibrating it, then modeling and calibrating it within the Gazebo simulation environment
- Building a test fixture with modified PASS hardware models, fiducial markers, and a sensor to replicate the short-range ISAM operation
- Developing simulated models of the hardware in Gazebo and adjusting lighting and material properties to match
- Calculating expected pose data from ArUco markers based on CAD geometry
- Running hardware and simulation tests and gathering pose data from the ArUco markers in the captured images
- Comparing pose data from markers between simulation environment, hardware tests, and expected results to evaluate sensor modeling accuracy

B. Test Fixture Design

For the PASS hardware demonstrations, a Grapple Tool is needed to align the tri-truss structures very precisely and is currently designed to use a depth sensor for feedback; therefore, the team wanted to analyze a variety of hardware and simulation testing configurations to understand how camera eyepoint distance and orientation affects the accuracy of the optical pose data. A test fixture was designed and built (See Figure 3) where an Intel RealSense D435i is mounted to a linear motion carriage with 15 discrete measurement locations at 0.0254 m intervals. The tri-truss node is mounted to a turntable with 36 discrete angular measurement locations at 0.1745 radian intervals. The test fixture design was imported into the Gazebo simulation environment to replicate the close proximity operations between the end of the Grapple Tool and the tri-truss node. The camera intrinsics (i.e., resolution, focal point, etc.) of the D435i were modeled in the simulation as well. A simple controller was programmed in ROS to incrementally change the eyepoint distance and location, which can be seen in Figure 4.

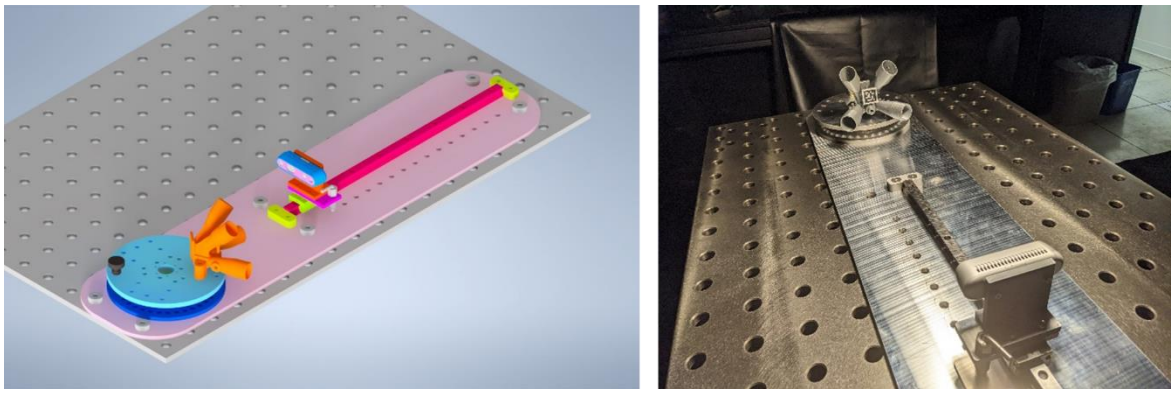

Figure 3 Small-Scale Test Fixture for PASS Hardware

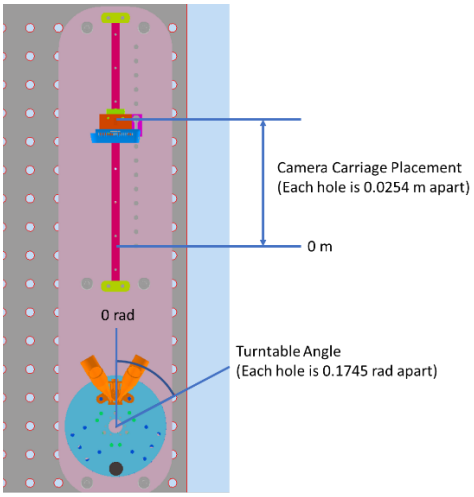

Figure 4 Camera Distances and Orientations for Testing

C. Pose Estimation

The team used the ArUco software package for pose estimation [16] – a commonly used tool among the ISAM community at NASA. ArUco is an open-source C++ library dependent on OpenCV [17] which can be used to create square fiducial markers. The markers are composed of a white binary pattern encoded onto a black square, each having a unique identification number (ID) linked to an ArUco dictionary. These markers allow for the localization of an object in a scene with respect to a sensor eyepoint by using a camera's intrinsics (i.e., camera matrix and distortion coefficients) obtained through a calibration process. When these markers are affixed to a known location on a structure, the pose of the structure can be derived from the pose of the marker. Having multiple markers with unique IDs in a scene reduces ambiguity for pose estimation of the structure overall.

For this project, the team used the 0 and 1 tag IDs associated with the MIP36H12 ArUco dictionary. All tags were 0.8 in (0.02032 m) on each side. The configuration of the tags on the simulated PASS hardware can be seen in Figure 5 below. It should be noted that the reference frame in ArUco was oriented differently than the global reference frame in the Gazebo environment and that the rotation vector returned from the ArUco algorithms is output in Angle-Axis format rather than Euler angles [18]; therefore, rotation transformations were applied to the extracted ArUco data to compare it with the expected pose data which was calculated in the Gazebo reference frame. Figure 6 shows the relative orientation of the reference frames used for analysis.

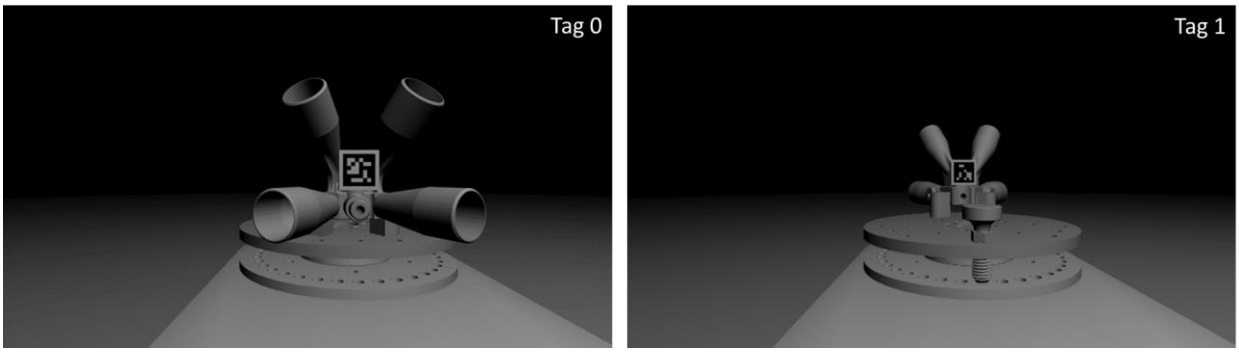

Figure 5 ArUco Tags on Test Fixture

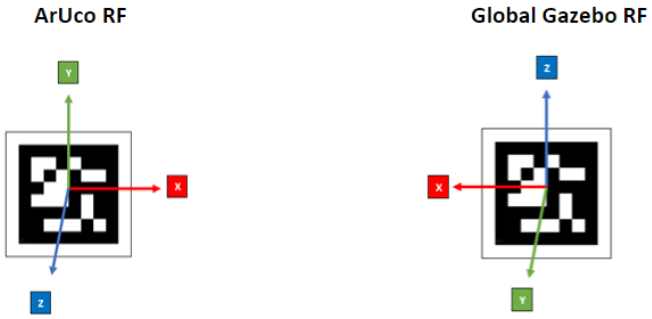

Figure 6 ArUco and Gazebo Global Reference Frames

D. Expected Pose Calculations

The expected translation and orientation of the ArUco marker centroids with respect to the camera eyepoint were determined through geometric transformations between the turntable and the camera carriage. Constructing reference frames at convenient locations such as the center of the turntable with the tri-truss node, the bottom of the carriage pin, and at the center of each ArUco marker allowed for a simpler mathematical definition of the relative positions of each of these components. These frames were defined using three-dimensional axes that follow the right-hand rule convention, and a transform tree was created from the eyepoint of the RealSense D435i camera to the individual ArUco markers.

An arbitrary global reference frame was defined in Gazebo (See Figure 7) to ensure consistent use of the positive and negative direction. The camera eye point reference frame is co-oriented with the global frame. All following transformations will follow the same convention. Translation in the camera carriage along the rail is represented as a positive change in the Y-axis, and the counterclockwise rotation of the turntable is described as positive rotation about the Z-axis.

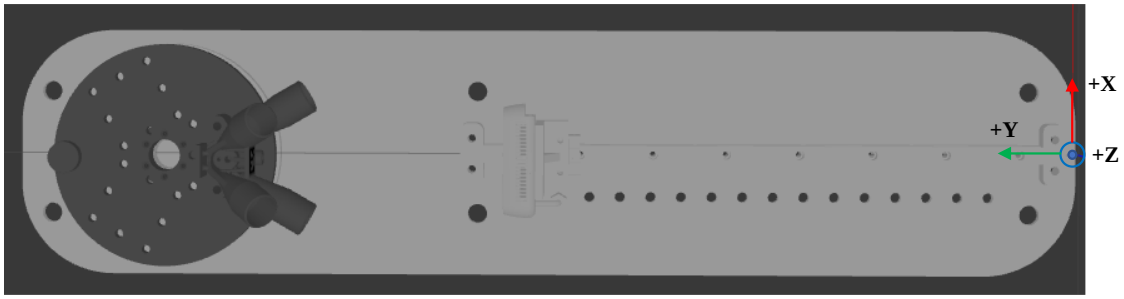

Figure 7 Global Reference Frame and World Origin

The transform tree begins at the eyepoint of the camera defined as the center of the RGB module lens in the camera. Further detail can be found within Intel's D400 documentation online [19], which shows the location of the RGB module origin from the center of the camera face. From this eyepoint reference frame, we can find the necessary static transformation vector to link the camera reference frame to the reference frame of the carriage as shown in Figure 8.

A similar process was employed to link the reference frame of the camera carriage to that of the turntable's center as shown in Figure 9. This transformation required a dynamic variable denoted by p, which indicates the number of pin holes the carriage is away from pin hole "0". The 0^{th} pin hole is defined as the hole closest to the tri-truss node, and the 14^{th} pin hole is furthest from the tri-truss node. For the final transform, a dynamic variable θ was used to account for the rotation of the turntable about the Z-axis. At 0 rad, θ represents the default state with the side of the tri-truss node displaying Tag 0 facing directly towards the camera (See Figure 4). Given that there are two ArUco markers on the node, two unique vector transformations are needed to define the position of the respective markers.

A summarization of the transformations can be found in Table 1, which highlights the transformation vector from the camera eyepoint to the center of each ArUco marker.

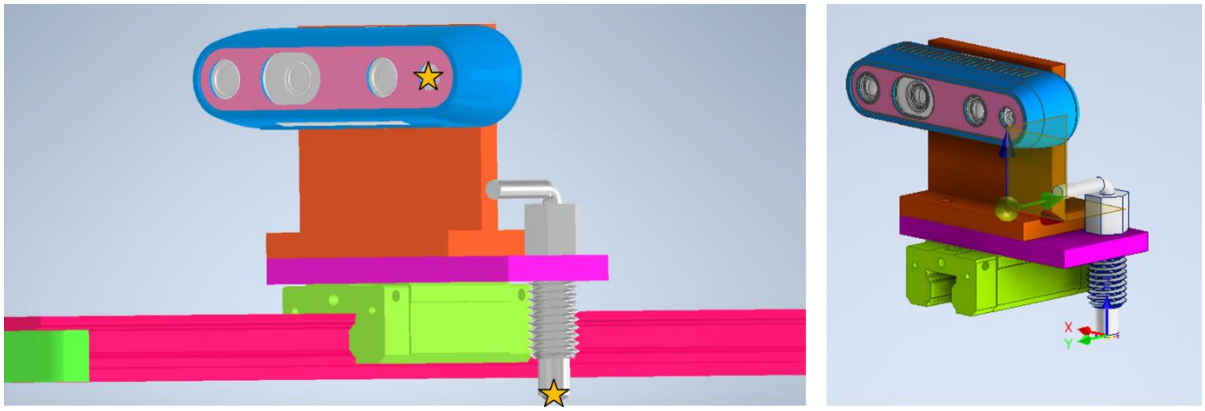

Figure 8 Camera Carriage Reference Frame

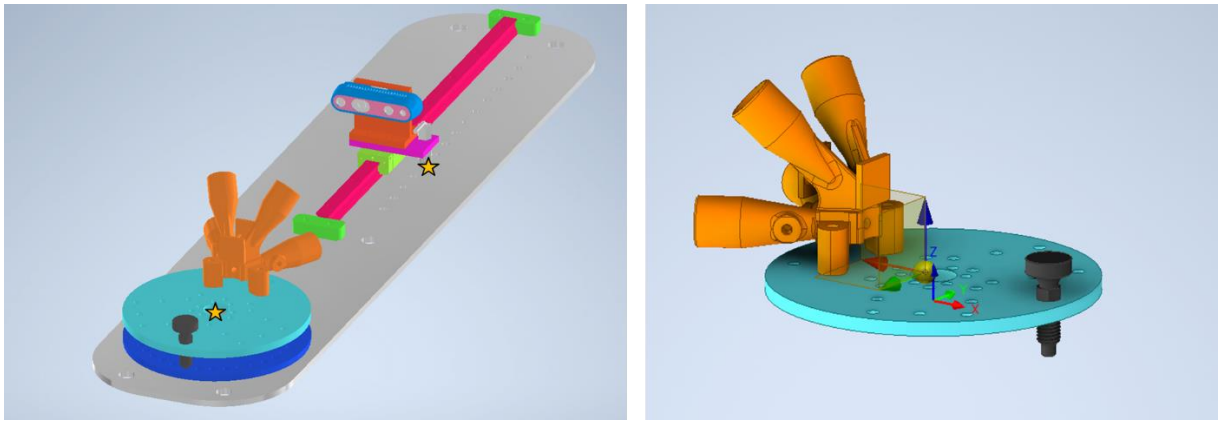

Figure 9 Turntable Reference Frame

Table 1 Transformations between Test Fixture Elements

Reference Points	X (m)	Y (m)	Z (m)
Eyepoint to Camera Carriage	-0.004249	-0.03860486	-0.0798068
Camera Carriage to Turntable	0.036749	0.321749 + 0.0254p	0.02396
Turntable to ArUco Marker Tag 0	0.0806$sin(\theta)$	-0.0806$cos(\theta)$	0.03967814
Turntable to ArUco Marker Tag 1	0.0381$sin(\theta)$	-0.0381$cos(\theta)$	0.03967814
Eyepoint to ArUco Marker Tag 0	0.0325 + 0.0806$sin(\theta)$	0.28314414 + 0.0254p - 0.0806$cos(\theta)$	−0.01616866
Eyepoint to ArUco Marker Tag 1	0.0325 + 0.0381$sin(\theta)$	0.28314414 + 0.0254p - 0.0381$cos(\theta)$	−0.01616866

III.Results

A. Preliminary Results

The pose error of the simulation environment and the PASS hardware was extracted from ArUco for 270 configurations of eyepoint distances and orientations. The complete results showing the absolute translation and rotation error of the simulation and hardware data against the expected data can be seen in Figure 12-Figure 23 in the Appendix. The absolute error listed along the vertical axis of each graph represents the absolute value of the difference between the expected pose of the marker based on the CAD of the test fixture and the processed pose data from the ArUco software. A high-precision threshold line is depicted on each graph to show how the absolute error data compares to the maximum allowable error for high-precision operations – 0.005 m translationally and 0.0873 rad rotationally. Minimum and maximum rotational and translational values are highlighted in green and red on each graph, respectively. Examples comparing hardware and simulation images can be seen in Figure 10 and Figure 11 below. A summary of the maximum and minimum absolute translation and rotation errors for both Tag 0 1 across the different configurations can be seen in Table 2 below. Note that the axes listed in Table 2 and the figures within the Appendix refer to those in the global Gazebo reference frame, as shown in Figure 6.

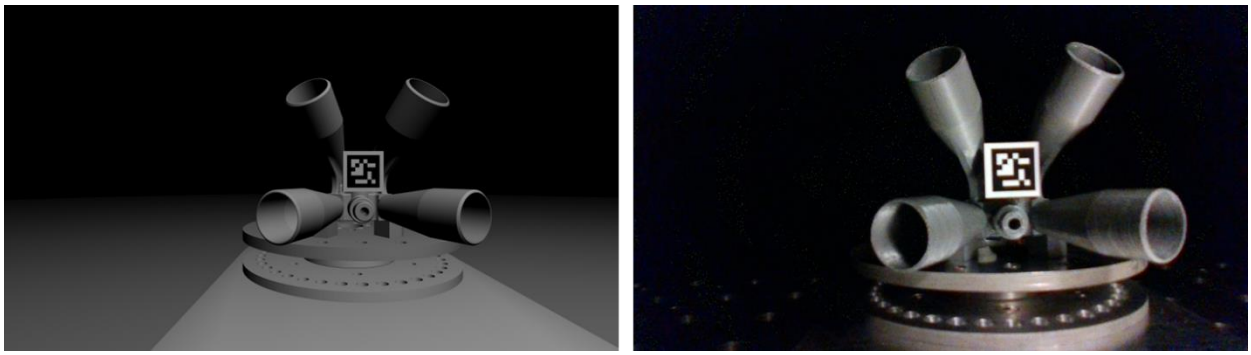

Figure 10 Tag 0 with camera at 0 m/0 rad from Reference Point

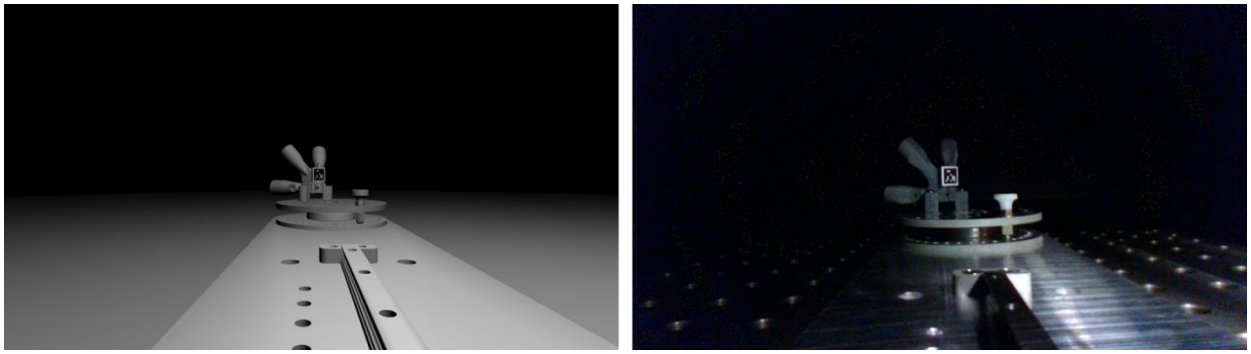

Figure 11 Tag 1 with camera at 0.254 m/3.665 rad from Reference Point

Table 2 Pose Error Measured

			Maximum Error Values		Minimum Error Values	
Axis	**Unit**	**Tag ID**	**Simulation**	**Hardware**	**Simulation**	**Hardware**
Tx	m	0	0.0065	0.0135	0.0000	0.0042
		1	0.0036	0.0152	0.0000	0.0062
Ty		0	0.0435	0.0318	0.0001	0.0074
		1	0.0593	0.0859	0.0004	0.0051
Tz		0	0.0087	0.0079	0.0071	0.0032
		1	0.0088	0.0071	0.0076	0.0031
Rx	rad	0	0.0243	0.0112	0.0000	0.0001
		1	0.0181	0.0173	0.0000	0.0001
Ry		0	0.0779	0.0410	0.0000	0.0093
		1	0.0662	0.0672	0.0001	0.0003
Rz		0	0.6694	0.2911	0.0002	0.0000
		1	1.3370	0.8592	0.0006	0.0001

The errors calculated along each axis varied depending on the distance and orientation of the RGB lens relative to the ArUco markers. Translationally along the X-axis for both Tag 0 and 1 (See Figure **12** and Figure 18), the simulation errors were generally within the acceptable range to support high-precision operations. The hardware errors fell outside that range with a general upward trend as the eyepoint got farther from the ArUco marker. For both the simulation and hardware, error generally increased as the turntable was rotated 0.349 rad or more. This increase in error as the eyepoint got farther away or as the marker was oriented at a greater angle makes sense, as camera image resolution decreases the farther you get from an object. Because the hardware errors showed a consistent linear trend, this might be something that could easily be corrected for within an autonomous framework that calculates real-time command decisions based on pose data.

Translation errors along the Y-axis for Tags 0 and 1 (See Figure 13 and Figure 19), which aligns with the track of the camera carriage, were significantly larger for both the simulation and hardware data. As the eyepoint distance from the marker increased, or as the turntable angle from the center line increased, the absolute error increased. As with the X-axis data, this is to be expected given the degradation of image resolution as the eyepoint gets farther away. However, the significant errors along this axis may be due to calibration issues with the D435i model in Gazebo as well as with the hardware. Additional testing would need to occur to understand if this trend is common when analyzing new data sets.

Translation errors along the Z-axis for Tags 0 and 1 (See Figure 14 and Figure 20) were just above the high-precision threshold for both the simulation and hardware. The consistent trendline of error for both sets of data could be corrected for within an autonomous framework. The only strange trend observed, though, was that the absolute error of the hardware data decreased as the eyepoint got further from the marker. This is not consistent with the data collected for the X and Y axes. More experimentation is needed to understand why this trend is apparent.

Rotational errors along the X-axis for Tags 0 and 1 (See Figure 15 and Figure 18) were well within the threshold for high-precision operations. Errors did not exceed 0.0243 rad for either the simulation or the hardware data, with very flat trendlines. The angle of the turntable had a negligible effect on the measured error. Similar trends can be seen for the rotational error along the Y-axis for both makers (See Figure 16 and Figure 22), with errors not exceeding 0.0779 rad. For these data sets, the error did increase slightly as the eyepoint distance increased from the marker, but still well within the threshold for high-precision operations.

Rotational errors along the Z-axis for Tags 0 and 1 (See Figure 17 and Figure 23) were significantly greater for both the simulation and hardware data sets. In general, the error increased as the eyepoint distance and the turntable angle increased, as expected. In some cases, a greater amount of error was measured for turntable angles closer to 0 rad, which was unexpected. Several of the images processed in ArUco did not return data for the Z-axis, leaving holes in the data sets for analysis. Overall, the Z-rotation data from this experiment is unreliable for assessing high-precision operations, and new experiments would need to be conducted to verify these strange trends.

Accounting for these inaccuracies within the autonomous robotic control frameworks may be necessary for these systems to complete their tasks reliably. Understanding the error trends could affect what robotic operations are tested in the simulation environment for the final PASS demonstrations, as there may be critical failure points at certain orientations of the sensors relative to the tri-truss nodes. This may also help other ISAM technology developers identify limitations in testing scope for their hardware configurations and adjust their testing and operational plans accordingly.

B. Discussion

Some of the things that may have affected the results include improper lens distortion modeling, faulty calibration files, lighting discrepancies, material rendering in the simulation environment, machining errors on the test fixture, or possibly inaccuracies with the pose processing algorithms in ArUco. According to the D400 documentation, the image focus threshold for the depth module on the D435i is 0.1 m, meaning any images taken at closer distances than this may be fuzzy. While this threshold is not explicitly called out for the RGB module, it may still affect the clarity of the images taken.

All these potential issues would need to be understood and mitigated to capture more reliable pose data. The team intends to run additional experiments to extract multiple data sets for the same eyepoint configurations to rule out any outlying data points seen on the graphs.

IV. Conclusions

It is apparent that there are many eyepoint configurations in which the simulation model achieves accuracy levels acceptable for the high-precision ISAM activities required for iSAT; however, there are still configurations in which both the hardware and simulation sensor data show accuracy levels that are less than ideal. Pose error changes based on eyepoint distance and orientation. These inaccuracies can be applied as error models within the robotic control system algorithms, tested in the simulation environment to understand how the autonomous system behaves accounting for the error, then tested with real hardware once confidence has been built in Gazebo.

There are methods by which the simulation and the hardware test could be improved to better reflect the final operational environment – whether that be in orbit or on a planetary surface – which may affect the pose data returned. A series of experiments and a Monte Carlo analysis should be performed to understand how consistent the sensor pose data is from the simulation environment compared to the hardware. The team will be applying the lessons learned from this project to enhance modeling fidelity and simulation capabilities for several ongoing NASA ISAM projects, including PASS, HI-C LITE, Lightweight Surface Manipulation System (LSMS) AutoNomy capabilities Development for surface Operations and construction (LANDO) [20], Tall Lunar Tower (TLT) [21], and MoonTycoon [22]. Because Gazebo/ROS is a widely used framework in the greater ISAM community, eventually the team wants to apply these lessons to other ISAM project collaborations. Assessments of other modeling and simulation environments for ISAM applications may occur in the future as well, including MuJoCo [23], the Marshall Vehicle Representation In C [24], Trick [25], and the Dynamics And Real-Time Simulation [26].

ISAM technology developers must have reliable methods for verifying and validating their systems before launch, especially for autonomous, high-precision operations that rely upon sensor feedback like those that will be used to assemble the iSAT. Using a trusted simulation environment for initial testing will lead to more robust and reliable technology designs. The accuracy of the optical sensor data returned from the Gazebo/ROS environment proves promising for evaluating high-precision operations and should be tested further as the simulation fidelity is improved.

V. References

[1] R. Mukherjee, N. Siegler and H. Thronson, "The Future of Space Astronomy will be Built: Results from the In-Space Astronomical Telescope (iSAT) Assembly Design Study," in *70th Internation Astronautical Congress*, Washington DC, 2019.

[2] NASA, "Gateway," 10 June 2022. [Online]. Available: https://www.nasa.gov/gateway.

[3] I. M. Wong, E. J. Siochi, M. L. Grande, R. W. Moses, W. J. Waltz, S. R. Silbernagel, E. G. Hayward and M. E. Barkhurst, "Design Analysis for Lunar Safe Haven Concepts," in *AIAA SciTech Forum 2022*, San Diego, 2022. doi: 10.2514/6.2022-1567.

[4] R. A. Leslie, D. W. geyer, K. Cunningham, P. C. Glaab, P. S. Kenney and M. M. Madden, "LaSRS++ An Object-Oriented Framewor for Real-time Simulation of Aircraft," 1998.

[5] B. N. Kelley and M. Vaughan, "A Distributed Simulation-to-Flight Framework to Support Investigating Trust/Trustworthiness in Multi-Agent Systems," in *AIAA Scitech 2021 Forum*, Virtual, 2021.

[6] "Gazebo," Gazebo, [Online]. Available: gazebosim.org. [Accessed 2021].

[7] J. Friz, N. Perreau, I. Wong, J. Neuhaus, G. Zimmerman and I. Gomez, "On-orbit/On-surface Servicing, Assembly, and Manufacturing (OSAM) Architecture Simulation System (OASiS)," in *AIAA ASCEND 2022*, Las Vegas, 2022.

[8] L. Santan, M. Maximo, L. Goes and I. M. da Fonseca, "Development of a High Fidelity Space Robot Gazebo-Based Simulator for Space Close-Proximity Operations Experiments," in *26th International Congress of Mechanical Engineering*, Florianopolis, 2021.

[9] W. Qian, Z. Xia, J. Xiong, Y. Gan, Y. Guo, S. Weng, H. Deng, Y. Hu and J. Zhang, "Manipulation Task Simulation using ROS and Gazebo," in *2014 IEEE International Conference on Robotics and Biomimetics*, Bali, 2014.

[10] K. Chow, N. Oune, J. Singh, D. Thomlinson and T. Battista, "Simulating Spacecraft Docking and Berthing Using ROS and Gazebo," in *AIAA SCITECH 2022 Forum*, San Diego, 2022.

[11] R. Corria and R. Ventura, "Payload transportation in microgravity with single and multiple cooperative free-flyer robots," in *2021 IEEE International Conference on Autonomous Robot Systems and Competitions*, Santa Maria da Feira, 2021.

[12] A. Saglam and Y. Papelis, "Scalability of Sensor Simulation in ROS-Gazebo Platform with and without Using GPU," in *2020 IEEE Spring Simulation Conference*, Fairfax, 2020.

[13] "ROS - Robotic Operating System," Open Robotics, [Online]. Available: https://www.ros.org/. [Accessed 2022].

[14] B. N. Kelley, J. R. Cooper, J. Puid-Navarro, M. Vaughan, W. J. Waltz, B. D. Allen, W. Doggett, T. V. Avila, A. K. McQuarry, S. A. Shazly, R. M. Slick and R. A. Williams, "Designing a Software Architecture for the Precision Assembly of Space Structures," in *AIAA Scitech 2022 Forum*, San Diego, 2022.

[15] M. Hammond, A. Dempsy, W. Ward, S. Stewart, J. H. Neilan, J. S. Friz, C. Lamuta and V. Cichella, "A Hybrid Soft Material Robotic End-Effector for Reversible In-Space Assembly of Strut Components," *Frontiers in Robotics and AI, Space Robotics*, 2023.

[16] R. Muñoz Salinas, "ArUco: An efficient library for detection of planar markers and camera pose estimation," [Online]. Available: https://docs.google.com/document/d/1QU9KoBtjSM2kF6ITOjQ76xqL7H0TEtXriJX5kwi9Kgc/edit#heading=h.z7pnof1q4bst. [Accessed 2021].

[17] "OpenCV," [Online]. Available: https://opencv.org/. [Accessed 2021].

[18] "Detection of ArUco Boards," OpenCV, [Online]. Available: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html. [Accessed 2022].

[19] "Intel RealSense D400 Series Product Family," Intel, January 2019. [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/Intel-RealSense-D400-Series-Datasheet.pdf.

[20] J. Cline, M. Vaughan, W. J. Waltz and I. M. Wong, "LANDO: Developing Autonoous Operations for Planetary Surfaces," in *AIAA SCITECH 2022 Forum*, San Diego, 2022.

[21] M. K. Mahlin, "Tall Lunar Tower," in *Lunar Surface Innovation Consortium - Fall Meeting*, El Paso, 2022.

[22] J. Cawley, "NASA Project Takes Off with New 3D Lunar Simulation," NASA, 22 March 2022. [Online]. Available: https://www.nasa.gov/feature/nasa-project-takes-off-with-new-3d-lunar-simulation.

[23] "MuJoCo: Advanced physics simulation," DeepMind Technologies Limited, 2021. [Online]. Available: https://mujoco.org/.

[24] J. S. Orr, J. H. Wall, T. S. VanZwieten and C. E. Hall, "Space Launch System Ascent Flight Control Design," in *2014 American Astronautical Society (AAS) Guidance, Navigation, and Control Conference*, Breckenridge, 2014.

[25] J. M. Penn and A. S. Lin, "The Trick Simulation Toolkit: A NASA/Open source Framework for Running Time Based Physics Models," in *AIAA Modeling and Simulation Technologies Conference (SciTech 2016)*, San Diego, 2016. doi: 10.2514/6.2016-1187.

[26] A. Wall and D. Martin, "The DARTS Simulation Laboratory," NASA JPL, [Online]. Available: https://www-robotics.jpl.nasa.gov/how-we-do-it/facilities/the-darts-simulation-laboratory/.

VI. Appendix

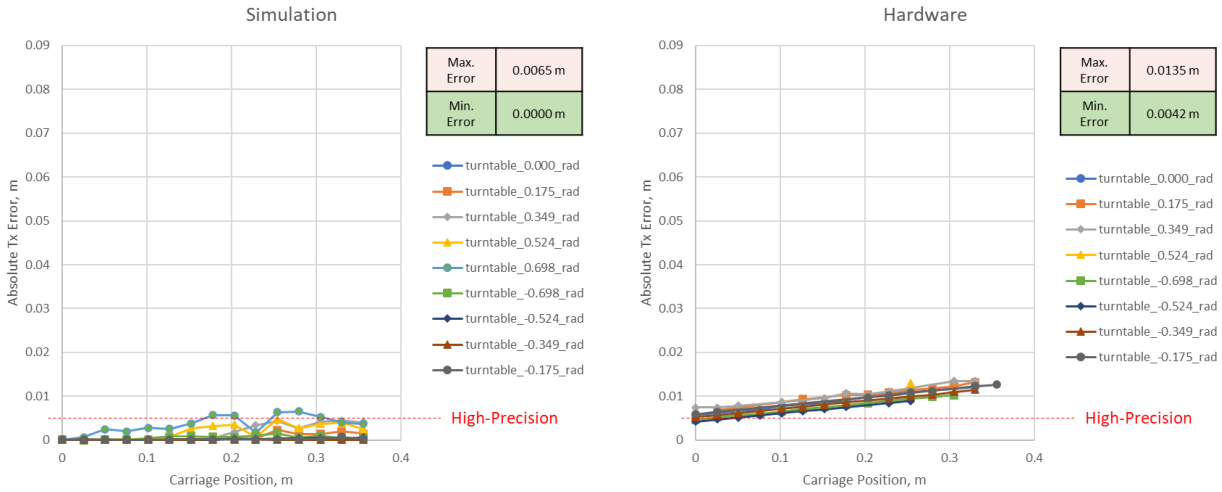

Figure 12 **Translation Error Along X-Axis for Tag 0**

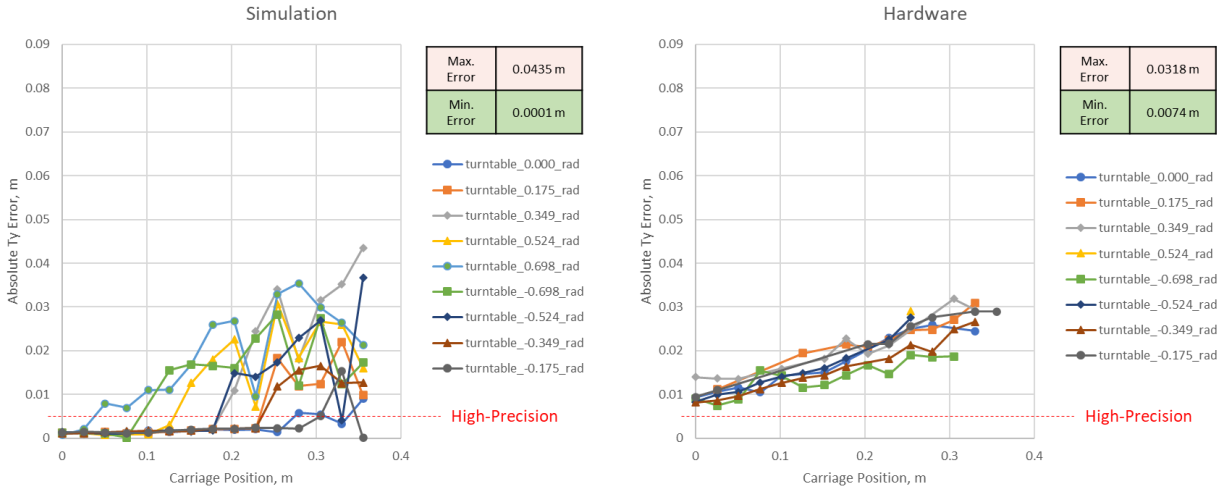

Figure 13 Translation Error Along Y-Axis for Tag 0

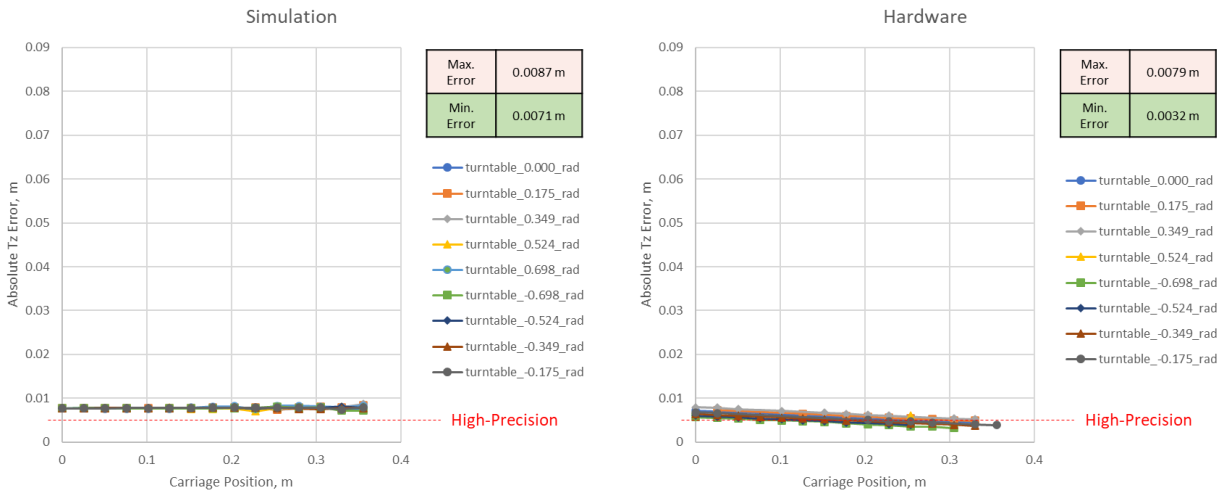

Figure 14 Translation Error Along Z-Axis for Tag 0

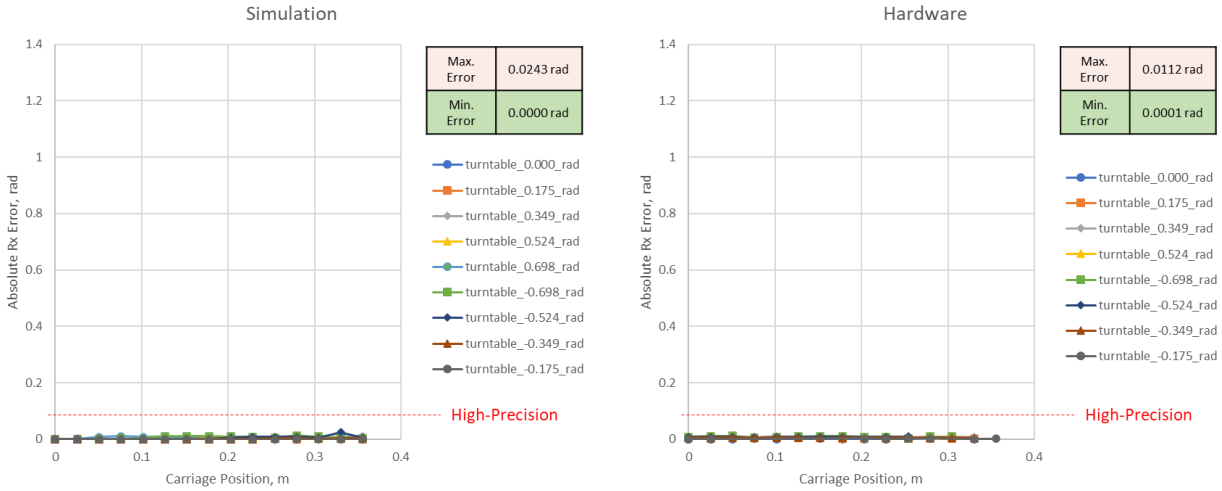

Figure 15 Rotation Error Along X-Axis for Tag 0

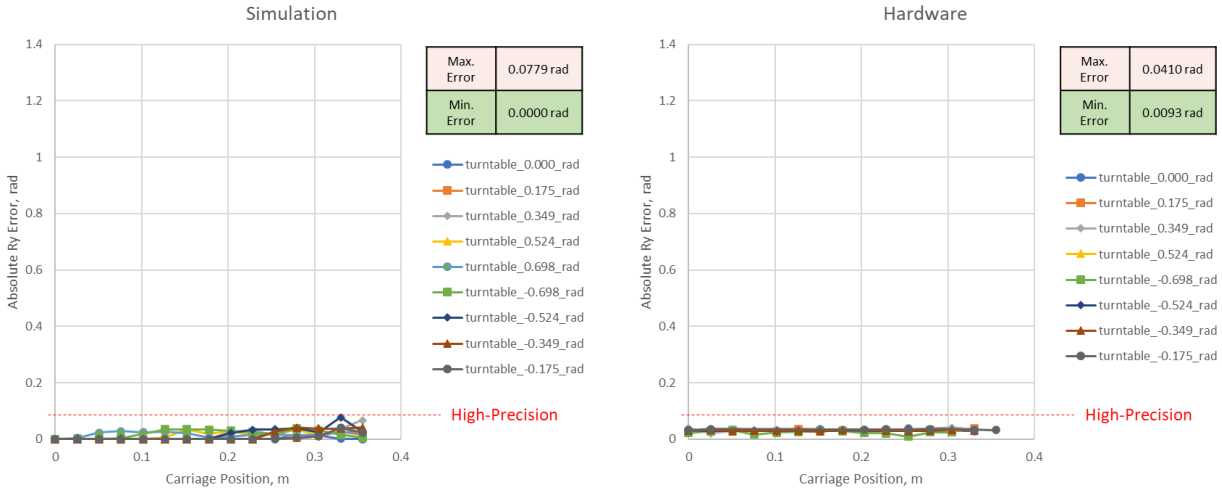

Figure 16 Rotation Error Along Y-Axis for Tag 0

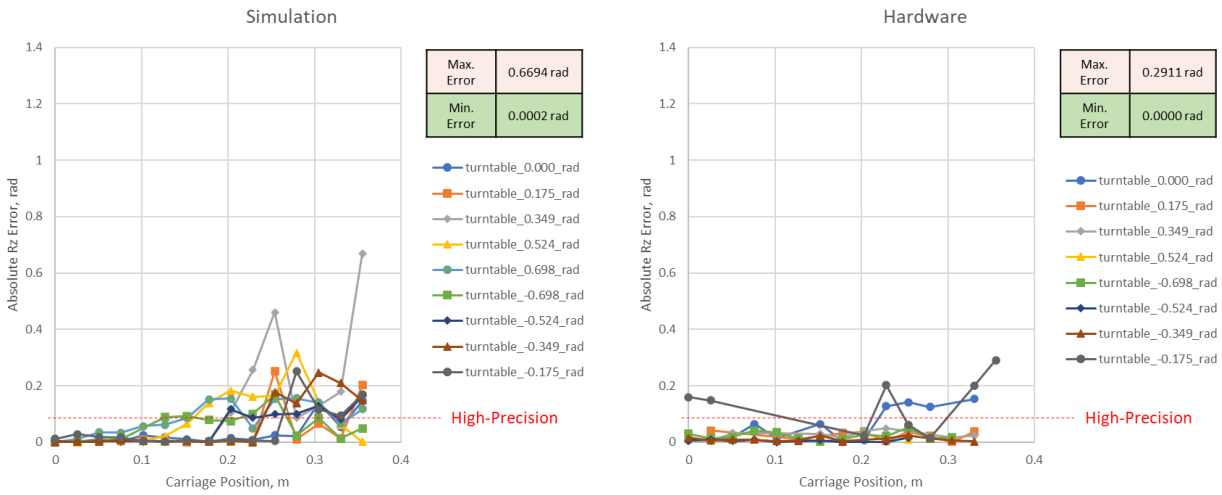

Figure 17 Rotation Error Along Z-Axis for Tag 0

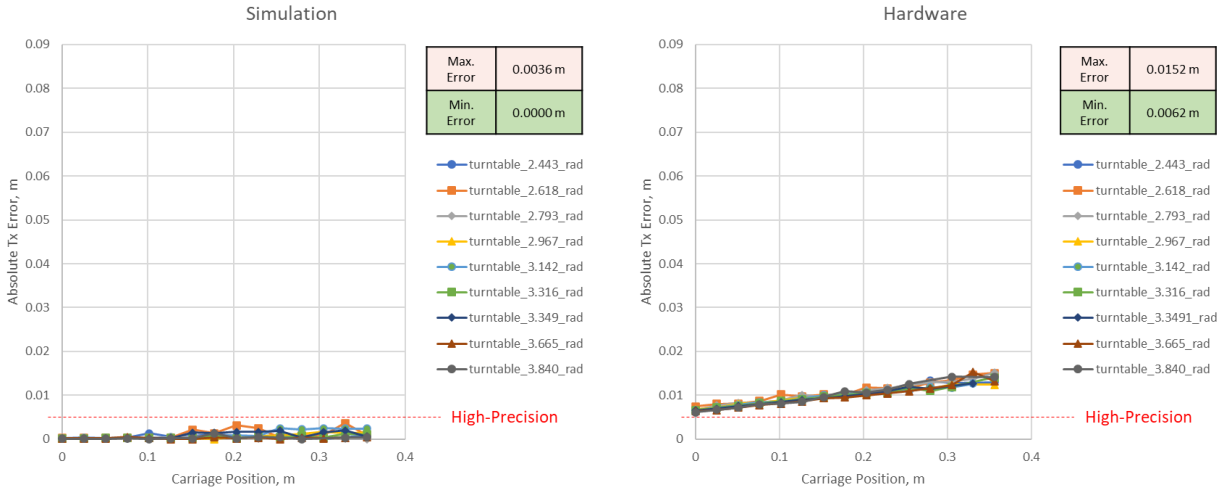

Figure 18 Translation Error Along X-Axis for Tag 1

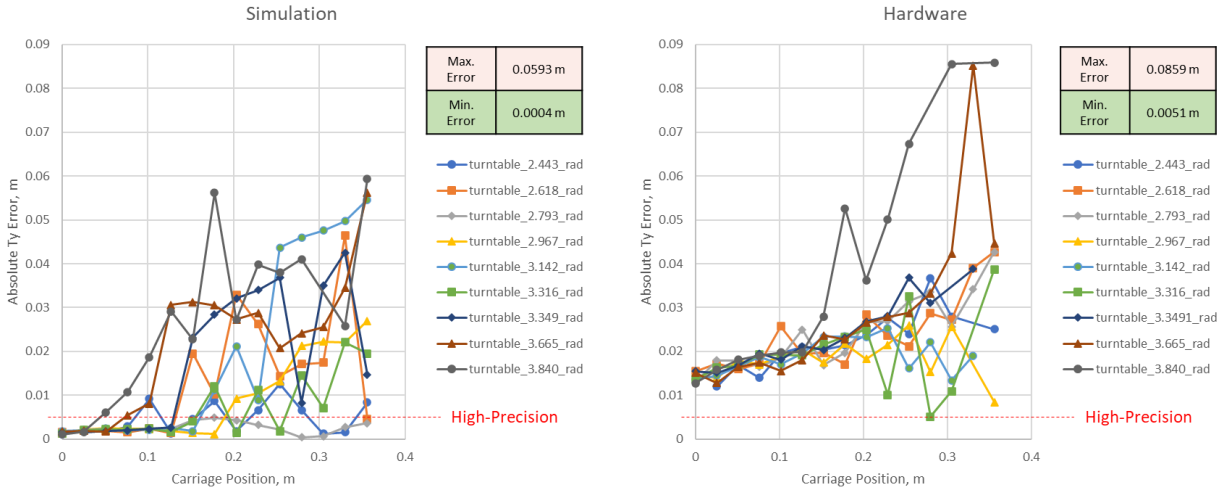

Figure 19 Translation Error Along Y-Axis for Tag 1

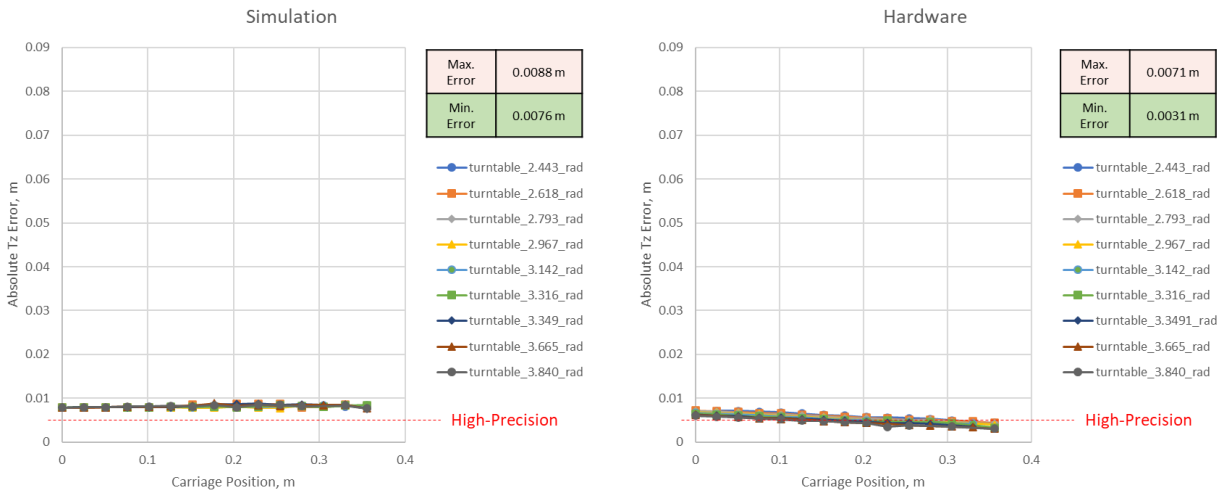

Figure 20 Translation Error Along Z-Axis for Tag 1

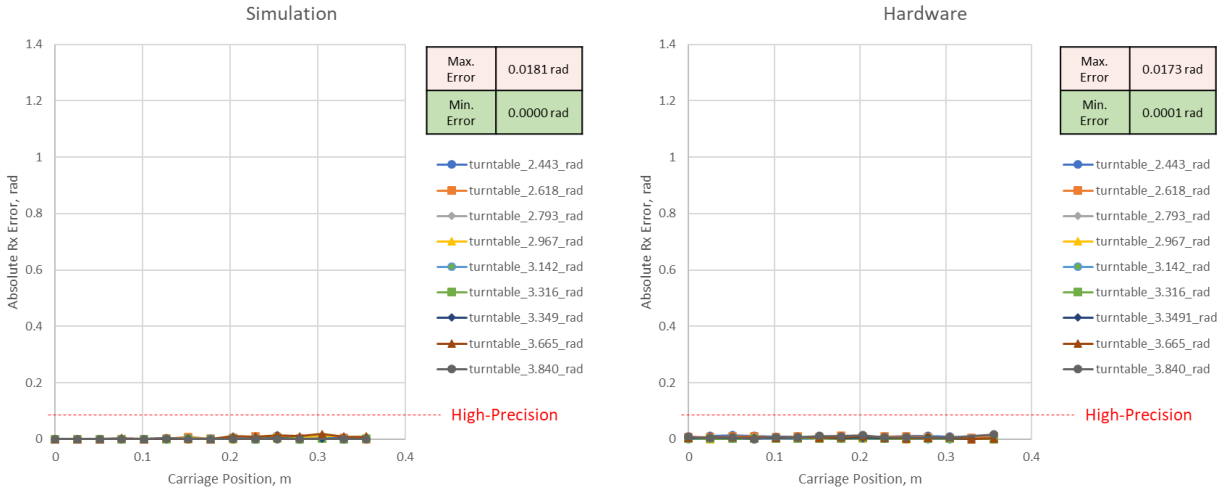

Figure 21 Rotation Error Along X-Axis for Tag 1

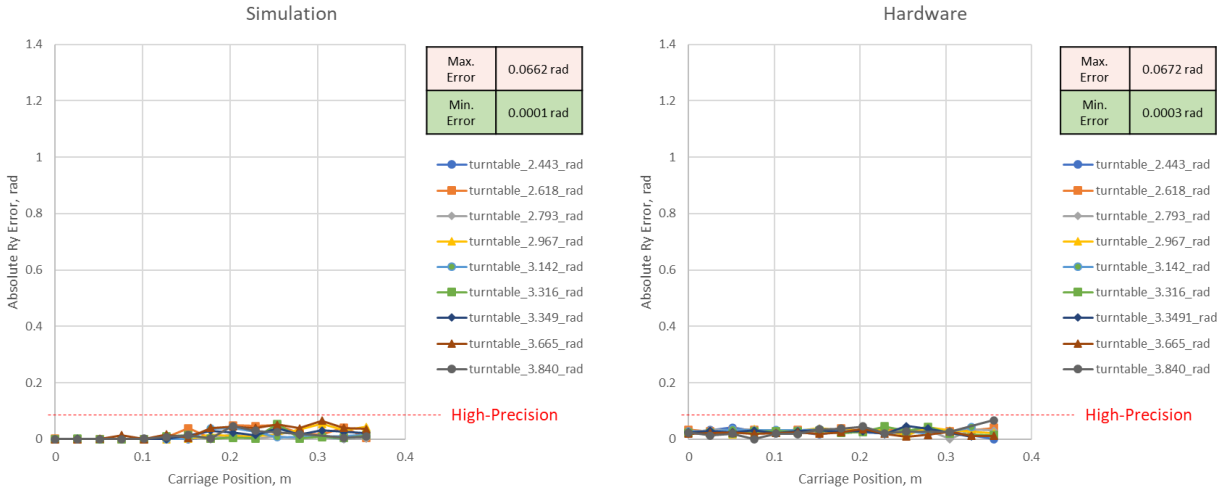

Figure 22 Rotation Error Along Y-Axis for Tag 1

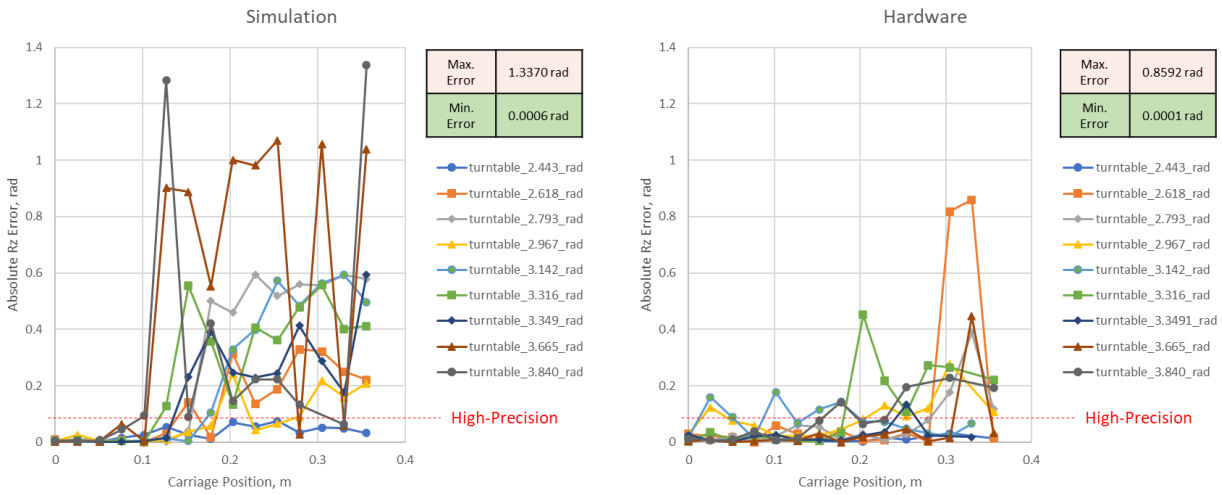

Figure 23 Rotation Error Along Z-Axis for Tag 1