An Assessment of 3D Tracking Systems and Lidar Data for RPO Simulation

Tallak E. Meland

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Aerospace Engineering

Kevin K. Schroeder, Chair
Jonathan T. Black
Samantha P. Kenyon

10 August 2023
Blacksburg, Virginia

Keywords: Lidar, RPO, Local Tracking Systems, Hardware-in-the-Loop, UAV, Point Cloud, ROS, Virtual Reality

Copyright 2023, Tallak E. Meland
An Assessment of 3D Tracking Systems and Lidar Data for RPO Simulation

Tallak Meland

ABSTRACT

This thesis aimed to develop a rendezvous and proximity operation simulation to be tested with physical sensors and hardware, in order to assess the fidelity and performance of low-cost off-the-shelf systems for a hardware-in-the-loop testbed. With the push towards complex autonomous rendezvous missions, a low barrier to entry spacecraft simulator platform allows researchers to test and validate robotics systems, sensors, and algorithms for space applications, without investing in multimillion dollar equipment. This thesis conducted drone flights that followed a representative rendezvous trajectory while collecting lidar data of a target spacecraft model with a lidar sensor affixed to the drone. A relative orbital motion simulation tool was developed to create trajectories of varying orbits and initial conditions, and a representative trajectory was selected for use in drone flights. Two 3D tracking systems, OptiTrack and Vive, were assessed during these flights. OptiTrack is a high-cost state-of-the-art motion capture system that performs pose estimation by tracking reflective markers on a target in the tracking area. Vive is a lower-cost tracking system whose base stations emit lasers for its tracker to detect. Data collection by two lidar types was also assessed during these flights: real lidar data from a physical sensor, and virtual lidar data from a virtual sensor in a virtual environment. Drone flights were therefore performed in these four configurations of tracking system and lidar type, to directly compare the performance of higher-cost configurations with lower-cost configurations. The errors between the tracked drone position time history and the target
position time history were analyzed, and the low-cost Vive and real lidar configuration was demonstrated to provide comparable error to the OptiTrack and real lidar configuration because of the dominance of the drone controller error over the tracking system error. In addition, lidar data of a target satellite model was collected by real and virtual lidar sensors during these flights, and point clouds were successfully generated. The resulting point clouds were compared by visualizing the data and noting the characteristics of real lidar data and its error, and how it compared to idealized virtual lidar data of a virtual target satellite model. The resulting real-world data characteristics were found to be modellable which can then be used for more robust simulation development within virtual reality. These results demonstrated that low-cost and open-source hardware and software provide satisfactory results for simulating this kind of spacecraft mission and capturing useful and usable data.
GENERAL AUDIENCE ABSTRACT

As space missions become more complex, there is a need for lower-cost, more accessible spacecraft simulation platforms that can test and validate hardware and software on the ground for a space-based mission. In this thesis, two position tracking systems and two lidar data collection types were assessed to see if the performance of a low-cost tracking system was comparable to a high-cost tracking system for a space-based simulation. The tracking systems tested were the high-cost state-of-the-art OptiTrack system and the low-cost Vive system. The two types of lidar data collected were real lidar from a physical sensor and virtual lidar from a virtual sensor. These assessments were performed in four configurations, to test each configuration of tracking system and lidar type. First, a simulation tool was developed to simulate the orbital dynamics of a spacecraft that operates in proximity to another spacecraft. After choosing an orbit and initial conditions that represent one such potential mission, the resulting trajectory was uploaded to a drone which acted as a surrogate for a spacecraft, and it flew the uploaded route around a model satellite, collecting lidar data in the process with a lidar sensor affixed to the drone. The tracking systems provided the drone with its position data, and the lidar sensor on the drone collected lidar data of a model satellite as it flew. The data revealed that the low-cost tracking system performance was comparable to the high-cost tracking system because the drone’s controller error dominated over the tracking system errors. Additionally, the low-cost drone and physical lidar sensor generated high quality point cloud data that captured
the geometry of the target satellite and illustrated the characteristics of real-world lidar data and its errors. These results demonstrated that low-cost and open-source hardware and software provide satisfactory results for simulating this kind of spacecraft mission and capturing useful and usable data.
# Contents

List of Figures .......................................................................................................................... viii
List of Tables .............................................................................................................................. x
Nomenclature ............................................................................................................................. xi

1. Introduction ......................................................................................................................... 1
   1.1. Background and Motivation ......................................................................................... 1
   1.2. High Fidelity RPOD Simulation Platforms ................................................................. 2
   1.3. SpaceDrones .................................................................................................................. 5
   1.4. Position Tracking Systems ......................................................................................... 8
   1.5. Thesis Organization ..................................................................................................... 10
   1.6. Research Questions ..................................................................................................... 10

2. Literature Review and Theory ........................................................................................... 12
   2.1. Rendezvous and Proximity Operations ..................................................................... 12
   2.2. Lidar ............................................................................................................................ 14
   2.3. Astrodynamics Equations of Motion ........................................................................ 19

3. Simulations ........................................................................................................................... 25

4. Flights and Data Collection ............................................................................................... 30
   4.1. Experimental Setup .................................................................................................... 30
   4.2. Quadrant 1 .................................................................................................................. 36
   4.3. Quadrant 2 .................................................................................................................. 37
   4.4. Quadrant 3 .................................................................................................................. 38
   4.5. Quadrant 4 .................................................................................................................. 38

5. Results and Discussion ....................................................................................................... 39
5.1. Tracking System................................................................. 39
5.2. Lidar Data................................................................. 46
5.3. Future Work ............................................................. 54

6. Conclusions........................................................................ 55

Bibliography ........................................................................ 57

Appendix.............................................................................. 60
List of Figures

Figure 1: The Northrop Grumman RPOD simulator [12] .......................................................... 3

Figure 2: The EPOS 2.0 simulator [15] .................................................................................. 5

Figure 3: The components involved in SpaceDrones simulations, which may be physical or virtual ...................................................................................................................................... 7

Figure 4: Communication architecture between OptiTrack 3D position tracking, Raspberry Pi, and ground station [9] ........................................................................................................................................ 8

Figure 5: An example of an OptiTrack setup with four cameras ............................................. 9

Figure 6: A Vive base station (left) and tracker (right) [16] .................................................... 10

Figure 7: Main phases of a rendezvous mission [17] ............................................................... 13

Figure 8: Planar, inertial frame of target and chaser and target-centered rotating frame [17] ................................................................................................................................................. 14

Figure 9: Relative orbital motion between a chaser and target spacecraft ............................ 20

Figure 10: Relative orbit simulation as solved by NERM, LERM, and HCW techniques 27

Figure 11: The waypoints of a drone flight, shown as sample points of the simulated orbital motion trajectory .................................................................................................................. 28

Figure 12: CAD model of the satellite used for lidar data collection ................................ 30

Figure 13: L515 Lidar Camera [28] .......................................................................................... 31

Figure 14: Six-rotor drone chassis outfitted with lidar camera and Vive tracker on the tracking fixture on top [7] ...................................................................................................................... 32

Figure 15: The OptiTrack setup with camera arrangement (left) and Vive setup with base station arrangement (right) around the flight area ................................................................................. 33

Figure 16: The virtual environment, with a virtual satellite, and a virtual drone and lidar sensor ................................................................................................................................................ 34
Figure 17: Complete measured position time history ................................................................. 39

Figure 18: Measured position and desired position time histories during flight trajectory
............................................................................................................................................. 41

Figure 19: x, y, z vs sample number for each quadrant ............................................................. 42

Figure 20: Average absolute error and standard deviation in x, y, z, and the position
vector norm for each quadrant .................................................................................................... 43

Figure 21: Average absolute error, and error standard deviation, of position vector norm
for each quadrant ........................................................................................................................ 44

Figure 22: Point cloud from Vive & Virtual Lidar configuration .............................................. 48

Figure 23: Point cloud from one trial of Vive & Real Lidar configuration ............................... 49

Figure 24: Point cloud from OptiTrack & Virtual Lidar configuration ................................. 50

Figure 25: Point cloud from OptiTrack & Real Lidar configuration ........................................ 51

Figure 26: Characteristics of physical lidar and physical satellite point clouds ..................... 53

Figure 27: Vive setup with base stations elevated around flight area, physical satellite
model located at tracking system origin, and physical drone and lidar sensor
circumnavigating the satellite .................................................................................................. 60

Figure 28: Physical satellite model used for data collection .................................................... 61

Figure 29: OptiTrack cameras set up around the flight area, with virtual (CAD) satellite
being scanned by virtual lidar sensor in Unreal Engine, and drone circumnavigating the
origin where the virtual satellite is placed ............................................................................. 61
List of Tables

Table 1: Quadrants of each tested configuration .......................................................... 34
Table 2: All flight parameters used in experiment setup .............................................. 35
Table 3: Flight dataset sizes and durations .................................................................. 46
Nomenclature

$x, y, z$ Coordinates with respect to tracking system origin
$a$ Semi-major axis
$b$ Semi-minor axis
$e$ Eccentricity
$\nu$ True anomaly
$\mu$ Standard Gravitational Parameter
$C$ Minimum value for min-max scaling data
$D$ Maximum value for min-max scaling data
$t_{\text{wait}}$ Wait time at each waypoint
$n$ Mean orbital motion
$\vec{\omega}^{ei}$ Mean orbital motion vector
$\vec{a}_g$ Gravitational acceleration vector
$\hat{e}$ Body frame
$\hat{i}$ Inertial frame
$\delta r$ Radial distance
$\delta \theta$ Azimuth
$\delta z$ Out-of-plane distance
$r_0$ Target spacecraft position vector
$r$ Chaser spacecraft position vector
$\delta r$ Relative position vector of chaser spacecraft
$\delta \vec{v}$ Relative velocity vector of chaser spacecraft
1. Introduction

1.1. Background and Motivation

Spacecraft simulation platforms offer the ability to test and validate the sensors, controls, and relative navigation of a rendezvous and proximity operation (RPO) mission to a robust level. Rendezvous and proximity operations play a critical role in many current and future space missions, especially in formation flying satellite missions. ESA’s Sentinel-1 is one such satellite constellation; it flies in a polar orbit and is equipped with synthetic aperture radar (SAR) sensors [1]. SAR is an active remote sensing technology in which radar is emitted, and the radar signal reflected from the Earth is interpreted to collect earth observation data [2]. SAR data requires a very long sensor antenna to achieve high enough spatial resolution. Because the required antenna lengths are infeasible to implement, a synthetic aperture is implemented [3]. This entails a sequence of data acquisitions from a shorter antenna – from SAR sensors on one or more satellites – and this data is combined to simulate a much larger antenna [3]. The purpose of the Sentinel-1 is Earth observation: its main goals include mapping land surfaces like forest and agriculture, surveilling the marine environment, and mapping for humanitarian aid. Given the nature of SAR data collection, proximity operations play a critical role in this satellite system.

In addition, the TerraSAR-X and TanDEM-X comprise a satellite constellation that fly in a closely controlled formation, at a separation distance of between 250 m and 500 m [4]. This satellite formation collects data to generate a digital elevation model of the Earth. Both satellites act as a large single-pass SAR interferometer [5]. As for a future mission that is under development, the Laser Interferometer Space Antenna (LISA) is a proposed
mission that would detect gravitational waves using laser interferometry between three identical spacecraft. These spacecraft would fly 2.5 million km apart, in a triangular formation in heliocentric orbit [6]. As can be seen from the current satellite constellations in operation, and the more complex designs of future missions, RPO is a significant aspect of space-based missions that requires accessible testing platforms for proper development.

Many spacecraft simulator platforms today are prohibitively expensive due to large-scale high-precision robotics equipment. In order to better engage academics and smaller research organizations in this domain, there is a need for low-cost simulation and hardware-the-loop solutions. Virginia Tech’s SpaceDrones spacecraft simulator platform is an example of such a solution, as it combines autonomous spacecraft research with sensors testing, data collection, high-quality virtual environments, and computer vision [[7], [8], [9], [10], [11]].

1.2. High Fidelity RPOD Simulation Platforms

Currently, there exist many rendezvous, proximity operations, and docking (RPOD) simulator labs for developing and testing missions and the spacecraft involved in these missions. These lab platforms are high-fidelity and expensive since they involve multimillion dollar pieces of equipment, fulltime staff, and specialized facilities to operate. NASA, Northrop Grumman, and the German Aerospace Center are examples of groups that operate such simulator platforms.
Northrop Grumman operates a robust Rendezvous, Proximity Operations, and Docking laboratory for testing its RPOD missions. Its robotic simulation platform has been used for developing and testing its Mission Extension Vehicle (MEV) for RPOD capabilities for its satellite servicing missions. The testing platform uses full scale mock-ups of the target and chaser vehicles to develop and test the RPOD sensors, actuators, and control algorithms that are used in its missions [12]. The simulator includes six degrees of freedom (DOF) robotic arms that mimic the attitude determination and control systems of the spacecraft (Figure 1). They move along tracks to simulate trajectories and maneuvers that will take place in the RPOD mission, such as an approach and holding phase by the chaser vehicle, and semi-random motion by the target vehicle [13]. The simulation platform has sensors such as visible and infrared narrow field-of-view cameras.

![Figure 1: The Northrop Grumman RPOD simulator [12]](image)

Similarly, NASA’s Johnson Space Center operates high-fidelity real-time simulators for developing, testing, and training for RPOD operations, both manned and unmanned [14].
This simulation facility contains their Six-Degree-of-Freedom Dynamic Test System (SDTS) and their Air Bearing Floor (ABF). The SDTS is a real-time 6-DOF short range motion base simulator that simulates the relative dynamics of two bodies in space in an RPOD mission. It can test full scale docking systems. The ABF is a 70 ft × 98 ft epoxy surface used for rendezvous and contact testing when a low friction environment is required [14]. Test articles are placed on perforated pads that distribute a cushion of compressed air between the pads and the floor, preventing contact. This air bearing floor limits RPO testing to 3-DOF since only rotation about one axis can be performed on a flat surface.

Finally, the European Proximity Operations Simulator (EPOS) 2.0 is a robotic hardware-in-the-loop test facility of the German Aerospace Center (DLR) [15]. The facility simulates rendezvous and inspection phases of robotic missions. The platform includes two 6-DOF robot arms, mockups of the target and chaser spacecraft, a 25 meter rail system for moving the robot arms, a variety of optical sensors, and a sun simulator (Figure 2) [15]. The EPOS 2.0 supports testing and validation of sensors; optical guidance, navigation, and control systems; and on-orbit servicing missions.
These state-of-the-art RPOD simulation platforms allow for extremely high-fidelity development and testing. However, they present a very high cost of entry for performing RPOD simulations given the multimillion-dollar price tag of the equipment necessary. This illustrates the need for an accessible testbed for space-based applications. The SpaceDrones platform was created for this purpose by a team at Virginia Tech’s Department of Aerospace and Ocean Engineering. The low-cost research platform allows for spacecraft dynamics research using UAVs. SpaceDrones can be used to emulate RPOs with drones as surrogates for robot arms. While it naturally cannot perform simulations of the same fidelity as the previously described platforms, SpaceDrones achieves a high level of performance with its sensors, 3D tracking systems, and 4- or 6-DOF UAVs.

1.3. SpaceDrones

SpaceDrones is a collaborative laboratory that allows for multidisciplinary research for small satellites and autonomous space applications [8]. The SpaceDrones platform is a low-cost spacecraft simulator that consists of a robotics architecture primarily built on top of
the Robot Operating System (ROS). The ROS framework allows users to send and receive sensor data, object position data, and drone diagnostic data over a local area network. The SpaceDrones platform allows for hardware-in-the-loop research that can be used to develop and test algorithms on the ground for space-based applications. It supports testing of autonomous drone flights with onboard computer vision and machine learning capabilities, as well as 3D tracking capabilities. The drones serve as a surrogate for spacecraft in this research. The SpaceDrones platform can also use virtual environments created in high-quality game engines such as Unreal for its computer vision applications. The drone can be tracked in 3D space, and a virtual version of the drone can be seen flying around a given environment while the physical drone flies around the safe and contained lab space. This allows risky maneuvers to be tested virtually without putting the actual physical drone in danger. Overall, SpaceDrones allows different configurations of physical versus virtual systems to be tested. Figure 3 illustrates this loop of different configurations, in which one can start with all virtual components and continually minimize the reality gap by introducing physical hardware, a physical environment, and a physical sensor.
Figure 3: The components involved in SpaceDrones simulations, which may be physical or virtual.

The SpaceDrones platform consists of a drone, a position tracking system, a Pixhawk flight controller, a Raspberry Pi companion computer, and ROS for internode communication [9]. This relatively low-cost hardware allows for algorithms and systems to be tested on the ground. It is a distributed system which prevents issues with centralized systems such as limited ground station processing capacity, and network bandwidth and latency [9]. Figure 4 displays this communication architecture with the OptiTrack tracking system.
1.4. Position Tracking Systems

This thesis assessed two 3D position tracking systems: OptiTrack and Vive. OptiTrack is a higher-cost tracking system. OptiTrack provides real-time position tracking of any object that is equipped with specialized reflective markers. These markers are small and very lightweight, with a mass of a few grams. The system tracks position and orientation data accurate to 1 mm at up to 240 Hz which provides precision navigation inputs in place of onboard sensors [9]. The lab setup consists of 16 high-speed tracking cameras, models 13 and 13W, which have a resolution of 1280 × 1024. These cameras cost $2499 each as of June 2023, which totals approximately $40,000 for the full lab tracking setup.
Vive is a lower-cost 3D tracking system. It consists of at least one laser-emitting base station and one tracker. The base sweeps the room with laser pulses, and the tracker detects the pulses to solve for its position and orientation relative to the base station. The lab setup consists of four base stations with the tracker affixed to the drone. The tracker has a mass of 75 grams. Each station has a 150° horizontal field of view and a 110° vertical field of view. As of June 2023, each station costs $199 and each tracker costs $129.99, which totals approximately $930 for the full lab tracking setup.

Figure 5: An example of an OptiTrack setup with four cameras
1.5. **Thesis Organization**

The objective of this thesis is to leverage the SpaceDrones platform to simulate an RPO with lidar sensors in different flight configurations that test low and high fidelity systems and minimize the reality gap that exists in a purely virtual simulation. In the following chapter, the literature surrounding RPO, lidar, and the relevant astrodynamics is reviewed. Chapter 3 discusses the orbital dynamics simulations that led to the chosen flight trajectory, chapter 4 outlines the experiment setup and the breakdown of the system configurations for each flight. Chapter 5 then discusses the results of the drone flights and the generated data, and chapter 6 offers conclusions from these experiments and answers the central research questions.

1.6. **Research Questions**

1. Does a low-cost configuration of physical systems and sensors, like those in the SpaceDrones lab, allow for satisfactory position tracking performance in an emulation of an RPO?
2. Can minimizing the reality gap with low-cost systems and a gaming engine in the SpaceDrones platform yield high quality lidar data for ground testing of a space-based application?
2. Literature Review and Theory

2.1. Rendezvous and Proximity Operations

Rendezvous and proximity operations are a major operational technology required for many missions that involve more than one spacecraft. RPO techniques are used in missions such as in-space assembly, space station resupply, in-orbit spacecraft repair, and on-orbit satellite servicing [17].

The first rendezvous and docking between two spacecraft occurred in 1966 when Neil Armstrong and Dave Scott performed a manual rendezvous in a Gemini vehicle and docked with an unmanned Agena target vehicle. The first automatic rendezvous and docking operation occurred in 1967 when the Cosmos 186 and 188 vehicles docked [17]. Since then, rendezvous and docking has regularly occurred on programs such as Skylab, Mir, and the ISS.

The rendezvous and docking process involves a series of orbital maneuvers that bring the chaser to the vicinity of the target. The chaser first must be launched and brought to the same orbital plane as the target and then achieve the same orbital height, phase angle, and eccentricity. Figure 7 describes the main phases of an RPO mission.
During the launch and phasing, absolute measurements in an Earth-centered inertial frame are used for navigation. Trajectories during these phases are therefore usually represented in this frame. At the far and close-range rendezvous phases, the chaser and target motion are represented in a relative frame centered at the target. Figure 8 shows the trajectories of the target and the chaser in both the orbital plane frame and the local orbital frame of the target. The chaser trajectory shown is an eccentric orbit with an apogee on the target orbit and a perigee at a distance of $\Delta h$ below it. The orbit direction is labeled as V-bar which points in the direction of the velocity vector. The radial direction is labeled as R-bar.
While phasing maneuvers rely on absolute navigation measurements, navigation during close range rendezvous operations relies on relative measurements of range and direction. It is in this RPO phase that lidar technologies, as discussed in the previous section, play a significant role in developing autonomous RPO capabilities. These capabilities are desired because of the increased complexity of rendezvous missions. For example, new missions are to occur far from Earth where the communication delays are too long for safe contact. An example of a platform being developed for this problem is NASA’s OSAM-1 [18]. This spacecraft aims to extend satellites’ lifespans even if they were not designed for such a service [19].

2.2. Lidar

Light Detection and Ranging (lidar) is an active remote sensing technology that is used for determining ranges of target objects. It is used to find the range to a certain point on a target object by emitting light and measuring the time it takes to return to the sensor. Therefore,
the range to a specific target point is $r = c \cdot t/2$ where $c$ is the speed of light. As a result, lidar is a principal technology used for spacecraft rendezvous missions since the data it provides allows for spacecraft relative navigation between a chaser spacecraft and a target spacecraft [20].

Lidar sensors generally lie in three major categories based on how they sense a 3D scene: scanning, detector arrays, and spatial light modulators. Scanning lidars emit light by sweeping a narrow laser beam across the field of view and sensing the return with one detector. A set of mirrors and lenses directs the return light to the single detector. These lidar sensors are easy to calibrate since they contain one detector. The laser beam can be pointed very precisely because of the system of mirrors and lenses, and the result is very high-resolution and dense point clouds. The downsides to scanning lidars include the wear down of moving parts and the potentially long duration of time for scanning its entire field of view. This long duration can especially affect a navigation application if objects being scanned undergo large relative motion during the scan of a single frame: the resulting point cloud can contain motion blur [20].

Secondly, detector arrays emit a broad laser pulse to illuminate the entire field of view at once. An array of detectors then senses the return of the light. These lidars measure the laser time-of-flight to each pixel on this array. Time of flight can be measured at each pixel with either a flash lidar or a continuous wave lidar. Flash lidars send out a laser pulse and measure the time that it takes for the return pulse to be detected at each pixel. They are the most common configuration used in space applications and can be used at short range and
long range. Since these lidars illuminate the field of view at once, the 3D point cloud is generated at once and largely avoids motion blur. Continuous wave lidars modulate the intensity of the laser and measure the phase difference between the emitter and each pixel on the detector. There are no moving parts in this lidar configuration, but they are harder to calibrate. Because of their range versatility, flash lidars are the preferred configuration for autonomous rendezvous and docking applications [20].

Lastly, spatial light modulators are a newer type of lidar sensor. They emit light in a specific pattern, and a single detector senses the light return. These lidars use compressed sensing algorithms to generate a 3D point cloud from the return signals.

Lidar configurations have been used on many missions and have continued to adapt for newer challenges. Three significant lidars that have been used in space include the Trajectory Control Sensor (TCS), the Videometer (VDM), and the Rendezvous Sensor (RVS) / Telegoniometer (TGM).

The TCS was a scanning lidar that flew on the Space Shuttle between 1995 and 2011. It was used for relative navigation during rendezvous with the Mir Space Station, the Hubble Space Telescope, and the International Space Station. The TCS required reflectors to be placed on its target at known locations. When in operation, the TCS tracked these reflectors and provided the navigation system with range and bearing information to reach them [20].
The VDM is a relative navigation sensor that the ESA used on their Automated Transfer Vehicle (ATV). The VDM collects range, range-rate, and line-of-sight measurements, and provides this data to the ATV GNC algorithms at a rate of 1 Hz from a range of 300 meters to docking. The VDM also can collect relative position and relative attitude data to support docking from 30 meters to contact. This lidar relies on retro-reflector assemblies mounted to the aft end of the ISS Service Module. There are two arrays of reflector assemblies: a triangular layout of three reflector assemblies, and a pyramidal layout of five reflectors. The VDM uses all of these reflector assemblies from long range to midrange. Once the range is under 30 meters, the VDM uses the pyramidal array to find the relative position and attitude for rendezvous [20].

Lastly, the RVS is the primary relative navigation sensor on JAXA’s H-II Transfer Vehicle resupply spacecraft. The RVS is a scanning lidar that operates from 1.5 km to docking. It requires retro-reflectors for its maneuvers. RVS provides range and line of sight angles from the H-II Transfer Vehicle to reflectors on the Japanese Experiment Module on the ISS between a range of approximately 750 meters until docking [20].

In addition to these current lidars that have been used for several missions, new lidar technologies have been flown and tested on missions to analyze their performance. Six flight experiments were performed between 2009 and 2011 where a new generation of lidar technologies was tested. These lidar systems were consequently improved through analysis of their performance. The lidar technologies consisted of the Neptec TriDAR, the ASC
Neptec’s TriDAR (triangulation + LIDAR) is a scanning lidar that was developed as a guidance sensor for automatic rendezvous and docking missions to noncooperative targets. It includes a hybrid 3D sensor along with a model-based tracking algorithm to provide 6-DOF pose (position and orientation) information in real time [21]. TriDAR’s software uses geometric information obtained from successive 3D point clouds to match against the known shape of the target object to calculate its pose. TriDAR tries to extract maximum geometric information at minimal cost to data volume and processing, an approach called more information, less data (MILD) [21]. Significantly, this lidar system does not require reference markers such as reflectors on the target spacecraft [22]. This advantage allows for RPO with targets that were never constructed with markers, and that would be infeasible or impossible to provide with reference markers in the future. Additionally, TriDAR flew on the STS-128, STS-131, and STS-135 missions to the ISS as a technology demonstration. During the test flights, TriDAR successfully acquired and tracked the ISS in real time, providing state vectors and closing rate information to the Space Shuttle crew [21]. TriDAR is now used on Northrop Grumman Innovation Systems’ Cygnus spacecraft cargo resupply missions to the ISS. It guides the final rendezvous phase from a maximum range of about 1 km to final capture position at about 10 meters [23].

The DragonEye is a flash lidar with a 128 × 128 pixel detector array and an operational range of around 1000 meters. SpaceX selected this lidar system for proximity operations
and tested it on STS-127 and STS-133. As with the TriDAR, the DragonEye was a successful targetless solution for proximity operations. This test demonstrated that flash lidars can be used in RPO and docking.

Finally, the VNS Flash Lidar is a flash lidar with a $256 \times 256$ pixel focal plane array and a variable field of illumination of either $12^\circ$ or $20^\circ$. The $12^\circ$ option is typically used at long range while the $20^\circ$ option is typically used at short range. The VNS lidar was tested on STS-134 where lidar images were captured at 30 Hz, and the sensor demonstrated the ability to measure range from 5 km to docking.

### 2.3. Astrodynamics Equations of Motion

The equations of relative motion are of great interest for developing RPO simulations. Spacecraft require a standoff distance during rendezvous maneuvers to avoid collision with the target spacecraft. Depending on the application, this distance must then be closed. In the context of lidar scanning, the spacecraft must approach to within the required range for collecting point cloud data. These maneuvers require relative motion between spacecraft to be controlled. The Hill-Clohessy-Wiltshire (HCW) equations are a solution to the relative motion between two satellites when the target satellite is in a circular orbit around the orbiting body [24]. One satellite is the reference body or target, which is assumed to be in circular orbit, and the other satellite is the chaser (Figure 9).
Figure 9: Relative orbital motion between a chaser and target spacecraft

Here, $\vec{r}_0$ is the target’s position, $\vec{r}$ is the chaser’s position, and $\delta \vec{r}$ is the relative position of the chaser. Cylindrical coordinates ($\delta r$, $\delta \theta$, and $\delta z$) with the target as the origin are used. The components of the vectors are radial ($\hat{e}_r$), in-track ($\hat{e}_\theta$), and out-of-plane ($\hat{e}_z$). Because the $\hat{e}$ frame is tied to the target body, the angular velocity of the $\hat{e}$ frame equals the mean motion of the circular orbit: $\omega^e = n \hat{e}_z$. The chaser’s equation of motion is equation (1), its acceleration vector in this $\hat{e}$ frame is equation (2), and its acceleration vector in the inertial frame is equation (3).

\begin{align*}
\vec{r} &= (r_0 + \delta r) \hat{e}_r + r_0 \delta \theta \hat{e}_\theta + \delta z \hat{e}_z \quad (1) \\
\frac{e^2}{dt^2} \vec{r} &= \delta \dot{r} \hat{e}_r + r_0 \delta \ddot{\theta} \hat{e}_\theta + \delta \dot{z} \hat{e}_z \quad (2)
\end{align*}
\[
\frac{d^2 \vec{r}}{dt^2} = \frac{e d^2}{2} \vec{r} + 2\vec{\omega}^e i \times \frac{e d}{dt} \vec{r} + \vec{\omega}^e i \times (\vec{\omega}^e i \times \vec{r}) \tag{3}
\]

The motion of the two satellites is governed by two body motion; therefore, the chaser’s gravitational acceleration is given by the following:

\[
\ddot{a}_g = -\frac{\mu \vec{r}}{r^3} = -\frac{\mu [(r_0 + \delta r)\hat{r}_r + r_0 \delta \theta \hat{r}_\theta + \delta z \hat{r}_z]}{(r_0^2 + 2r_0 \delta r + \delta r^2 + r_0^2 \delta \theta^2 + \delta z^2)^{3/2}} \tag{4}
\]

If \(\delta r\), \(\delta \theta\), and \(\delta z\) are assumed to be small, their higher order terms can be neglected. Using the binomial theorem and dropping higher order terms, the denominator can therefore be simplified to the following:

\[
(r_0^2 + 2r_0 \delta r + \delta r^2 + r_0^2 \delta \theta^2 + \delta z^2)^{-3/2} \approx r_0^3 \left(1 + 2 \frac{\delta r}{r_0} \right)^{-3/2} \tag{5}
\]

\[
\left(1 + 2 \frac{\delta r}{r_0} \right)^{-3/2} = 1 - \left(\frac{3}{2}\right) \left(2 \frac{\delta r}{r_0} \right) + \cdots \approx 1 - 3 \frac{\delta r}{r_0} \tag{6}
\]

Substituting this approximation in (4), and once again dropping higher order \(\delta\) terms, the result gives the following:

\[
\ddot{a}_g \approx -\frac{\mu r_0 \hat{r}_r}{r_0^3} - \frac{\mu}{r_0^3} (-2\delta r \hat{r}_r + r_0 \delta \theta \hat{r}_\theta + \delta z \hat{r}_z) \tag{7}
\]
By revisiting (3) and substituting known terms, like terms from (3) and (7) can be matched to find the three linear equations of relative motion:

\[
\begin{align*}
\ddot{\delta r} - 2nr_0\delta \dot{\theta} - 3n^2 \delta r &= 0 \\
r_0 \delta \ddot{\theta} + 2n \delta \dot{r} &= 0 \\
\delta \ddot{z} + n^2 \delta z &= 0
\end{align*}
\]

Here, \( n = \sqrt{\mu / r_0^3} \) is the orbital mean motion, with \( \mu \) being the standard gravitational parameter. These equations can be solved using standard ODE solution methods. Their solutions and their derivatives are given by the following:

\[
\begin{align*}
\delta r(t) &= -\left( \frac{2}{n} r_0 \delta \dot{\theta}_0 + 3 \delta r_0 \right) \cos nt + \frac{\delta \dot{r}_0}{n} \sin nt + 4 \delta r_0 + \frac{2}{n} r_0 \delta \dot{\theta}_0 \\
\delta \theta(t) &= \delta \theta_0 - \left( 3 \delta \dot{\theta}_0 + \frac{6n \delta r_0}{r_0} \right) t + \left( \frac{4 \delta \dot{\theta}_0}{n} + \frac{6 \delta r_0}{r_0} \right) \sin nt + \frac{2 \delta \dot{r}_0}{nr_0} \cos nt - \frac{2 \delta \dot{r}_0}{nr_0} \\
\delta z(t) &= \delta z_0 \cos nt + \frac{\delta \dot{z}_0}{n} \sin nt \\
\delta \dot{r}(t) &= \left( 2r_0 \delta \dot{\theta}_0 + 3n \delta r_0 \right) \sin nt + \delta \dot{r}_0 \cos nt
\end{align*}
\]
\[
\delta \dot{\theta}(t) = \left(-3\delta \dot{\theta}_0 - \frac{6n\delta r_0}{r_0}\right) + \left(\frac{6n\delta r_0}{r_0} + 4\delta \dot{\theta}_0\right) \cos nt - \frac{2\delta \dot{r}_0}{r_0} \sin nt \quad (15)
\]

\[
\delta \ddot{z}(t) = -\delta z_0 n \sin nt + \delta \dot{z}_0 \cos nt \quad (16)
\]

From here, the relative position vector can be defined as \(\delta \vec{r}^T = \{\delta r, r_0 \delta \theta, \delta z\}\), and the relative velocity vector can be defined as \(\delta \vec{v}^T = \{\delta \dot{r}, r_0 \delta \dot{\theta}, \delta \dot{z}\}\). The solution can then be placed in the following matrix form:

\[
\begin{bmatrix}
\delta \dddot{r}(t) \\
\delta \ddot{v}(t)
\end{bmatrix} = \Phi \begin{bmatrix}
\delta \dddot{r}_0 \\
\delta \ddot{v}_0
\end{bmatrix} \quad (17)
\]

\[
\Phi = \begin{bmatrix}
\Phi_{rr} & \Phi_{rv} \\
\Phi_{vr} & \Phi_{vv}
\end{bmatrix} \quad (18)
\]

\[
\delta \dddot{r}(t) = [\Phi_{rr}(t)]\delta \dddot{r}_0 + [\Phi_{rv}(t)]\delta \ddot{v}_0 \quad (19)
\]

\[
\delta \ddot{v}(t) = [\Phi_{vr}(t)]\delta \dddot{r}_0 + [\Phi_{vv}(t)]\delta \ddot{v}_0 \quad (20)
\]

Here, the matrix \(\Phi\) is the state transition matrix, which is only a function of time and the mean motion of the target’s orbit. The four matrices within the state transition matrix are given as the following, which forms the complete HCW solution [24]:

\[
\Phi_{rr}(t) = \begin{bmatrix}
4 - 3 \cos nt & 0 & 0 \\
6(\sin nt - nt) & 1 & 0 \\
0 & 0 & \cos nt
\end{bmatrix} \quad (21)
\]
Equation (12) reveals that the in-track direction is not purely periodic since it has a secular term that is linear with respect to time. This result illustrates that $\delta \theta$ can grow unbonded over time since the two satellites may have different orbital periods. If a bounded motion needs to be enforced, this secular term must equal zero, and the correct initial conditions must be in place:

$$3 \delta \dot{\theta}_0 + \frac{6n \delta r_0}{r_0} = 0$$
3. **Simulations**

To begin, a simulation was performed of the astrodynamics of a target and chaser spacecraft in an RPO scenario. A relative orbital motion simulation tool was created to simulate trajectories of varying orbital elements and initial conditions. Trajectories were calculated using nonlinear equations of relative motion (NERM), linear equations of relative motion (LERM), and the Hill-Clohessy-Wiltshire (HCW) equations. The NERM and LERM equations were solved and propagated with a variable-step, variable-order PECE solver of orders 1 to 13. The algorithm is a multistep solver as it requires the solutions at several preceding time points to calculate the current solution [25]. This algorithm may be more efficient than Runge-Kutta methods at strict tolerances, as is the case in orbital dynamics problems. By plotting trajectories as solved by these three methods, the errors among them could be calculated and plotted; this divergence in solutions was used to illustrate when certain orbital assumptions began to break down. The NERM solution was valid for all relative distances and for eccentric target orbits. The LERM solution assumed that the relative distance of the satellites was much smaller than the target's position from the Earth. Finally, the HCW solution assumed both a circular target orbit and small satellite relative distances as compared to the target’s position from the Earth.

A trajectory was then developed that would be used for drone flights to assess the performance of the tracking systems and the lidar data collected. The target orbit was taken to be located at the orbit of the International Space Station, both because it is a common target of RPOs and because a large proportion of RPO missions take place in a near-zero
eccentricity orbit. The semi-major axis was 6738 km, and the eccentricity was zero. The standard gravitational parameter of $3.986\times10^{14}$ m$^3$/s$^2$ was used for the Earth. This simulation allowed an input for the desired number of orbital periods to simulate. Since the target orbit’s eccentricity was zero and the relative distances of the two satellites were small given the nature of RPO, the HCW trajectory was chosen as the solution of choice for this thesis.

This simulation employed a bounded relative motion condition by enforcing that the chaser’s initial relative velocity in the $y$ direction obey the following: $\dot{y}_0 = -2nx$. This bounded condition was enforced since it was desired to have drone flights circumnavigate the central target, in order to collect all the lidar data desired. The simulation then solved and plotted the trajectory (Figure 10). As the figure illustrates, the three solution methods are nearly equivalent.
This thesis simulates the two-body problem since these applications are for low earth orbit missions, and therefore don’t involve a third body. The two bodies are the Earth and a chaser spacecraft. This assumption is satisfactory for the experiments at hand.

After simulating the relative orbital motion between a chaser and target spacecraft, the resulting chaser trajectory was normalized with min-max scaling to fit the available flight volume of the lab space. Trajectory data was scaled by the following, where $x'$ is the normalized value, and $C$ and $D$ are the limits of the new range of data [26]:

$$x' = C + \frac{(x - \min(x))}{\max(x) - \min(x)}(D - C)$$  \hspace{1cm} (26)

The distance from the tracking system origin on the ground of the lab to the center of the model target satellite was measured for each flight. This distance, the z offset, was added
to the z coordinate of each point in the trajectory. As this trajectory only contained position and velocity time history, a yaw parameter was appended in order to have the drone circumnavigate the scale model satellite while pointing the lidar camera at the satellite continually. The yaw value was found by (27). In the lab, the yaw angle was defined as the angle between the -x axis direction and the camera’s pointing direction. These negative terms ensured that the camera pointed towards the origin of the tracking system.

\[ \text{yaw} = \arctan 2(-y, -x) \]  

(27)

The final parameter appended to the trajectory was a wait time, \( t_{\text{wait}} \), which was the amount of time instructed for the drone to wait between arriving at one waypoint and then moving on to the next waypoint. This \( t_{\text{wait}} \) was taken as 0.5 seconds, in order to fly a relatively smooth and continuous flight. This flight was simulated to confirm the trajectory was correct and ready to perform the desired flight.

![Figure 11: The waypoints of a drone flight, shown as sample points of the simulated orbital motion trajectory](image)

28
The duration of the flight could also be checked by multiplying the wait time $t_{\text{wait}}$ by the number of waypoints (the number of rows of the trajectory matrix). This check was performed to verify the drone would have ample battery life for flight. Finally, the complete trajectory matrix with $t_{\text{wait}}$ and yaw values was constructed and uploaded to the drone.
4. Flights and Data Collection

4.1. Experimental Setup

Several experiments were performed on the SpaceDrones platform for this thesis. The lab drone, outfitted with an Intel RealSense L515 Lidar Camera, was flown on the flight trajectories previously simulated. The drone circumnavigated a scale model of a radio satellite which was used as a model for a target satellite in low earth orbit. The model satellite was kept static, with no rotation. Lidar data was collected. The satellite was approximately 160 cm in length from one end of the solar panel to the end of the other solar panel, the hub and panels were 17 cm high, and the hub was approximately 48 cm from one side to the opposite side. The physical model satellite was affixed to the lab ceiling and floor with clear wire to allow only the satellite to be detected by the lidar camera. A CAD model of this satellite was created for the flights where virtual lidar data is collected, as well as to have a baseline geometry throughout the process.

![Figure 12: CAD model of the satellite used for lidar data collection](image)

This satellite features representative instruments and other components that allow it to be a good surrogate for the experiments. It features two solar panels, a thruster, and
communications. The features vary from large and relatively detectable to small, thin, and less detectable.

The L515 camera that was used in the flights is designed for indoor applications and is a good choice for applications that require depth data at high accuracy and high resolution. The camera is composed of an IR laser, a MEMS, an IR photodiode, an RGB imager, a MEMS controller, and a vision ASIC [27]. The MEMS is used to scan the IR laser beam across the field-of-view. The vision ASIC then processes the data from the reflected beam and outputs a depth point that represents the distance of a specific point in the scene from the camera. A point cloud is generated from all the depth points collected to represent the full scene. The camera has a range of 9 meters, it has an output resolution of $1024 \times 768$ at a depth frame rate of $30$ fps, and it is only 100 grams for easy use on small autonomous systems. The camera has a depth accuracy of approximately 5 mm to 14 mm, and it has a depth field of view of $70^\circ \times 55^\circ$. Importantly, this camera can be used with Intel’s open-source SDK 2.0 which offers a variety of wrappers that support several programming languages for software development. This development kit allows this hardware to be integrated into the SpaceDrones platform.

![Figure 13: L515 Lidar Camera [28]](image)
The drone used in these flights is a 4-DOF hexcopter developed by a team within Virginia Tech’s Department of Aerospace and Ocean Engineering. Figure 14 shows the design of the body along with the lidar camera and Vive tracker affixed.

![Image](image.png)

Figure 14: Six-rotor drone chassis outfitted with lidar camera and Vive tracker on the tracking fixture on top [7]

OptiTrack and Vive tracking systems were used. OptiTrack is the higher-cost system while Vive is the lower-cost system. For the OptiTrack configurations, six specialized OptiTrack reflective markers were affixed to the top of the drone, where the central body attaches to the six arms. For the Vive configurations, the Vive tracker was affixed to the top fixture. Figure 14 shows the top fixture used for these attachments that allow for easy change from OptiTrack reflectors to Vive tracker. In the OptiTrack setup, 16 OptiTrack cameras were placed at elevated locations and arranged around the flight area, and in the Vive setup, four base stations were placed at elevated locations and arranged around the flight area. The lab space had an area of approximately 5.5 x 3.5 meters and a height of approximately 3 meters. The origins of the tracking systems were set approximately at the center of the room.
For the virtual lidar flights, the satellite CAD model was imported into a virtual environment in Unreal Engine. Unreal Engine is a 3D computer graphics video game engine that allows for the development of very high quality scenes for virtual reality applications [29]. The environment was a lidar scan previously performed of the full lab space itself. The virtual environment had a virtual sensor on a virtual drone that simulated the collection of lidar data. This virtual lidar is a rendered grayscale depth image of all the meshes within the virtual sensor’s field of view in the virtual environment. It is not true lidar, but it returns the ranges of points as seen by the virtual sensor. The drone was flown in the real lab without the physical satellite needed. During the flight, the position measured from the tracking system was sent to the virtual sensor in Unreal, and virtual lidar data was collected from this position. Figure 16 illustrates this virtual setup.
The experiments were broken into four configurations depending on the tracking system and type of lidar used. Table 1 shows what the four configurations entail, and Table 2 outlines the full parameter list for each of these flights.

Table 1: Quadrants of each tested configuration

<table>
<thead>
<tr>
<th></th>
<th>Virtual Lidar</th>
<th>Real Lidar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vive</strong></td>
<td>Quad 1</td>
<td>Quad 2</td>
</tr>
<tr>
<td><strong>OptiTrack</strong></td>
<td>Quad 3</td>
<td>Quad 4</td>
</tr>
</tbody>
</table>
These experiments and their results involve several sources of error which must be broken down and discussed. The tracking systems, the drone’s controller, and the lidar camera all introduce errors through their design and precision. This results in three possible distinct position values: true position, measured position, and desired position. The following outlines the introduction of errors into the system, and therefore the sources of noise in the data collection:

- Measured position is sent from the tracking system to the drone controller. The error between the true position and the measured position is a function of the tracking system’s precision.
- The drone controller attempts to converge the error to zero between measured position and desired position. The error between true position and desired position is therefore a function of the drone controller’s design and the tracking system’s precision.
- Lidar data is collected during a flight with the lidar sensor located at the measured position at a given timestep. The point cloud data error is therefore a function of the lidar camera’s precision and the tracking system’s precision.

### Table 2: All flight parameters used in experiment setup

<table>
<thead>
<tr>
<th>Quad</th>
<th>Tracking System</th>
<th>Lidar Type</th>
<th>Semi-major axis [m]</th>
<th># of Periods</th>
<th>Samples per Period</th>
<th>ecc</th>
<th>Initial State [m m/s]</th>
<th>Flight vol x [m]</th>
<th>Flight vol y [m]</th>
<th>Flight vol z [m]</th>
<th>twait [s]</th>
<th>Satellite Offset [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vive</td>
<td>Virtual</td>
<td>6738e3</td>
<td>1.2</td>
<td>75</td>
<td>0</td>
<td>[-10 0 0 0 2<em>n</em>10 0]</td>
<td>-3.1 to 1.6</td>
<td>-1.5 to 1.5</td>
<td>0 to 1.9</td>
<td>0.5</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>Vive</td>
<td>Real</td>
<td>6738e3</td>
<td>5.2</td>
<td>75</td>
<td>0</td>
<td>[-10 0 0 0 2<em>n</em>10 0]</td>
<td>-2.5 to 2.2</td>
<td>-1.5 to 1.5</td>
<td>0 to 1.9</td>
<td>0.5</td>
<td>0.88</td>
</tr>
<tr>
<td>3</td>
<td>Optitrack</td>
<td>Virtual</td>
<td>6738e3</td>
<td>1.2</td>
<td>75</td>
<td>0</td>
<td>[-10 0 0 0 2<em>n</em>10 0]</td>
<td>-3.1 to 1.6</td>
<td>-1.5 to 1.5</td>
<td>0 to 1.9</td>
<td>0.5</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>OptiTrack</td>
<td>Real</td>
<td>6738e3</td>
<td>1.2</td>
<td>75</td>
<td>0</td>
<td>[-10 0 0 0 2<em>n</em>10 0]</td>
<td>-3.1 to 1.6</td>
<td>-1.5 to 1.5</td>
<td>0 to 1.9</td>
<td>0.5</td>
<td>1.22</td>
</tr>
</tbody>
</table>
The errors that are present change depending on which of the four configurations from Table 1 the experiment is in. Quadrants 2 and 4, which use real lidar, have error from the lidar camera. In these quadrants, the drone has the three distinct position values discussed above associated with it at any time. However, with virtual lidar in quadrants 1 and 3, the true position and the measured positions are equivalent. This is the case because the tracking system sends the measured position to the virtual environment, and it is at this measured position (rather than the true position, which cannot be directly known) in which the virtual lidar data is collected.

The following sections outline the hardware and software configuration of each experiment performed. In addition to testing high and low-cost systems, these configurations represent an effort to minimize the reality gap that exists between a purely virtual simulation and a physical hardware-in-the-loop simulation. These configurations test either virtual or physical sensors, flight hardware, and environment.

4.2. Quadrant 1

The Quadrant 1 configuration uses virtual lidar and a virtual environment with the Vive tracker system. This means the target satellite is not physically used, but rather the CAD model is imported into the virtual environment. Here, the Vive collects the position time history of the drone, and the drone receives this data to adjust its position. This configuration is more trivial compared to the others. The virtual environment is an idealized model of the target satellite rather than the as-is physical satellite with imperfections and possible geometric changes. The virtual lidar sensor is not true lidar but
simply models the data collection within a field of view to mimic the physical sensor. This configuration does not explore the effect of true vs measured position error since they are equivalent in this case. This means no noise is expected in the resulting point cloud. Therefore, this is largely a software-in-the-loop problem as the physical drone does not necessarily even need to be flown.

However, while more trivial than the other configuration, this configuration offers the capability of full 6-DOF flight simulations since a physical drone is not necessarily needed. This offers an improvement to the 4-DOF limits of the drone used during these flights; however, a purely virtual demonstration falls short of demonstrating and testing physical systems and sensors which is the greater goal of a space simulator platform.

### 4.3. Quadrant 2

Quadrant 2 represents the most useful configuration of the four, given its low-cost tracking system and its minimization of the reality gap. This configuration offers real-world data from a physical lidar sensor as it scans a physical satellite model, and it tests the much lower cost Vive tracker. These factors make this configuration the hardest one in which to achieve high-quality data, and therefore the most valuable for assessing the SpaceDrones capabilities. Five flight trials were performed for this quadrant given its importance.

Quadrant 2, if shown to be comparable to the high-cost tracking solution, is the highest value configuration since it would mean that it is possible to achieve data collection and
tracking at a high enough quality for ground testing. This configuration could be used for more testing of spacecraft dynamics for further RPO studies.

4.4. Quadrant 3

Quadrant 3 returns to a more trivial, idealized configuration consisting of the OptiTrack system and virtual sensor and environment. It is largely similar to quadrant 1 except for the increase in cost. One benefit could be that, if flown with a physical drone, this configuration could provide virtual simulations where the measured position more closely resembles the true position of the drone. Therefore, this quadrant is a more niche configuration that would provide further benefit in applications where Vive tracking doesn’t provide as much value, such as for tracking flexible, nonrigid objects.

4.5. Quadrant 4

Quadrant 4 offers the highest fidelity data by combining OptiTrack with physical sensors in a physical environment. The OptiTrack system, with better tracking, would in theory return a less noisy point cloud than quadrant 2. Ultimately, it is quadrant 2 and quadrant 4 that are of greatest interest for direct performance comparison.
5. Results and Discussion

5.1. Tracking System

The drone flights resulted in measured position time history, the desired position time history, and inertial point cloud data. First, the measured position data was analyzed to measure the error in each configuration between measured and desired trajectory. These measured position paths were visualized (Figure 17).

Figure 17: Complete measured position time history
Each flight involved three flight phases: drone flight from the ground to the trajectory start point, the uploaded trajectory itself, and the drone’s return to home and shutdown command. For the purposes of this work, the data from the uploaded flight itself was extracted for analysis.

In order to directly compare the measured and desired position time histories, the geometry of the desired trajectory was constructed. This process was possible since the desired trajectory was simply an ellipse and could be described exactly by its known semi-major and semi-minor axes. These values were easily found by referring to the flight volume minimums and maximums from Table 2 as these were the coordinates to which the simulation trajectory was scaled for the drone flights. This geometry was then discretized into the same number of datapoints as the measured time history.
Figure 18: Measured position and desired position time histories during flight trajectory

The measured position and trajectory data were directly compared to find the absolute errors in the coordinates x, y, and z, as well as the norm of the position vector at each time step. Figure 19 shows how these values compare across each data sample.
Figure 19: x, y, z vs sample number for each quadrant

The average error across all data points was calculated for x, y, z, and the norm, and the standard deviation was found for each of these error distributions. Figure 20 displays these results for each flight.
Figure 20: Average absolute error and standard deviation in x, y, z, and the position vector norm for each quadrant

Once each flight’s errors were calculated, each flight’s average norm error and its standard deviation were plotted to directly compare the performance of each flight configuration. Figure 21 shows this final error comparison.
This final comparison of errors showed that the Vive & Real Lidar and the OptiTrack & Real Lidar configurations had comparable errors in their trajectories. The Vive & Real Lidar configuration had an average absolute error of 7.5 cm, and the OptiTrack & Real Lidar configuration had an average absolute error of 7.9 cm. In fact, each configuration regardless of tracking system returned an absolute error of at least 7.5 cm, indicating that this was the best achievable outcome. This demonstrated that the drone controller error dominated in these experiments rather than the tracking system error. This result is further

Figure 21: Average absolute error, and error standard deviation, of position vector norm for each quadrant
supported by the documented tracking system error from these systems’ vendors, which claim precision of millimeter magnitude.

In addition to this result, Figure 21 indicates that the OptiTrack & Virtual Lidar flight returned an average absolute error of 12.7 cm. This error is approximately 1.6x higher than the other flights’ average absolute errors. While this error appears to indicate a worse performance from the OptiTrack configuration, this result cannot be concluded since only one trial of this configuration could be performed, and since the drone controller error dominated in all flights. Additionally, the mean error of the other three configurations lies nearly within the error bar of this configuration, which further bars this conclusion.

These tracking system results showed that for this kind of drone-based spacecraft simulation, the Vive tracker provided a satisfactory tracking performance, with no decrease in performance when compared to the OptiTrack results. However, this result is due to the nature of drones and their controller error: OptiTrack precision is higher than Vive, but this advantage has no bearing in this context. Additionally, while a minimum average error of 7.5 cm may be higher than desired, it is important to consider that this is simply the absolute error – as the flight trajectory size is scaled up, the relative error decreases since the drone controller error remains constant. Therefore, if the trajectory ellipse were doubled in size, the relative error would halve. Significantly, given these results, only a Vive setup is needed to perform these experiments, which would save approximately $39,000 when compared to the OptiTrack setup option.
Overall, this result demonstrates that introducing physical hardware (i.e. a drone and tracking hardware) into the hardware-in-the-loop system is a viable option that continues to yield data of a satisfactory fidelity as compared to the idealized target trajectory that simulated, virtual hardware would fly. In other words, by using physical hardware, it was possible to minimize the reality gap and still gain satisfactory results for the case of an RPO demonstration.

5.2. Lidar Data

Each flight also captured lidar data which resulted in an inertial frame point cloud. Two frames of lidar data were captured each second. Given that the duration of each flight was between one and five minutes, this resulted in very large datasets. For each flight, a region of interest was defined in order to extract just the desired lidar data of the target satellite and remove the data that was captured of the surrounding lab space. This region was easily defined since the geometry of the satellite was known. In addition, a time region of interest was specified, which was simply the start and end times of the flight trajectory itself. These reductions of the dataset resulted in a much smaller workable dataset. Table 3 displays the elapsed time of each flight and the reduction in dataset size.

<table>
<thead>
<tr>
<th></th>
<th># Periods</th>
<th>Total Data Stream Elapsed Time [s]</th>
<th>Trajectory Elapsed Time [s]</th>
<th>Full Point Cloud # Data Points</th>
<th>Region of Interest # Data Points</th>
<th>Percent Reduction in # Data Points [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vive &amp; Virtual Lidar</td>
<td>1</td>
<td>93.89</td>
<td>36.84</td>
<td>11,285,051</td>
<td>20,875</td>
<td>99.82</td>
</tr>
<tr>
<td>Vive &amp; Real Lidar</td>
<td>5</td>
<td>279.81</td>
<td>186.81</td>
<td>10,591,764</td>
<td>174,568</td>
<td>98.35</td>
</tr>
<tr>
<td>OptiTrack &amp; Virtual Lidar</td>
<td>1</td>
<td>87.78</td>
<td>37.23</td>
<td>10,330,461</td>
<td>20,618</td>
<td>99.80</td>
</tr>
<tr>
<td>OptiTrack &amp; Real Lidar</td>
<td>1</td>
<td>113.80</td>
<td>36.38</td>
<td>3,106,128</td>
<td>30,383</td>
<td>99.02</td>
</tr>
</tbody>
</table>
Each resulting point cloud was translated to be centered at the origin, for direct comparison.

Figure 22, Figure 23, Figure 24, and Figure 25 show the final point cloud for each flight configuration.
Figure 22: Point cloud from Vive & Virtual Lidar configuration
Figure 23: Point cloud from one trial of Vive & Real Lidar configuration
Figure 24: Point cloud from OptiTrack & Virtual Lidar configuration
Figure 25: Point cloud from OptiTrack & Real Lidar configuration
This point cloud data illustrates the fidelity of each flight configuration and the resulting noise. Figure 22 and Figure 24 were captured with virtual lidar and a virtual satellite. Therefore, these point clouds represent the ideal case with the ideal geometry. These point clouds are nearly identical since the measured position at each time step is the point at which lidar data is collected in the virtual environment, acting functionally as the true position. This virtual lidar detects all the instruments of the target satellite, including those that are thin structures. Overall, these point clouds nearly reconstruct the original CAD model itself given the nature of the virtual sensor.

On the other hand, Figure 23 and Figure 25 show the result of introducing a physical lidar sensor and physical satellite. The result captures the actual as-built geometry of the physical satellite, with defects such as a detaching and bent left solar panel (Figure 26). Also, the physical lidar sensor could not capture the small instruments of the satellite as the resolution does not allow for such small-scale data collection. However, the large structures that are the solar panels and hub are clearly resolved. Given the error of the tracking systems and the lidar camera, noise exists in the point clouds from these configurations. This is a result of the lidar camera’s precision, and the error between the measured position and the actual position of the drone at a given time. In other words, at a given timestep, a lidar depth image is collected with the depth data based on the true position of the lidar camera. However, since the measured position of the camera is not equal to its true position, that frame of depth data is measured as having originated from a different, incorrect point. The accumulation of this measured position versus true position error across each time step results in this slight drift across lidar frames. Figure 26
highlights how this frame drift appears in the point cloud, making the satellite features appear thicker.

Overall, the point cloud results reveal that minimizing the reality gap by using a physical lidar sensor and physical satellite returns high quality lidar data of value for this kind of spacecraft simulation testing. These experiments provided real-world data with real-world error, which in turn can be used for more robust simulation development that is not too reliant on the ideal case. These point cloud characteristics are modellable – CAD models can be changed to reflect the as-built condition of the satellite, and noise can be added to simulate the erroneous points that result from real lidar scans. Additionally, while this real lidar data provides valuable real-world data, virtual data is still of great value. Virtual data collection requires the simplest setup of any configuration; there is no need for any physical
sensors, model, or even physical hardware like a drone. Therefore, virtual data collection represents the simplest, cheapest starting point for this kind of spacecraft simulation.

5.3. **Future Work**

With point clouds captured, there are several computer vision techniques that could be applied in future work. Point cloud matching, such as with iterative closest point algorithms, could be performed in order to directly compare virtual sensor point clouds with physical sensor point clouds. This could be especially useful for change detection, such as for satellite inspection applications.

In addition, a 6-DOF drone could be used for further exploration of simulated trajectories and tracking system capabilities since this would better match actual spacecraft motion. Virginia Tech has developed an unmanned vehicle, an omnicopter, that can undergo decoupled translational and rotational motion. This vehicle could explore more complex trajectories than the 4-DOF hexcopter allowed in this work.
6. Conclusions

This thesis incorporated lidar sensors into the SpaceDrones platform in order to explore the performance of different configurations of physical and virtual hardware, sensors, and environment. These configurations were tested by creating a relative orbital simulation tool and selecting a trajectory to simulate an RPO. Two position tracking systems were tested to find how the low-cost option compared to the high-cost option.

Ultimately, the average absolute error for each flight configuration revealed that the low-cost Vive tracking system returned approximately the same results as the high-cost OptiTrack system. While OptiTrack specifications indicate that a higher precision is expected from it, the data suggested that the drone controller error dominated the error in this research. This was determined by the disparity between the tracking system vendor precision specifications and the average error attained in each flight, and well as by the fact that each configuration had very similar average errors, despite the tracking system used. This showed that the tracking system had very little bearing on the absolute error. The nature of drones eliminated any advantage that the OptiTrack system had. Therefore, low-cost tracking systems like Vive are well-suited for this kind of simulator platform that uses drones. Since this kind of testing platform only needs a Vive setup, approximately $39,000 can be saved as compared to the OptiTrack setup, making this a great setup for academics.

As for the lidar data generation, it was found that minimizing the reality gap with the low-cost systems in the SpaceDrones lab yielded high quality and valuable lidar data that highlighted the characteristics of real-world data and real-world errors. The point clouds
demonstrated the variations between the as-built satellite and the CAD model, the frame drift error that results from measured position error, the loss of small feature details, and erroneous points. This data can then inform more robust simulation development by modelling these types of errors. While the real-world data was of great value, the idealized virtual lidar data generation was shown to be of great value as well. Virtual lidar generation does not require any physical model, hardware, or sensor which makes it an easy first step in the loop to minimize the reality gap, at very little cost. This demonstrates that virtual reality is a significant option that is well-suited for these kinds of spacecraft simulations.

Overall, a simulation platform such as SpaceDrones offers a high quality testbed for academics to perform ground-based simulation and testing without the prohibitive costs that are normally associated with such testbeds. Low-cost tracking systems have been shown to perform as well as high-cost options because of the use of drones; low-cost lidar sensors, physical models, and drone hardware have been demonstrated to return useful real-world data; and virtual reality has been leveraged to start the simulation process in a very low-cost and simple way.
Bibliography


Appendix

Figure 27: Vive setup with base stations elevated around flight area, physical satellite model located at tracking system origin, and physical drone and lidar sensor circumnavigating the satellite
Figure 28: Physical satellite model used for data collection

Figure 29: OptiTrack cameras set up around the flight area, with virtual (CAD) satellite being scanned by virtual lidar sensor in Unreal Engine, and drone circumnavigating the origin where the virtual satellite is placed