University of Colorado Boulder

Department of Aerospace Engineering Sciences

Senior Pojects – ASEN 4018

Coneceptual Design Document (CDD)

VANTAGE

Visual **A**pproximation of **N**anosat **T**rajectories to **A**ugment **G**round-based **E**stimation

---

Monday 1st October, 2018

Project Customers

Name: Penina Axelrad Email: penina.axelrad@colorado.edu Phone: (303) 492-6872	Name: John Gaebler Email: John.A.Gaebler@colorado.edu Phone: (407) 760-4295

Team Members

Name: Aaron Aboaf Email: aaron.aboaf@colorado.edu Phone: (720) 212-1974	Name: Dylan Bossie Email: dybo6303@colorado.edu Phone: (719) 502-9826
Name: Sean Downs Email: sean.p.downs@colorado.edu Phone: (303) 502-0393	Name: Justin Fay Email: jufa5768@colorado.edu Phone: (970) 403-7165
Name: Marshall Herr Email: mahe1052@colorado.edu Phone: (817) 475-8298	Name: Joshua Kirby Email: joki2454@colorado.edu Phone: (303) 916-3127
Name: Lara Lufkin Email: lalu4246@colorado.edu Phone: (719) 486-9408	Name: Richard Moon Email: rimo1046@colorado.edu Phone: (720) 934-5508
Name: Nicholas Renninger Email: nire1188@colorado.edu Phone: (303) 999-5725	Name: Zach Talpas Email: zata1281@colorado.edu Phone: (303) 520-4939
Name: Jiarui Wang Email: jiwa7945@colorado.edu Phone: (253) 205-1115	

Contents

List of Figures

List of Tables

1. Project Description

1.1. Purpose

The primary motivator for this project is to augment existing, ground-based CubeSat Space Situational Awareness (SSA) by observing CubeSat deployments from the perspective of the space-based deployer. The current method of tracking low-earth-orbit (LEO) CubeSats is to obtain orbit parameters via ground-based surveillance sites, typically using surveillance radar. These sensors will take multiple measurements of the CubeSats during their pass through the site's field-of-view (FOV), collecting them into a series of "tracks" which represent a small portion of that satellite's orbit. By assimilating these orbital tracks from many sites located across the globe, the Joint Space Operations Center (JSpOC) is capable of acquiring and maintaining the orbital parameters of every LEO CubeSat in orbit around the Earth[7]. Due to the limitations inherent to ground-based tracking systems, tracking data for CubeSats is often not available until several minutes to several hours after CubeSat deployment. The existing system therefore leaves room to be improved upon.

The delay inherent to ground-based tracking depends on several conditions. Although the telescopes at collection sites can typically aim anywhere on the site's horizon, the telescopes themselves will have relatively small fields-of-view. As a result, the viewing window for a particular CubeSat will often be small. During this time period, it is essential that the weather be sufficiently clear in that section of the sky and that the satellite maintains a sufficient visual magnitude to be identified at orbital range. If some of the first collections fail, data for that satellite will be unavailable until a successful collection can be made in the future. Additionally, CubeSats are often deployed clustered together. It can therefore require a great deal of time and tracking data points to distinguish the individual CubeSats in the cluster, which is not ideal for the CubeSat operators.

The VANTAGE project's use-case will significantly reduce delays in obtaining orbital tracks, as well as identifying CubeSats, by associating relative trajectory measurements with specific CubeSats at close-range immediately after deployment. Currently, the project is being designed towards a use-case on the NanoRacks CubeSat Deployer System (NRCDS) on the International Space Station (ISS). From the NRCDS, VANTAGE will have a unique vantage point providing different data collection possibilities in comparison to ground-based tracking. VANTAGE will provide a guaranteed collection window and will assist in verifying the CubeSat deployment manifest. The early data this provides will greatly reduce the initial uncertainty in the orbit of each deployed CubeSat. This decrease in initial uncertainty provides the project's marketability; VANTAGE will make it easier for orbital-debris mitigating organizations (NASA, NOAA, FCC) to authorize deployments from a VANTAGE-enabled deployer, due to the diminished uncertainty in CubeSat locations following deployment.

The purpose this year is to produce a proof-of-concept system which will be verified in a laboratory setting to measure the trajectory of identified mock CubeSats while integrated with a team-built test system. This test system will include a data interface, a power supply, and a structural interface which will all simulate the NanoRacks use-case system. The final test will involve using the VANTAGE proof-of-concept system to image up to 6 mock CubeSats simultaneously which are moving under the influence of a team-built test-rig linearly up to 100 m away from the system at a constant velocity of up to 2.0 m/s. The system will observe each mock CubeSat's trajectory (relative position and velocity) and track each mock CubeSat image-to-image, identifying them via a TBD combination of object recognition and dynamical predictions. Raw images, trajectory estimates, and identification data will all be stored on-board and relayed to the test system within 90 minutes of the first CubeSat's deployment, which is the amount of time between NanoRacks CubeSat Deployer Tube deployments of up to six CubeSats each. As an industrial benchmark, the above project goals for this year correspond to an overall NASA Technology Readiness Level[25] of 3-4 for the project.

1.2. Project Objectives and Levels of Success

Table 1 sets forth the levels of success for the VANTAGE project, which in turn have been developed into a set of requirements in Section 1.5. Higher levels of success imply fulfillment of the lower tier objectives in addition to the objectives stated at the higher level. Level one success would be considered baseline success, defined by VANTAGE interfacing with a test-bench computer and power supply to take images and position/velocity measurements (and their uncertainties) of Mock CubeSats during testing and then send this data back to the test-bench computer. Level two and level three success are built on level one success, but would simply be adding more functionality desired but not required by the customer; completion of all level three success criteria would mean VANTAGE has most of the functionality needed for its eventual on-orbit application.

The VANTAGE system will be tested near the end of the Spring semester. This project can be considered the first phase of VANTAGE which will be proof of concept testing. A test rig will be built to simulate the on-orbit application of VANTAGE on the ground. The test rig will interface with VANTAGE to mimic a possible on-orbit interface and

the test rig will deploy Cubesat shaped objects at realistic velocities for VANTAGE to track. Procedures will be implemented such that the results that VANTAGE outputs can be cross referenced and confirmed to critically evaluate the performance of the system.

Table 1: VANTAGE Levels of Success

	Structures
Level 1	A VANTAGE payload structure exists with models of all potential system components represented by physical, non-operating models. The structure is compliant with the relevant NanoRacks ICD for VANTAGE's **TBD** use-case structural interface. **TRL 3**[*]
Level 2	All system components (e.g. avionics, sensors, etc.), which operate in a "flat sat"[†] configuration, can be packed into VANTAGE payload's structure. However, these same system components do not necessarily operate nominally while inside VANTAGE's payload structure. **TRL 4**
Level 3	The VANTAGE system operates while all engineering-grade system components are integrated mechanically into the ICD-satisfying VANTAGE payload structure. **TRL 4**
	Tracking (In ideal lighting conditions)
Level 1	The VANTAGE tracking software takes 2 still images of mock CubeSats within 10 seconds of each mock CubeSat's mock-deployment during laboratory testing. After capturing this image data, and potentially other active sensing data, VANTAGE's software processes the data into position/velocity measurements and uncertainty bounds of the mock CubeSats during their mock-deployment period. **TRL 3**
Level 2	VANTAGE software will track between 2 and 6 mock CubeSats out to a 100 m range. It will also be able to compare the deployment velocity of the mock CubeSats with predicted deployment velocities sent to VANTAGE by the test PC over a communications interface similar to that of the NanoRacks Deployer. **TRL 4**
Level 3	VANTAGE's software shall process image and sensor data in real time to support advanced tracking algorithms. **TRL 4**
	Command & Data Handling
Level 1	VANTAGE's C&DH system is able to receive commands from a user-interfacing PC, transmit output data to the PC, and manage internal processes. It will have enough physical memory to store enough grayscale images to accomplish desired position and velocity estimate calculations as outlined in the Tracking Levels of Success. All data is offloaded and processed by a PC. **TRL 3**
Level 2	VANTAGE accepts the NanoRacks formatted deployment manifest file and uses these to inform the system when to begin data collection. VANTAGE stores images/video of mock CubeSat deployment operations on-board and transfers these back to the PC which simulates the NanoRacks data interface within 90 minutes of the first (of up to 6) mock CubeSat's deployment. **TRL 4**
Level 3	VANTAGE's C&DH system is able to receive commands, transmit output data, and manage internal processes while taking sensor readings or processing data simultaneously (i.e. C&DH runs in parallel with science operations). Output format is compliant with TBD data format requirements given by the Customer and / or NanoRacks. **TRL 4**
	Power
Level 1	The VANTAGE electronics hardware will accept power which corresponds to the available power from the NanoRacks system in the use-case. **TRL 3**
	Test System
Level 1	The VANTAGE system is tested by imaging mock CubeSats. These mock CubeSats are affixed to a handheld structure, and they are moved manually in VANTAGE's FOV while VANTAGE Tracking operations are performed.
Level 2	Mock CubeSats are moved by a test-rig which moves a single CubeSat at constant linear velocity (less than 2.0 m/s) from (0-5)m up to 100m downrange of the VANTAGE payload.
Level 3	Multiple mock CubeSats (up to 6) are moved by a test-rig which moves these CubeSats at constant linear velocities (less than 2.0 m/s) from (0-5)m up to 100m downrange of the VANTAGE payload.

[*]Each success level defines which NASA Technology Readiness Level (TRL[25]) is satisfied for this subsystem by meeting this level of success.

[†]A non-integrated state in which all payload components are laid out individually and connected such that they interact and operate as they would in the fully integrated system.

1.3. Concept of Operations (CONOPS)

Figure 1 presents the eventual use-case COncept Of OPerationS (CONOPS) for the multi-year vision of the project.

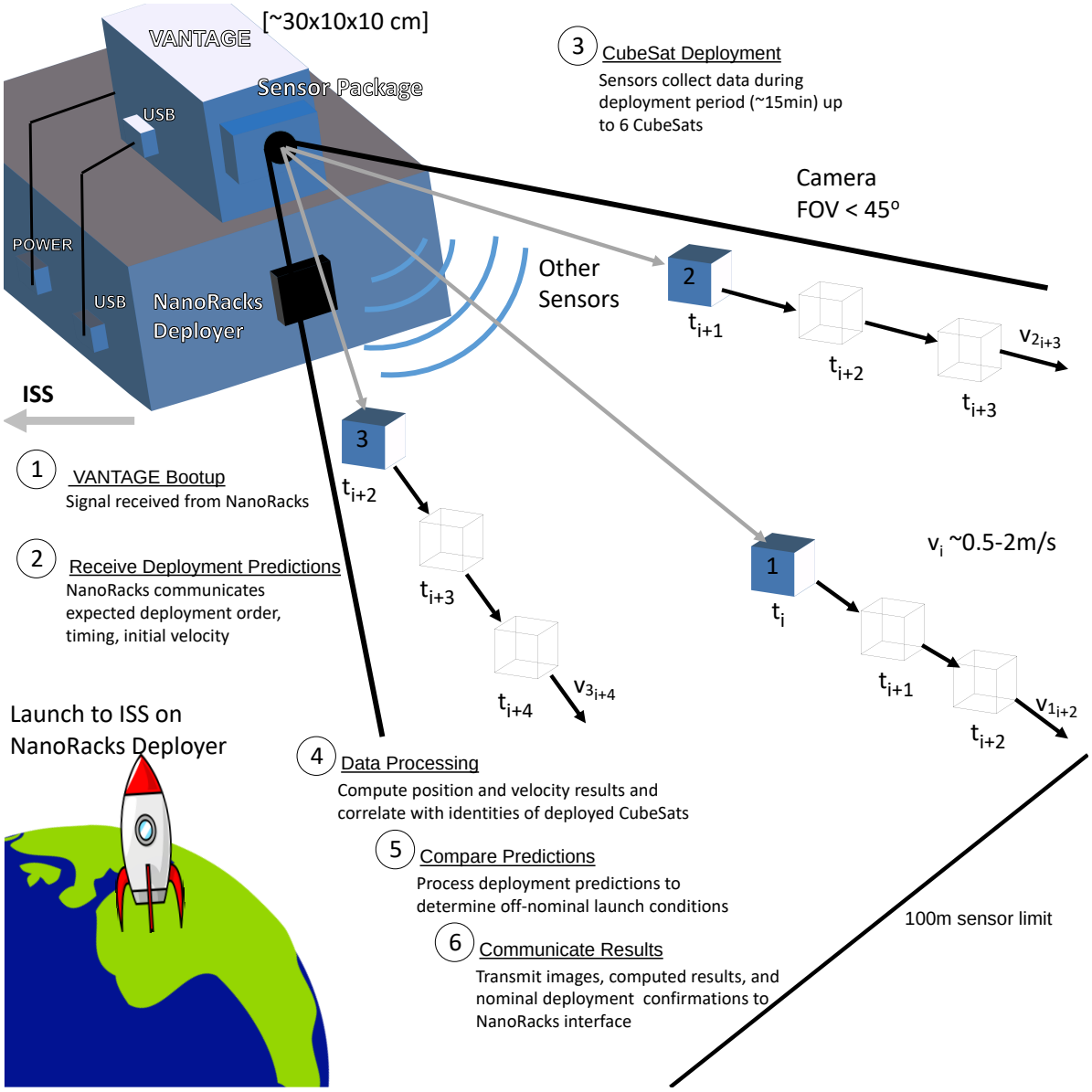

Figure 1. Eventual Use-Case CONOPS for the Multi-year Vision of the Project

Figure 2 presents the team CONOPS for our proof-of-concept project this year. It is worth noting that in #8 of this CONOPS diagram, 90 minutes has been chosen as the ultimate deadline to finish all processing and send data externally because NanoRacks has informed the team that the typical time between NanoRacks CubeSat Deployer (NRCSD) tube deployments (of up to 6 1U CubeSats each) is 90 minutes.

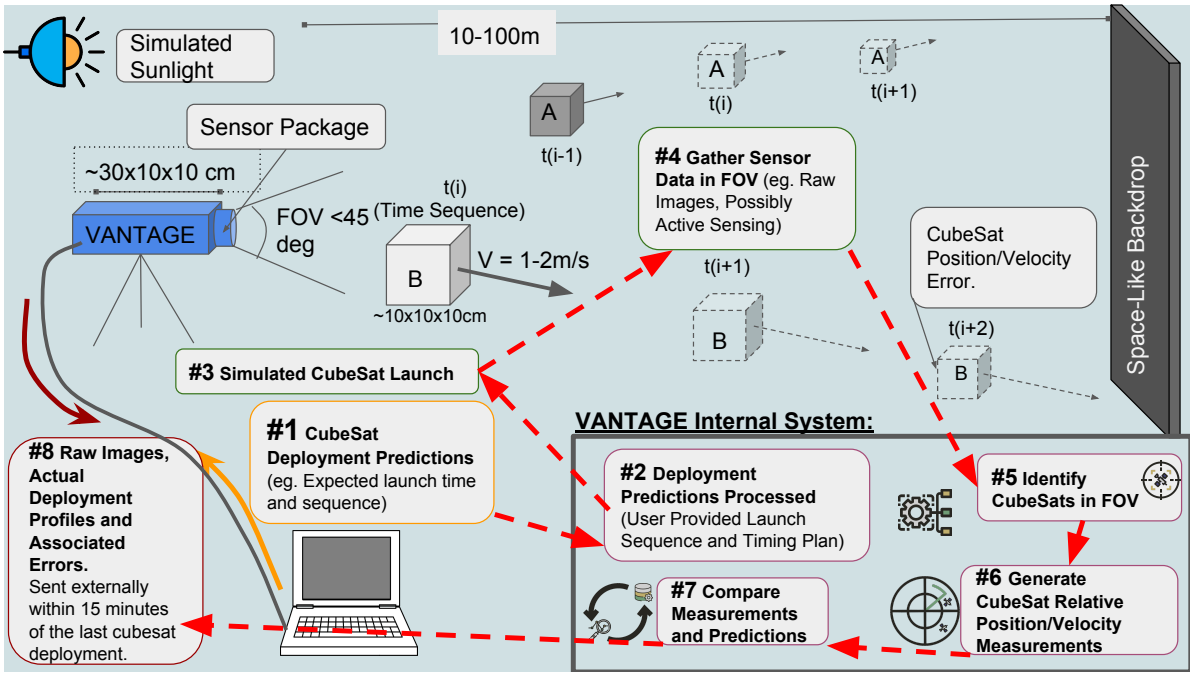

Figure 2. Team CONOPS for Our Proof-of-concept Project This Year

Figure 3 presents the functional block diagram for VANTAGE. Note that the maximum mass of the VANTAGE system is given for a sense of scale and not as an absolute maximum. The absolute maximum mass for a CubeSat in the NRCSD tube is 8.40 kg as listed in the NanoRacks NRCSD IDD. However, since it is possible for VANTAGE to also be designed to interface in the place of one NRCSD tubes, this is not a hard maximum.

1.4. Functional Block Diagram (FBD)

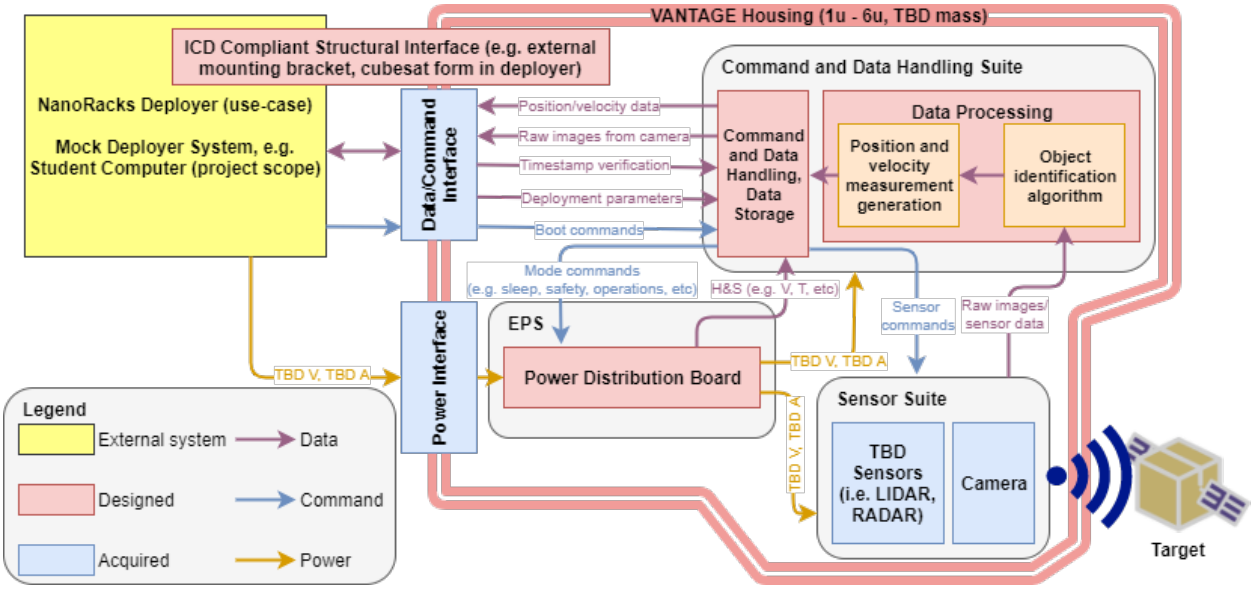

Figure 3. Functional Block Diagram (FBD) for VANTAGE

1.5. Functional Requirements

This section presents the Functional Requirements for the VANTAGE project. These requirements describe high-level functions which must be provided by the VANTAGE system in order to satisfy the need of the customer, and they have been presented to and confirmed by the customer. From each Functional Requirement (**FR**) will be flowed down several top-level Derived Requirements (**DR**), presented in Section 2.

It is worth pointing out the difference between **FR.1** and **FR.5**. **FR.1** requires that images be taken of deployed mock CubeSats but not that these images be used to obtain position and velocity vector measurements. **FR.5** requires that the system shall identify and track the mock CubeSats but does not specify that this must be done with a camera. The sensor suite to be used on VANTAGE is the object of a trade study presented in this paper.

FR.1: The system shall support in focus imaging of at most 6 mock 1U CubeSats while they remain between 3 and 100 meters from the VANTAGE payload.
FR.2: The system shall receive and interpret commands and the deployment manifest from a PC which simulates the NanoRacks use-case system.
FR.3: The system shall accept power analogous to that which is available from the NanoRacks use-case system.
FR.4: The system shall integrate mechanically with a structural interface which simulates the NanoRacks use-case system.
FR.5: The system shall uniquely detect and track up to 6 mock 1U-3U CubeSats while they remain between 3 and 100 m of the VANTAGE payload.
FR.6: The system shall estimate the position and velocity vectors of CubeSats between a distance of 3 and 100 m.
FR.7: The system shall recognize off-nominal deployment cases, which shall include impacts between mock CubeSats, off-nominal relative initial velocities, and off-nominal deployment times from the test system.
FR.8: The system shall report position/velocity vector measurements, off-nominal deployment cases, and raw images from the current mock deployment to the PC which simulates the NanoRacks use-case system before the next NanoRacks CubeSat Deployer (NRCSD) tube deployment would normally occur in the use-case.

2. Design Requirements

The Functional Requirements, labeled **FR**'s below, were each flowed down into several Derived Requirements (**DR**'s). Each derived requirement has a subsystem associated with it, denoted by a unique subsystem tag. These tags are as follows: **SN** for sensor subsystem, **EL** for electrical subsystem, **SW** for software subsystem, and **STR** for structural subsystem. Derived requirements were created by determining all the pieces necessary to verify each functional requirement. Some derived requirements fell out of meetings with the customer ,while others were created using calculations or experience in the area represented by the overarching functional requirement.

FR.1: The system shall support in-focus imaging of at most 6 mock 1U CubeSats while they remain between 3 and 100 meters from the VANTAGE payload.
Motivation: Customer Requirement. During a 90 minute ejection cycle, the NanoRacks Deployer deploys up to six 1U CubeSats. In order for the system to be useful to the customer, it must produce images of each mock 1U CubeSat while they remain between 3 and 100 meters from the VANTAGE payload. The customer has requested that structural details such as solar panels be clearly visible within these images, which will eventually drive derived-requirements having to do with resolution. These images may be used for mock CubeSat identification by the software subsystem and, in the use-case, would be used by the NanoRacks PR department.

DR.1.1-SN The system shall use a camera to capture images of mock CubeSats.
Motivation: The customer expects images of mock CubeSats to be a part of the deliverables. These images would be taken in the visible light wavelength.
Verification: Inspection. The system will be visually inspected to determine if a camera is a part of it. Images taken during other ground tests, such as the one done for **DR.6.1-SW**, will also be inspected to determine if mock CubeSats are within the FOV.

DR.1.2-SN Imaging subsystem shall have a FOV greater than 20°x20°.
Motivation: Mock CubeSats must be within the FOV of the camera in order for it to capture images of it, and this FOV is necessary based on calculations done using the Clohessy-Wiltshire equations applied to the use-case. To capture images of the the mock CubeSats deploying from the furthest tube possible at a distance of 3 meters,

a FOV of 20° is necessary. At a range of 100 meters, a CubeSat would drift less than 5° relative to the camera's boresight axis. These calculations can be seen in Appendix B.
Verification: Inspection. Technical specifications from the selected camera will provide the respective FOV.

DR.1.3-SN: Imaging subsystem shall produce at least 2 images of each mock CubeSat deployed by the test system.
Motivation: The customer expects at least 2 images of each mock CubeSat.
Verification: Ground testing. The sensor subsystem will be hooked up to a PC simulating the NanoRacks use-case system, and a command will be sent stating that a deployment will occur. A mock CubeSat will then be placed in front of the sensor subsystem and begin to move away. The images taken by the sensor subsystem with then be inspected to ensure that 2 images were taken of the mock CubeSat.

DR.1.4-SN: Imaging subsystem shall produce in-focus images of mock CubeSats.
Motivation: As the customer requires that detailed images be produced of each mock CubeSat, these images must be in focus in order for the structural details of each mock CubeSats to be visible within them.
Verification: Ground testing. The sensor subsystem will be hooked up to a PC simulating the NanoRacks use-case system, and a command will be sent stating that a deployment will occur. A mock CubeSat will then be placed in front of the sensor subsystem and begin to move away. The images taken by the sensor subsystem with then be inspected to ensure that the mock CubeSat is in focus.

FR.2: The system shall receive and interpret commands and the deployment manifest from a PC which simulates the NanoRacks use-case system.
Motivation: Customer Requirement. In order to receive commands and the deployment manifest, which consists of expected mock CubeSat deployment times and velocity, from the test system there must be an interface with the PC which simulates the NanoRacks use-case system.

DR.2.1-EL: The electronics subsystem shall interface with the PC which simulates the NanoRacks use-case system via a USB2.0 Port for all data communication needs.
Motivation: This is the communication port which NanoRacks has defined as the most feasible interface between the system and NanoRacks use-case system. It will therefore be a part of the proof-of-concept design.
Verification: Inspection. The system will be inspected to ensure that a USB2.0 port is present.

DR.2.2-SW: Software subsystem shall be capable of interpreting a deployment manifest file sent from the PC which simulates the NanoRacks use-case system.
Motivation: Reading in the mock deployment manifest is necessary to be able to determine the nominal ejection case for each mock CubeSat, as it contains the ejection order of mock CubeSats, timing of ejections, and predicted ejection velocity. Currently, no example of such a file has been provided by NanoRacks, which reduces the detail which can be included in this requirement.
Verification: Ground testing. A mock deployment manifest will be inputted to the software subsystem on a PC, and the interpreted data from the manifest will be displayed on screen for confirmation.

FR.3: The system shall accept power analogous to that which is available from the NanoRacks use-case system.
Motivation: Customer Requirement. In order for the system to one day be integrated with the NanoRacks Deployer, the system must be designed to accept the power that can be provided to it.

DR.3.1-EL: The system shall operate with up to 120 VDC with a ripple voltage of 3Vpp and less than 5 A, which simulates the power available from the NanoRacks use-case system.
Motivation: In order for VANTAGE to receive power, it must be able to accept the power supplied by the NanoRacks use-case system.
Verification: Ground testing. The electronics subsystem will be hooked up to a variable power source set to 120 VDC with 3 Vpp and 5 A. The requirement will be verified if VANTAGE successfully operates through a full mock deployment cycle and reports data while receiving power within this range.

DR.3.2-EL: The system shall draw less than 520 Watts.

Motivation: In order for VANTAGE to receive power, it cannot exceed the power available to the NanoRacks use-case system.
Verification: Ground testing. The electronics subsystem will be hooked up to a variable power source set to 120 VDC with 3 Vpp and 5 A. The wattage that the subsystem is using will then be measured.

DR.3.3-EL: The electronics subsystem shall enter a low power mode when not performing any operations (i.e. before a final test has been started, after a final test has been completed and all post-processing and communications have completed).
Motivation: The system cannot always be operating at full power consumption if it is ever to be useful to NanoRacks. It would exist on the ISS where power budgeting is important.
Verification: Ground testing. The electronics subsystem will be sent a command simulating that the deployment has finished. The wattage that the subsystem is using will be measured to both before and after the command is sent to ensure that power into the system has decreased.

FR.4: The system shall integrate mechanically with a structural interface which simulates the NanoRacks use-case system.
Motivation: Customer Requirement. For the system to be of use to the customer, it must be capable of attaching to the NanoRacks Deployer. If the system design does not fit the sizing, mass, and mounting requirements set forth by NanoRacks, there will be a more difficult path forward for design development.

DR.4.1-STR: The system shall meet the structural requirements listed in NanoRacks Interface Control Documents.
Motivation: The system must be designed such that it is capable of interfacing with the NanoRacks Deployer. The options exist to design for structurally interfacing externally (NanoRacks ICD for external mount not yet provided) or internally to a deployer tube (NanoRacks document number NR-SRD-029 Revision 0.36) on the NanoRacks system. These design options will be compared in the following section, but for either choice the system must comply with the requirements of the relevant Interface Definition Document.
Verification: Inspection. If the system is externally mounted, measurements will be taken to ensure that all requirements in a yet-to-be-provided NanoRacks external-mount ICD are met. If the system is internally mounted, measurements will be taken to ensure the all requirements in NR-SRD-029 Revision 0.36 section 4 are met, except for subsections 4.7 and 4.8.

DR.4.2-STR: The VANTAGE team shall build a structural interface that simulates the NanoRacks Deployer structural interface use-case system accurate up to 0.005 in.
Motivation: In order for testing to occur, an accurate structure must be built that simulates how the system would be mounted onto the NanoRacks Deployer.
Verification: Inspection. The structural interface's dimensions will be measured to ensure accuracy to 0.005 in.

FR.5: The system shall uniquely detect and track up to 6 mock 1U-3U CubeSats while they remain between 3 and 100 m of the VANTAGE payload.
Motivation: Customer Requirement. In order for position and velocity estimates to be made for each mock CubeSat, as well as off-nominal deployments, mock CubeSats must be identified using sensor data. This is distinct from FR.1 because the sensor used for identification does not necessarily have to be an image sensor.

DR.5.1-SN Sensor subsystem shall have a sensing FOV of at least 20°x20°.
Motivation: This FOV is necessary based on calculations done using the Clohessy-Wiltshire equations applied to the use-case. To capture sensor data of the mock CubeSats deploying from the furthest tube possible at a distance of 3 meters, a FOV of 20° is necessary. At a range of 100 meters, a CubeSat would drift 5° relative to the sensor's boresight axis. These calculations can be seen in Appendix B.
Verification: Inspection. Specifications from the selected sensor will provide the respective FOV.

DR.5.2-SW: The system shall detect whether a mock CubeSat is within its FOV 99% of the time while they remain between 3 and 100 m of the VANTAGE payload.
Motivation: The software subsystem must be able to find mock CubeSats within the sensor data that it processes.

Verification: Analysis. The simulated sensor data will be inputted to the software subsystem.

FR.6: The system shall estimate the position and velocity vectors of CubeSats between a distance of 3 and 100 m.
Motivation: Customer Requirement. The overall purpose of this system is to estimate the relative position and velocity vectors of mock CubeSats.

DR.6.1-SW: Software subsystem shall produce relative position vector estimates accurate up to 10 cm 1σ to a distance of 10 m, changing to an accuracy of at least a tenth of the range 1σ up to a distance of 100 m.
Motivation: The customer has explicitly asked for these error bounds on position estimates. In order for the produced estimates of position to one day be useful to the NanoRacks Deployer use-case system, they must be accurate. If the error bounds on these estimates are too large, the estimates themselves will not be useful in orbital determination.
Verification: Ground testing. The sensor subsystem will be connected to a PC simulating the NanoRacks use-case system, and a command will be sent stating that a deployment will occur. A mock CubeSat will then be placed in front of the sensor subsystem and begin to move away to a distance of 100 meters at a set velocity. The position estimates produced by the software subsystem will then be compared to the actual location of the mock CubeSat.

DR.6.2-SW: Software subsystem shall provide relative velocity vector estimates accurate up to 1 cm/s 1σ to a distance of 10 m, changing to an accuracy of 10 cm/s 1σ up to a distance of 100 m.
Motivation: The customer has explicitly asked for these error bounds on velocity estimates. Relative velocity estimates, just like the position estimates in **DR.6.1-SW**, must be accurate in order for them to one day be used for orbit determination.
Verification: Ground testing. This requirement will be verified by the same ground test described in the verification section of **DR.6.1-SW**. Velocity estimates will also be compared to the set velocity that the mock CubeSat is moved at.

FR.7 The system shall recognize off-nominal deployment cases, which shall include impacts between mock CubeSats, off-nominal relative initial velocities, and off-nominal deployment times from the test system.
Motivation: Customer Requirement. In the NanoRacks use-case system, it would be useful to know if off-nominal deployments occur. Knowing if impacts or off-nominal initial velocities occurred would help to explain possible loss of functionality of CubeSats. Off-nominal deployment times would assist in the early adjustment of expected CubeSat passover times of ground stations.

DR.7.1-SW: Software subsystem shall maintain current time, synchronized with global time from the PC which simulates the NanoRacks use-case system.
Motivation: In order for the time at which each mock CubeSat is deployed to be determined, the system must take in the global time.
Verification: Inspection. The software subsystem will be connected to a PC which simulates the NanoRacks use-case system, and the global time will be displayed on the screen.

DR.7.2-SW: Software subsystem shall recognize if mock CubeSats exit the test system greater than 3 seconds before/after predicted with a tolerance of 0.5 seconds 3σ.
Motivation: To be able to determine if mock CubeSats ejected at the correct time, an estimate must be made of when the mock CubeSat left the NanoRacks System Simulator. NanoRacks sometimes ejects CubeSats 2 seconds late/early, so ejections that are off by three seconds will be marked as off nominal.
Verification: Ground testing. The sensor subsystem will be hooked up to a PC simulating the NanoRacks use-case system, and a command will be sent stating that a deployment will occur, along with the time at which it will occur. A mock CubeSat will then be placed in front of the sensor subsystem and begin to move away at the inputted time plus 4 seconds. The requirement will be considered verified if the deployment is marked as off-nominal and the outputted time of deployment is accurate based on the defined error bound.

DR.7.3-SW: Software subsystem shall recognize impulsive changes in the relative velocity vector greater than 0.5 m/s 3σ between 3 and 100 meters from the VANTAGE payload.

Motivation: To be able to recognize a mock CubeSat impact, the rate at which the velocity of mock CubeSats changes must be tracked.
Verification: Ground testing. The sensor subsystem will be hooked up to a PC simulating the NanoRacks use-case system, and a command will be sent stating that a deployment will occur. A mock CubeSat will then enter the FOV of the sensor subsystem moving at a set velocity. At a later time, the velocity of the mock CubeSat will be doubled. The requirement will be considered verified if the software subsystem reports an impact.

DR.7.4-SW: Software subsystem shall recognize if initial relative velocities are less than 0.5m/s or greater than 2.0m/s with a tolerance of 0.1m/s 3σ.
Motivation: NanoRacks tells its customers that their CubeSats must withstand a launch velocity of 2.0m/s. If a CubeSat is launched a velocity greater than this, it may be damaged.
Verification: Ground testing. The sensor subsystem will be hooked up to a PC simulating the NanoRacks use-case system, and a command will be sent stating that a deployment will occur. A mock CubeSat will then enter the FOV of the sensor subsystem moving at a set velocity greater than 2.0 m/s. The requirement will be considered verified if the software subsystem reports a velocity greater than 2.0 m/s.

FR.8 The system shall report position/velocity vector measurements, off-nominal deployment cases, and raw images from the current mock deployment to the PC which simulates the NanoRacks use-case system before the next NanoRacks CubeSat Deployer (NRCSD) tube deployment would normally occur in the use-case.
Motivation: Customer Requirement. All the desired data that the system produces needs to be sent to the PC simulating the NanoRacks use-case system in order for it to be used. In the use-case, this would need to occur before the next deployment cycle begins. Deployments cycles occur every 90 minutes.

DR.8.1-EL: The electronics subsystem shall transmit all relative position and velocity vector estimates, as well as mock CubeSat deployment images back to the PC which simulates the NanoRacks use-case system within 15 minutes of final mock CubeSat deployment.
Motivation: In order for this data to be useful, it must be available within a short time period after the deployment of mock CubeSats. NanoRacks deploys a tube of up to 6 1U CubeSats in 90 minute intervals, so the electronics subsystem must be prepared to take in more data before the next deployment cycle begins. The customer has requested that the data be made available sooner than 90 minutes after the first CubeSat launch, hence the 15 minute requirement (which was confirmed with the customer).
Verification: Analysis. A data file simulating the desired data that would be collected during the deployment of 6 mock CubeSats will be loaded on the system and will be tasked to be transmitted via the USB2.0 port to a PC simulating the NanoRacks use-case system. The requirement will be considered verified if the data transfer occurs in under 15 minutes.

DR.8.2-EL The system shall store all images, sensor data, and estimates within an onboard data storage device.
Motivation: The sensor subsystem will be creating a large amount of data, and this data will not be processed and outputted to the PC simulating the NanoRacks use-case instantaneously. The data will therefore need to be stored onboard.
Verification: Analysis. The data storage device will be loaded with a file simulating the maximum amount of data that would ever be on it at one time. This maximum amount of data will depend on what type of sensor(s) are chosen.

---

3. Key Design Options Considered

There are a few key design challenges which present a broad solution space for the VANTAGE project. In order to refine these options down, it is important to explore this solution space and determine requirements-satisfying options upon which to trade. This section presents the various options considered over four trade spaces, and it describes whether they satisfy requirements and are feasible options in the scope of this project.

The first design trade is for programming languages. Software represents a significant portion of the project; consequently, the programming language chosen to implement VANTAGE's software will have major ramifications for development time, processing speed, etc.

The second design trade is for VANTAGE's sensor suite. While **FR.1** requires a camera for image capture, the type of sensor(s) whose measurements will be used to calculate relative position and velocity vectors is not constrained. This trade explores a design-space including passive sensors, active sensors, as well as various combinations of sensors.

The third design trade is on the system's avionics hardware. A software-heavy payload operating in space will certainly have need of computing hardware which satisfies certain performance, size, cost, power, etc. constraints. The electronics hardware selected for the project will have far-reaching implications for the scope and robustness of potential software algorithms. This trade explores the broad design space of possible Commercial-Off-The-Shelf (COTS) computing options, ranging from educational micro-processor boards to smartphones to personal computers.

The fourth design trade is for VANTAGE's structural interfacing scheme with the NanoRacks use-case system. One effect of selecting where the project is designed to eventually (not in this year's scope) integrate into the NanoRacks system on the ISS is to change VANTAGE's structural constraints, but the most significant effect of this trade is to change VANTAGE's perspective with respect to CubeSat deployment trajectories. Whether VANTAGE's line-of-sight is in-line with or off-set from CubeSat deployment trajectories vastly alters the software algorithm required to uniquely detect, track frame-to-frame, and measure position and velocity vectors for each CubeSat. This trade will explore the various options detailed by NanoRacks as possibilities for eventual integration with the NanoRacks use-case system on the ISS.

Choices made at this stage among these four design trades will have a lasting, defining impact on the entirety of the VANTAGE project. This section will detail the considerations for each key design option and will define whether the option will be considered in the trade studies in Section 4.

It is pertinent to note why independent algorithms and test-rig trades were not included in the CDD. The reason that an independent algorithms trade was not conducted in the CDD can be seen in Appendix C. The reason that an independent test-rig trade was not conducted in the CDD can be seen in Section 6. This note will be reiterated in Section 4.

3.1. Programming Language

The software developed for this project is the core of what will allow VANTAGE to succeed. VANTAGE has many complex software tasks to complete with limited resources (development time, constraints, etc.) in a limited amount of time (**DR.8.2**). Since our system has a discrete set of requirements and scope to fulfill, the software team is left with the option to trade on *how* these accomplishments will be done. For software development, this will translate into an informed selection of what programming language is used to fulfill the requirements. An intelligent decision of language is necessary for ensuring that the team maximizes efficient development, while still maintaining an environment capable of fulfilling requirements.

3.1.1. Python

Python is a popular choice for software development. Its simplicity and readability cuts down on hours invested in a project, by reducing the time spent by developers writing and debugging the code.

As a result of its popularity, Python is also one of the most crowd-supported programming languages. As seen in Fig 4, Python is by a wide margin the most discussed language on StackOverflow out of the languages traded in this study. However, a consequence of focusing on simplicity over efficiency comes at the cost of a significant drop in runtime, often to the point of being unusable for intensive applications. Since this application requires intensive computing and data processing, Python will experience a significant increase in runtime vs. other options. Additionally, it is a higher-level language, decreasing its feasibility with potential hardware choices made by the other subteams.

A key element in deciding on the language for this project is available packages; one of Python's strengths is its expanse of available open-source packages. The team's search for packages that might be applied to this project include: OpenCV, NumPy, and Pandas[35 34 36]. The requirements of applications for image processing, numerical analysis, matrix math, and computer vision are all satisfied with Python's range of available packages.

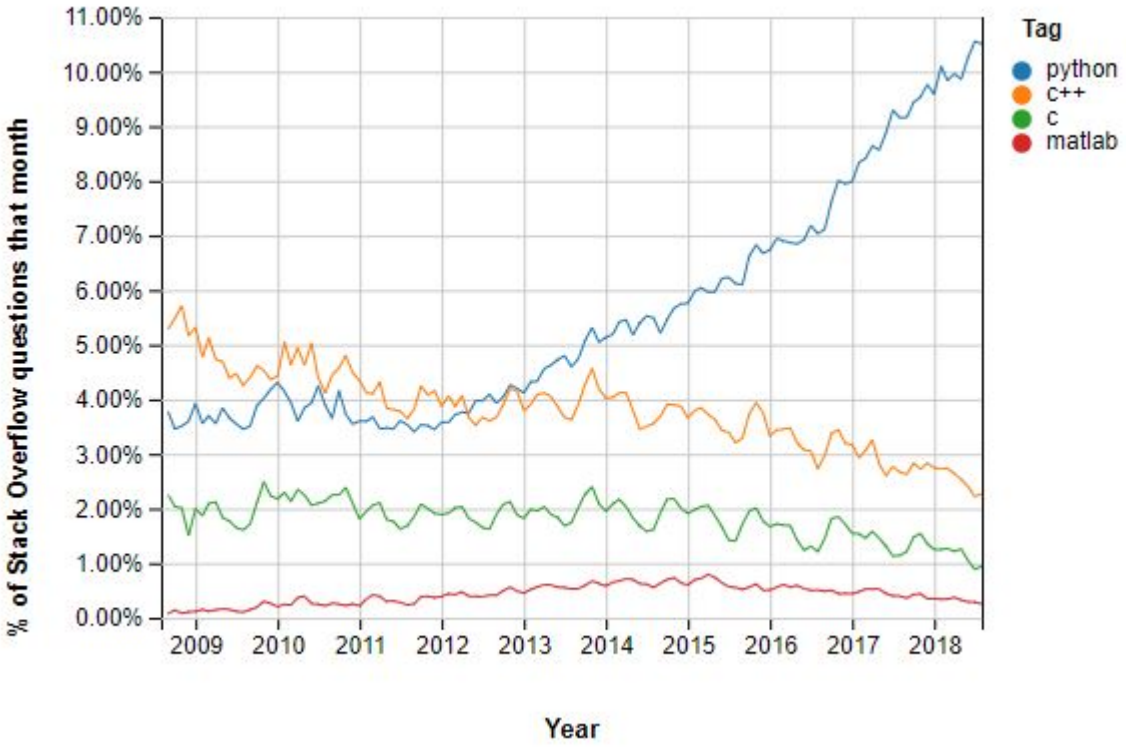

Figure 4. Percentage of total questions asked on StackOverflow over time, by language. [19]

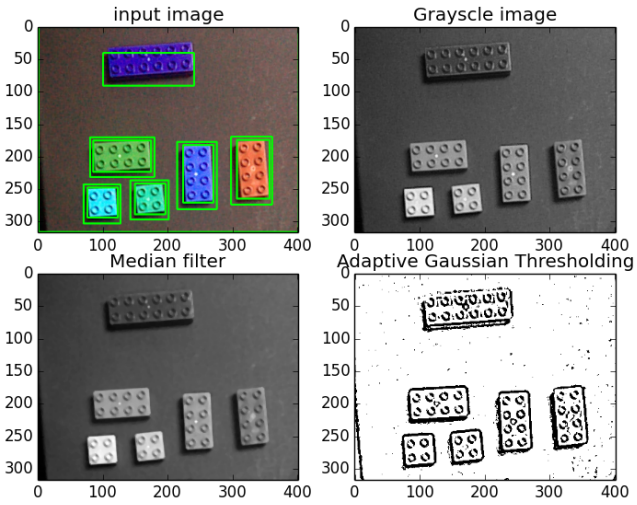

Figure 5. Example of image processing and detection algorithms in Python [22]**.**

Pros	Cons
Simple to handle packages and libraries	Typically worse runtimes than other options
Multiple image processing package options	Memory inefficient
Well-documented and supported language	Not compatible with all hardware options
	Higher processing demands

3.1.2. C++

C++ is one of the most traditional choices for software development. It is capable of operating at a lower level than languages like Python or MATLAB, enabling the application to work with information closer to machine code level if needed. Operating at a lower level implies that C++ also has potential to be more runtime and memory efficient than other languages in the same application. However, C++ is less documented, and more complicated than other high level languages, implying implementation difficulty when developing complex software systems such as VANTAGE. C++ has available packages such as armadillo, OpenCV, and ccv, fulfilling the requirements of image processing, scientific computing, and computer vision [35 29 30].

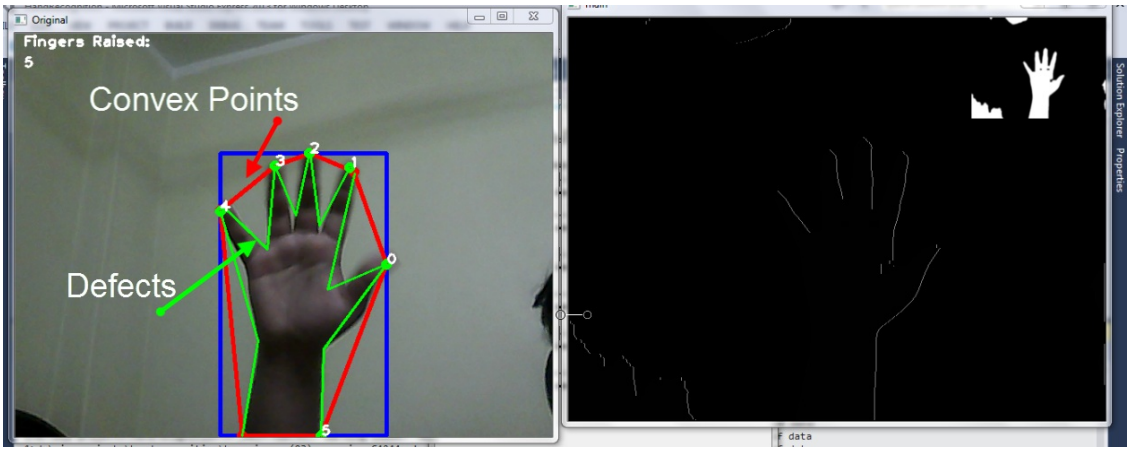

Figure 6. Example of image processing C++ [23]

Pros	Cons
Portable language	Fewer available packages
Exception handling (unlike C)	Package management less straightforward
Compatible with hardware options	
Memory efficient	

3.1.3. C

C is the 'lowest' of the high-level languages, and the 'lowest' of the languages considered in this study. This provides an potential further advantage from C++ in potential efficiency and memory-use improvements. However, it lacks traditional object orientation seen in other languages, increasing the difficulty of implementation for the developers. In addition, C is the language in which the team's developers have the least amount of experience, further increasing implementation time. However, C does have packages available such as ccv and librsb, and a different flavor of OpenCV, which will assist in fulfilling the mission requirements [35 30 33].

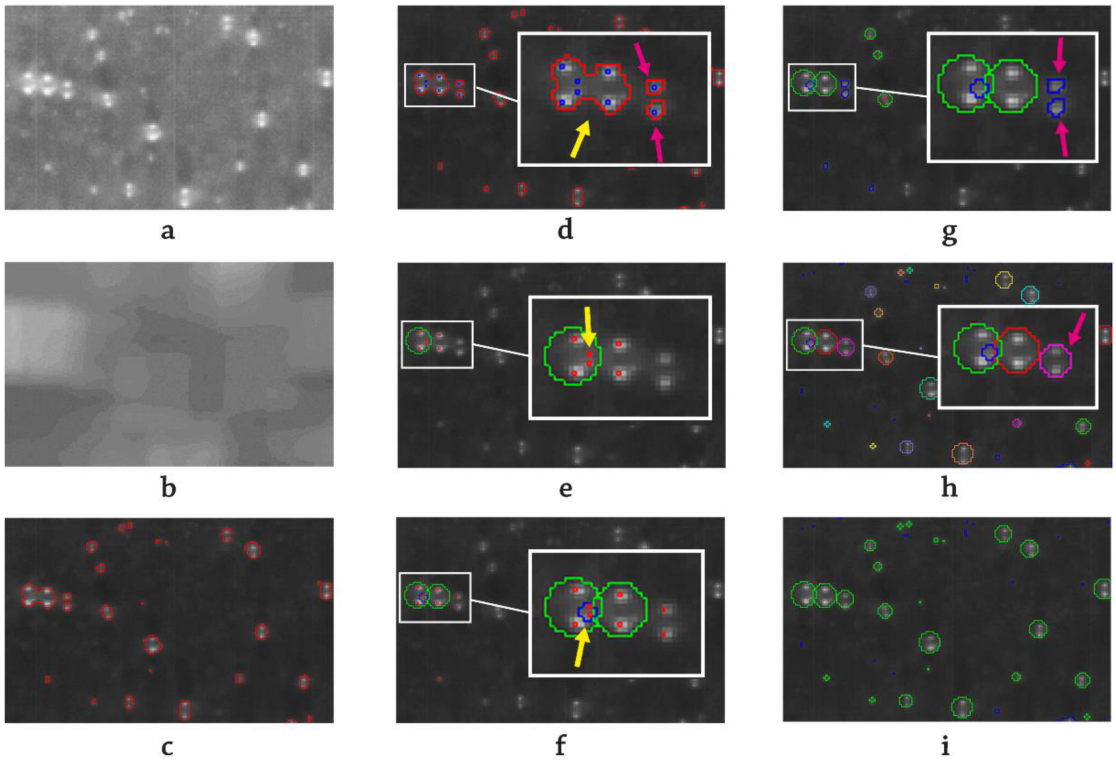

Figure 7. Example of image processing in C[24]

Pros	Cons
Very fast for many applications	Language team is least familiar with
'Bare-bones' language, portable	No exception handling
Memory efficient	Not implicitly optimized for scientific computing
	Fewest amount of available packages

3.1.4. MATLAB

MATLAB is a language renowned for its applications in matrix-related computations and scientific computing. Since much of image processing involves matrices and performing scientific analysis, it is a language that will likely shine when applied to VANTAGE. MATLAB is a thoroughly well-documented language for both its base code, and available packages and toolboxes, which will reduce implementation time for the developers. Beyond the scientific computing already available in base MATLAB code, included toolboxes which will augment the implementation of computer vision and image processing include the Image Processing and Computer Vision System Toolboxes[32 31].

When compared to the other high-level language in this trade study, Python, a useful comparison for image processing algorithms can be found below:

MATLAB Performance over Python	Average	Best
Engineering	3.2x	64x
Statistics	2.7x	52x
Graphics	31x	540x
Nested for loops	64x	64x

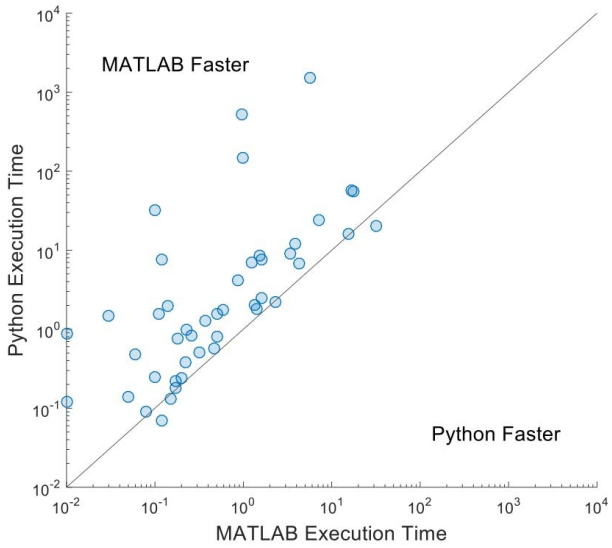

Figure 8. Comparison between MATLAB and Python in image processing applications[20].

While this doesn't comment on MATLAB's performance against the lower-level languages, it does make clear in terms of the higher-level languages, MATLAB is objectively more efficienct in image processing applications.

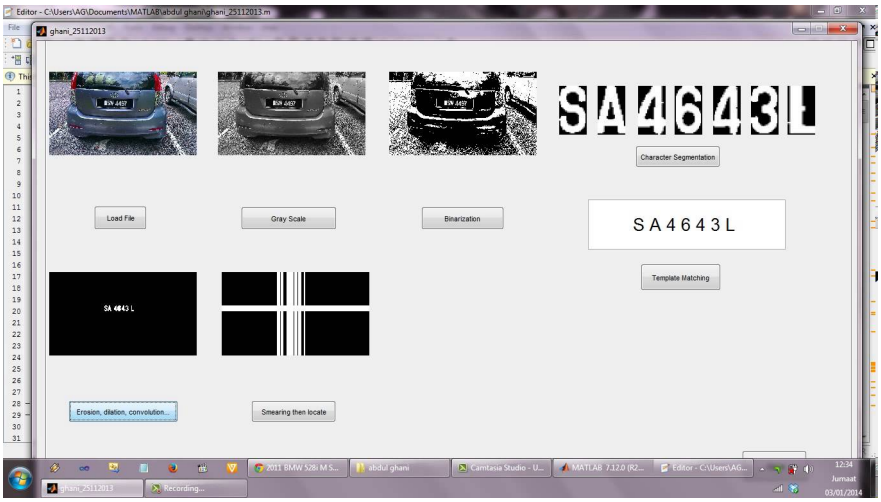

Figure 9. Example of MATLAB applied to image processing/detection algorithms[21].

Pros	Cons
Competent and efficient in matrix math	Higher processing demands
Vast array of available packages and toolboxes	Not compatible with all hardware options
High-quality documentation	Consumes more memory than more basic languages
Team is highly experienced with the language	

3.2. Sensor Suite

The sensor suite used in VANTAGE can significantly affect many other subsystems and is the major determinant of system accuracy. As such, it requires a wide range of design options to be considered. The customer has specified that the system should capture images of each deployment and output them via the data interface to the mock NanoRacks terminal. This specification requires a standard, visible light camera system to be included in the sensor suite. In accordance with this constraint the additional subsystem architectures considered will all include a camera.

3.2.1. Single Camera

A simple optical camera operating in the visible spectrum can be used for range measurements to an object of a known size. This technique has been demonstrated with optical navigation for cislunar space missions. If the size of an object in the FOV is known, the range to that object can be estimated by measuring the size of its projection on the camera sensor. Given the standard form factor of CubeSats, this technique may work well for VANTAGE. Machine vision techniques can be used to extract CubeSat features such as edges measuring 10 cm, allowing for a reasonable estimate of range. A high frame rate camera would be used to gather as much data as possible to provide accurate velocity estimates from position data. A significant disadvantage of this technique is its dependence on lighting conditions.

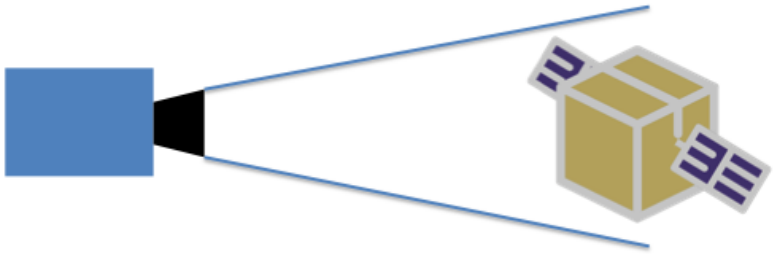

Figure 10. Diagram for a Single-FOV Optical Sensor Option

Pros	Cons
Simplicity	Less accurate range measurements
Low power consumption	Dependence on light conditions
Low cost	

3.2.2. Stereoscopic

Stereo vision techniques can be used to more accurately find the range to an object similar to depth perception in humans. Two cameras similar to that shown in the previous section (though likely lower performance to comply with budget constraints) can be used to triangulate the location of the CubeSat. This method compares features taken from images taken by the two cameras at the same time. The vectors from each camera to a certain object in can be used to loacte the object in three dimensional space. Using the known relative orientation and distance between the two cameras, the distance to an object can be determined with high accuracy at close ranges. However, in order to maximize accuracy of range measurements, the ”baseline distance”, or the distance between the two CubeSats must be large enough to give the cameras a substantially different view of the object . Since our system is constrained by DR 4.1 to a 12cm x 12 cm cross-sectional area pointed toward departing CubeSats, the base distance would be far less than 10 cm and any improvement over a single-camera system would be minimal.

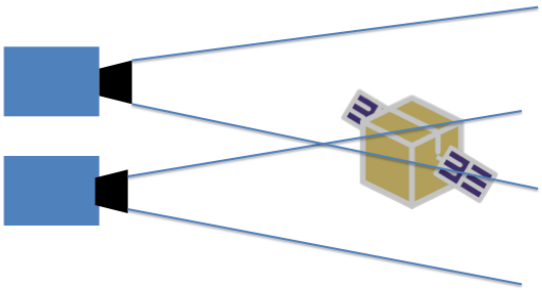

Figure 11. Diagram for a Stereoscopic-FOV Optical Sensor Option

Pros	Cons
Accurate range measurements	Higher software and hardware integration complexity
Possible accurate attitude estimation to augment ranging	Stereo effect limited for our small baseline distance
	Dependence on lighting conditions

3.2.3. Camera with Time of Flight 3d Camera

Time of flight (ToF) sensing is a commonly used technology for 3D mapping. As the name would imply ToF sensors measure the time it takes for a particle to travel from an emitter to objects in the field of view then back to the receiver on the camera. A common type of ToF sensors are pulsed shutter infrared (IR) cameras. These cameras flash an IR light source and measure the magnitude of returned IR radiation at each pixel for short time steps by rapidly pulsing the camera shutter. The magnitude of light returned for each shutter pulse can be used to internally calculate which pixels are at the distance corresponding to the time elapsed since the IR flash.

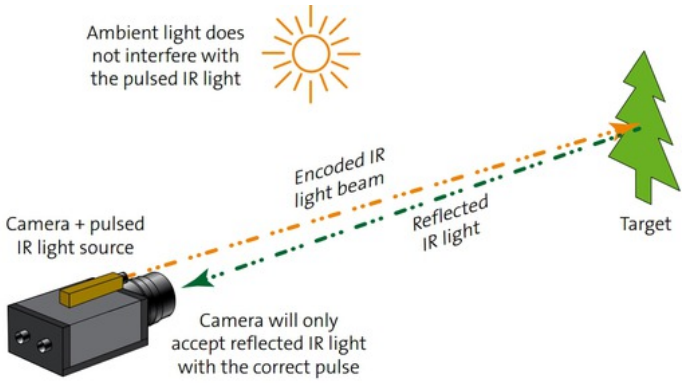

Figure 12. Diagram description of time of flight sensing [12].

Infrared time of flight cameras were initially considered for use in VANTAGE due to their high FOV, superior depth accuracy, and good resolution. These sensors are frequently used in 3d mapping applications because of these advantages. The drawbacks from a ToF camera are high power consumption and low range (10-20m). This architecture will be considered in the following trade study due to it's promising ability to augment close range measurements where a single camera system intended for 100m range could fail to produce meaningful measurements. ToF cameras can often output simple range data which is expected to be easy for the software team to process.

Pros	Cons
Inexpensive 3d mapping	Potential for IR interference
Large field of view	Short range
High depth accuracy	

3.2.4. Camera with Scanning LIDAR

Light detection and ranging (LIDAR) technologies are becoming more and more common in 3d mapping applications across a wide range of industries. They are most commonly used to map terrain or act as the primary sensing for autonomous vehicles. There are several different types of LIDAR systems, but they all operate on the same fundamental principles. A laser emitter produces an electromagnetic wave and emits it into the environment. Once this wave contacts an object it is reflected in some direction. Some of the reflected light returns to the device through a receiving lens and is registered by a photodiode. Meanwhile, a processor on-board keeps track of the amount of time that has passed since the laser was emitted. The total travel time for the light can then be used to calculate the distance travelled by the light.

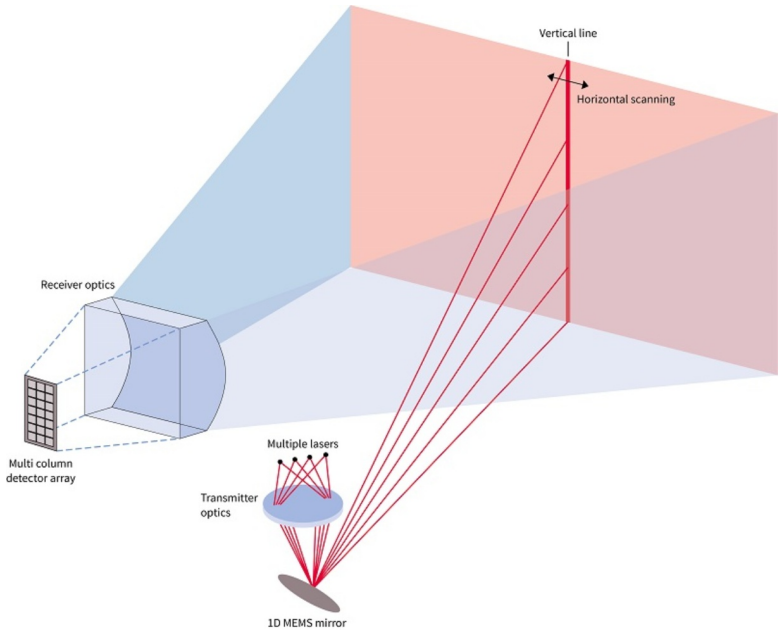

Figure 13. Diagram of scanning LIDAR operations [13].

A scanning LIDAR system diffuses the laser into a thin line then sweeps it over the field of view to create a matrix of points from which the light was reflected back to the receiver. This allows for the generation of 3 dimensional cloud of discrete points which is relatively easy to process from a software perspective. Scanning LIDAR was initially considered due to its frequent use in multi-object recognition applications such as autonomous vehicles, and will be traded against the other sensing architectures in the next section.

Pros	Cons
Large horizontal FOV	Moving parts
Long range	Difficult data format
	Poor vertical resolution

3.2.5. *Camera with Diffuse LIDAR*

Another possible application of LIDAR is the supplemental use of a non-scanning, diffused laser system. This option would work the same as scanning LIDAR, with the exception that it would only return data for a very thin field of view. In effect, it would function as a plane of light that each cubesat would pass through only once. To ensure that every cubesat would actually cross this plane the system would need to be mounted externally to the NanoRacks deployer.

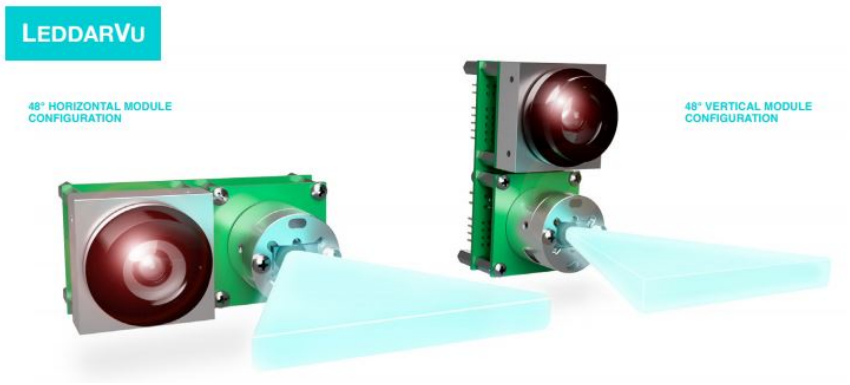

Figure 14. LeddarTech LeddarVu solid-state diffuse LIDAR [14].

This system would need to be mounted externally to the NanoRacks deployer because it has a very narrow vertical field of view. This means that the only place where it could see all deployed cubesats would be above them on the deployer. This diffused LIDAR sensor would also need to be facing down at some angle to capture all of the cubesats in its FOV. In spite of these drawbacks, this sensor will be traded in the following sections because it can meet requirements with exceptionally low power consumption. The data returned from this system is a small set of range measurements that would be extremely easy for the software team to process and integrate.

Pros	Cons
Simple software implementation	Low data-cost ratio
Low power consumption	mounting constraints
Good range	

3.2.6. *Camera with Flash LIDAR*

Flash LIDAR is another possible option that employs diffused laser ranging techniques. Flash LIDAR systems use linear arrays of laser emitters rather than a moving reflector to scan the entire range of the field of view. This linear array of laser emitters flash in rapid succession and the reflected signals are received by an array of photoelectric sensors. This allows the entire FOV to be imaged in an effective instant.

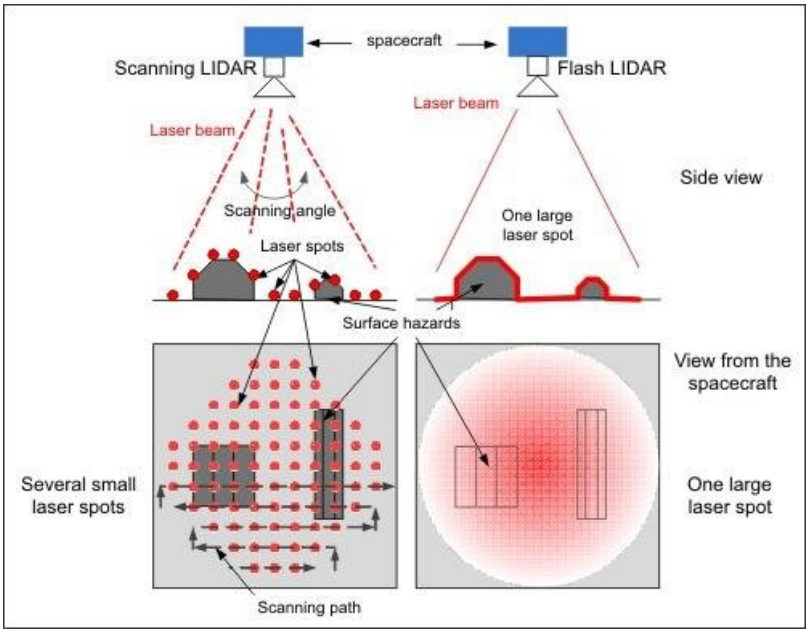

Figure 15. Comparison of flash LIDAR and scanning LIDAR operations[15].

Flash LIDAR was initially considered for this project because it has high range and depth accuracy. It also has the benefit of space flight history; a flash LIDAR sensor was used on the OSIRIS Rex mission for rendezvous operations[16]. While space readiness is far from a requirement for this project, it is certainly something to be considered for the benefit of the group that may inherit it the future. A major drawback to this system is the lack of commercial options available. There are only a handful of companies that produce these sensors, and that greatly restricts our options when purchasing parts down the road. Unfortunately, it also means that these sensors are quite expensive. A quote was obtained for a flash LIDAR system that would fit within our size constraints, and the quoted cost was $65000. This eliminates flash LIDAR from the trade study in the following section because we can't spend 1300% of our budget on a sensor that doesn't meet all of requirements in the first place.

Pros	Cons
High depth accuracy	Expensive
No moving parts	Limited stocked commercial options
Long range options	

3.2.7. Camera with Photoelectric Sensor

The final sensor architecture considered was a camera supplemented by a photoelectric sensor. A photoelectric sensor is not largely dissimilar from the other active sensors discussed in this section. Photoelectric sensors emit a laser towards the object of interest and capture the reflected light with a receiver. The major difference between this sensing and LIDAR is the receiver. Photoelectric receivers have an array of sensors, and the reflected signal hits a different sensor depending on the distance between the object and the transmitted laser.

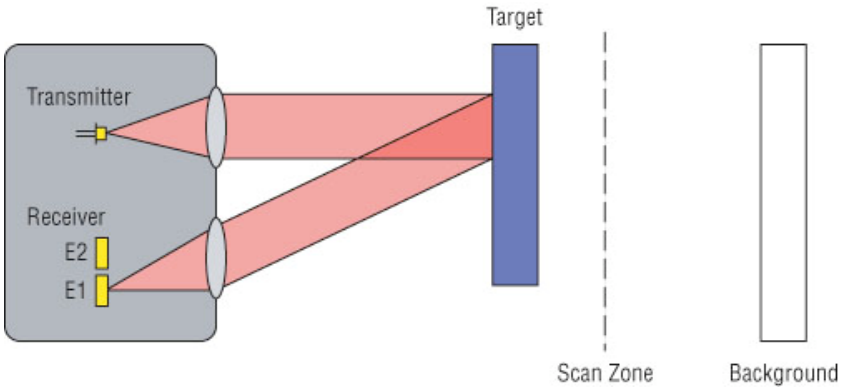

Figure 16. High-level diagram of photoelectric sensor fundamentals.[17].

Photoelectric sensors were initially considered because they are extremely cheap compared to the other sensors considered. They are also readily, commercially available. The major drawback of these sensors is the poor resolution. In fact, they only have single point resolution. This means that VANTAGE would need several pointed in specific directions to capture close range distance measurements. The number of these sensors pointed in different directions required to meet project requirements would not meet the size constraints, so this architecture will not be considered in the following trade study.

Pros	Cons
High accuracy	Single point resolution
Low cost	
Simple operation and integration	

3.3. Avionics

The Avionics design choice will affect the algorithms and software aspects of the VANTAGE project. VANTAGE Avionics will select from several different electronic packages, which are capable of recording all sensor data, calculating the desired position and velocity measurements, and reporting these measurement back to the NanoRacks system through a USB2.0 connection. There will be numerous different design options that are capable of meeting these requirements. However, the ideal processor structure should be easy for the software team to program, and interface well with the sensor package. This section compares different electronics packages that are capable of meeting all of VANTAGE's electronic requirements.

3.3.1. *Multi-Processor System-on-chip (Example Solution: Xilinx Zynq UltraScale+ MPSoC kit)*

Figure 17. Xilinx Zynq UltraScale+ MPSoC kit

MpSoC is a device that includes a Multi-Processor system-on-chip. This chip, shown in figure 17, includes an FPGA, dual-core Cortex-R5, real-time processing unit, and dual-core Cortex-A53. This board is a good option for the VANTAGE team because it is widely supported among commercial, civil and military applications. An example of a civil project that uses this board is a self-driving car machine vision project.[26] This application shows that the board is capable of real-time processing that would help the VANTAGE team obtain velocity and position measurements more quickly. This board has already passed the in-orbit mission radiation performance validation by the US Government. It is therefore a good option for the VANTAGE project, due to the fact that it could be easily transferred to a future space-ready system. One down side to the implementation of this board into the VANTAGE system is its use of hardware description language (HDL). The VANTAGE team lacks experience with this software package and learning to use it would significantly increase the software team's workload.

Pros	Cons
Fast, Reliable	Expensive
Space Usage History	Increased Software Complexity
Low Power Consumption	Embedded code Complexity
Machine Vision Optimization	Lack of Resources at CU

3.3.2. *System on chip (Example Solution: Xilinx Zynq-7000 SoC ZC702)*

Figure 18. Xilinx ZYNQ-7000 SoC ZC702

This System on Chip (SoC) board includes all of the components needed to function as a small computer but does not have an implemented operating system. Due to this lack of operating system, the circuit uses shorter wiring, has less power consumption, and has a faster-processing speed compared to a single board computer. This is because the board does not need to support extra software necessary to run an operating system. The down side to this model is that without an operating system the software team must use HDL to run the board. That being said, MATLAB and

Simulink can be used to generate HDL code that can then be implemented on the board. This would allow the software team to write in a familiar high-level language. Another positive aspect of the Zynq - 7000 is that it is a space graded SoC board and could therefore be used for a space ready iteration of the VANTAGE project. Though this board has many positive aspects, it does not come with a graphics processor capture card and our team would need to purchase and implement one to successfully integrate our sensor package.

Pros	Cons
Fast, Reliable	Expensive
Space Usage History	Software Complexity
Low Power Consumption	Lack of Resources at CU

3.3.3. Cell Phone (Example Solution: Samsung Galaxy S9)

Figure 19. Samsung Galaxy S9

Smart-phones that are being built today are part of a competitive market that strives to combine functionality, processing power, size and cost. One of the biggest advantages to this type of electronic package is that it has already been optimized in ways that are beneficial for our team. This package already has an implemented sensor package so the VANTAGE team would not need to spend time integrating the sensor and electronic systems. Another advantage is that NASA has already tested two phonesat projects in space based on the Google Pixel. Both of these projects were a success. This is a benefit for the VANTAGE team, since a phone could be readily iterated to a space ready application for future teams. One potential challenge of using a Phone is that if our team decides to use any active sensing options, they may be difficult to implement on the electronics package. Another down side to using a phone package is that it is not open source so if VANTAGE wants to make any changes to the electronics or software, it may be very difficult or impossible.

Pros	Cons
Fast	Reliability
Software API package include	Heat generation
Low Power Consumption	Space readiness
– – – – –	Not Open Source
– – – – –	Sensing Options

3.3.4. Next Unit of Computing (Example Solution: Intel NUC)

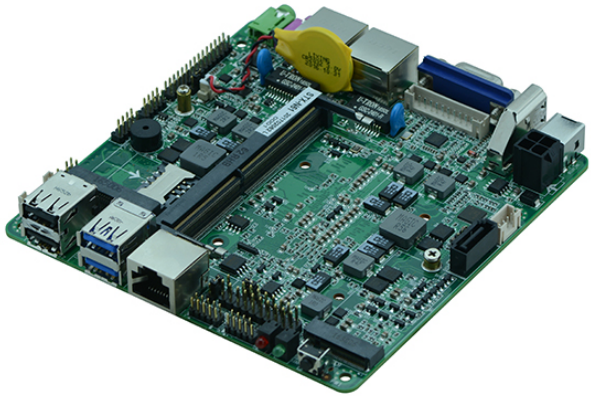

Figure 20. Intel Nuc

Next Unit of Computing (NUC) is potentially a good solution for the VANTAGE electronics package. NUC is a small-form-factor personal computer designed by Intel. This device has been marketed for 8 generations and has become widely available and well supported. This device uses much less Thermal Design Power (TDP) while maintaining a performance similar to a conventional PC. This option contains a wide range of design potential for the VANTAGE team due to its sensor integration capability, and processor power. This system also has the capability to support many high level software languages. One down side to this electronics package is its space readiness. This system uses airflow to cool the processor which produces a significant amount of heat. In space this cooling method would not be practical and another cooling method would need to be used.

Pros	Cons
Fast, Reliable	Expensive
Significant Support for High Level Languages	Space Readiness
Less Unknown Area Develop Time	High Levels of Heat Generation

3.3.5. *Single Board Computer (Example Solution: Raspberry Pi 3)*

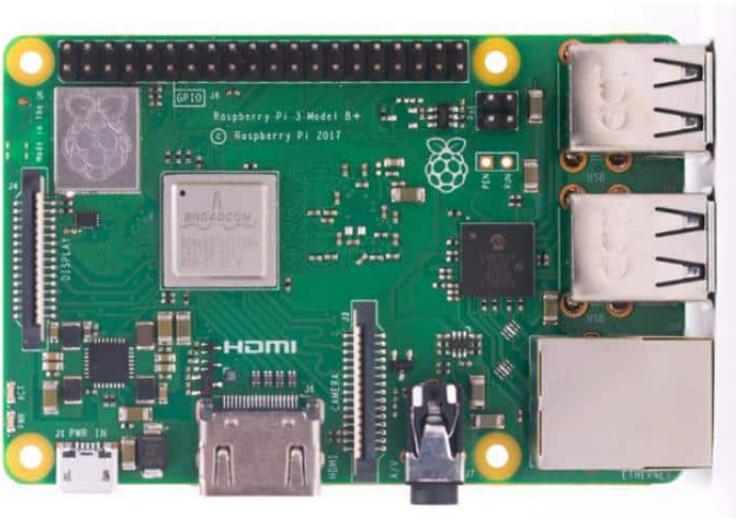

Figure 21. Raspberry Pi 3

Pros	Cons
Cheap	System Performance
Space Usage History	Limited Hardware
Low Power Consumption	

The Raspberry Pi 3 board is a good example of a single board computer. It runs on a Linux-based operating system. The VANTAGE team has minimal experience with these operating systems but learning these systems is still within our design scope. Raspberry Pi may be a good design choice due to its low market price and high level of open source support. This board is also capable of interfacing with many different sensor systems. Raspberry Pi also has a significant history of being used for space applications so would be a good board for future space ready iterations. One down side to this board is that it has very limited processing power. This would increase the time between data capture and processed results. This may also limit our sensor choice to options that do not need as much real time support.

3.4. Design for Structural Interface with NanoRacks

The structural design for VANTAGE describes all intra-mechanical and inter-mechanical interfaces of the planned VANTAGE payload. When considering the structural design for VANTAGE it is useful to reiterate explicitly that flight ready hardware is not being produced. The goal of the structural design is to serve as a demonstration that all physical components can be fit together in a single hardware package. Additionally, this package will interface with a flight-like interface on a ground based NanoRacks hardware model for testing. Additional technical complexity can be added to this by mounting physical components of the VANTAGE system inside the structural housing and demonstrating system functionality from an integrated platform. The NanoRacks CubeSat Deployer (NRCSD) is the cubesat deployment payload tube which houses and launches cubesats from the ISS NanoRacks platform. Figure 22 below shows the NanoRacks on-orbit hardware configuration with an open doors. This is the future planned use platform for VANTAGE which will be simulated by ground based hardware for VANTAGE testing this Spring. VANTAGE will evaluate possible positions either within one NRCSD (eight are seen in figure 22) or in place of one NRCSD. In either case VANTAGE will perform fit checks with hardware supplied by NanoRacks.

The most important overall factor that needs to be evaluated is the placement and position of the VANTAGE payload on the with respect to the other NRCSD tubes. Inherent in this position is how the NanoRacks hardware module operates with the VANTAGE payload as well. Once possible structural interface, discussed in section 3.4.1, required VANTAGE to directly contact other payloads and withstand deployment forces imparted by the push plate

of the NRCSD. All interface options have implications on the system field of view and the performance accuracy of the software. Both of these factors are mathematically addressed later in the trade study. The discussion below concentrates of the pros and cons of each design option, its feasibility, and how it satisfies requirements. Each design option is then evaluated for trade study purposes in section 4.4.

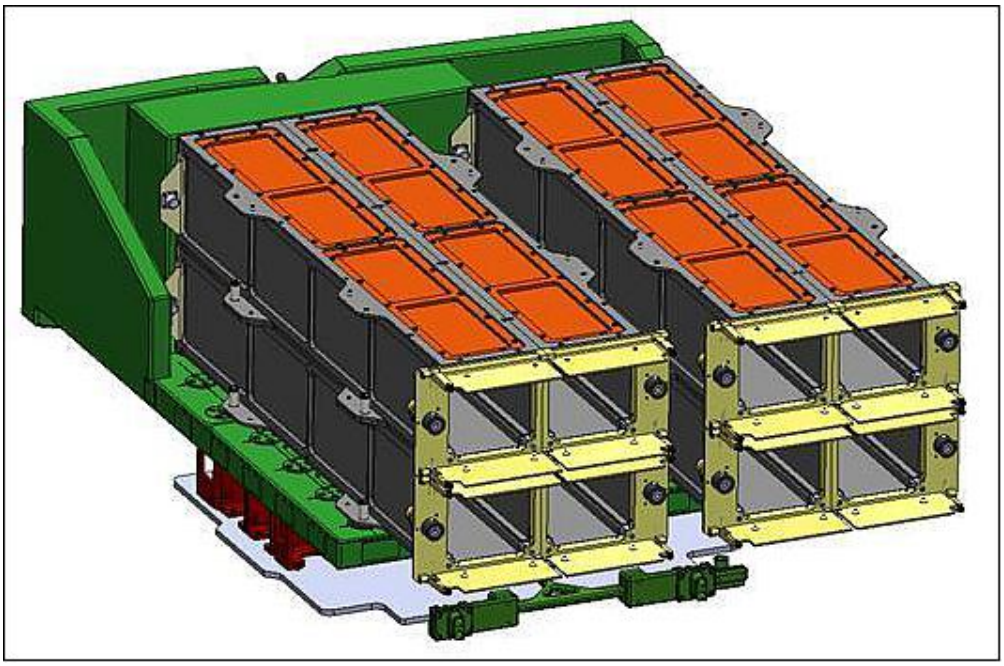

Figure 22. The On-Orbit NanoRacks CubeSat Deployment Platform With Eight NRCSD Payload Tubes

3.4.1. Inside NanoRacks CubeSat Deployer Behind Other Payloads

In this configuration, VANTAGE is inside the ground based test version of the NRCSD and is the last payload in the deployment order. For this configuration the doors of the NRCSD will open, the payloads in front of VANTAGE will be ejected, and then vantage will stop at the end of the payload tube before it is ejected. It is from this position in the payload tube that VANTAGE will begin its main operational sequence to image the cubesats downrange. Figure 23 shows the position of VANTAGE inside the NRCSD hardware before ejection begins and after ejection ends. VANTAGE can be seen in red and the mock payloads are seen in grey. This is an ideal option for NanoRacks because sections of the NRCSD in whih VANTAGE is stored are still able to hold satellites that can be ejected into orbit. Since this is an ideal situation for the future planned use case of VANTAGE it is being evaluated as a potential structural interface location and operation mode.

This design satisfies **FR.4**, **DR.4.1-STR**, and **DR.4.2-STR**. These requirements call out the interface between VANTAGE and the NanoRacks ground based test hardware. The interface control document (ICD) available from NanoRacks defines this interface and will be the basis for all derived lower level requirements. This configuration is feasible, but may have implications for the accuracy of estimated relative positions and velocities of mock payloads. Since the structural interface location and operation mode of VANTAGE based on the physical location of the payload are unique in this structural interface concept, it is considered in the trade study.

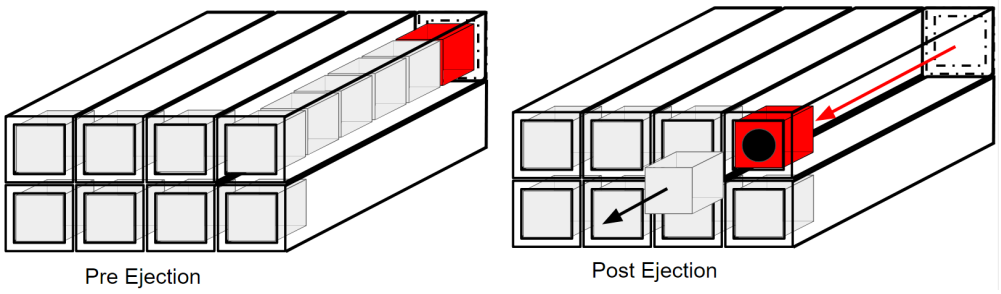

Figure 23. Vantage Position Inside NRCSD Behind Other Payloads

Pros	Cons
No extra volume to move around during maneuvers prior to ejection	Cannot see ejection of payloads in front of the VANTAGE payload
Completely inline with at least 4-5U of ejected payloads	Limited space available for payload either 1U or 2U
Structural interface is clearly defined by the NanoRacks NRCSD ICD	VANTAGE structure must be able to withstand push plate forces since it is acting as the push plate
	May need to design and test a latching mechanism to stop VANTAGE from leaving the deployment tube

3.4.2. Inside NanoRacks CubeSat Deployer Alone

In this configuration, VANTAGE is the only payload in the ground based test version of the NRCSD. Vantage will reside at the front end of the deployment tube and will be able to see outside the deployment tube once the doors are opened. Figure 24 shows the position of VANTAGE in the deployment tube which is unchanged during the deployment sequence of other cubesats. VANTAGE will be mounted in that position inside the deployment tube so that is does not shift during movement prior to or during the simulated deployment sequence.

This design satisfies **FR.4**, **DR.4.1-STR**, and **DR.4.2-STR**. These requirements call out the interface between VANTAGE and the NanoRacks ground based test hardware. This interface must meet almost all of the same internal constraints as the configuration described in section 3.4.1 except that additional constraints will need to be evaluated with NanoRacks to secure the VANTAGE payload in position at the front of the NRCSD deployment tube.

This design is feasible and within the scope of the VANTAGE project since it is essentially the same as the configuration described above. Even though there is some discussion required with the customer to further define the structural interface a majority of this definition already exists.

Again, just as described in section 3.4.1 the location of the VANTAGE payload has major implications on the software development for the recognition of mock CubeSat payloads. Since this is a feasible option for the location of the VANTAGE payload it is included in the trade study.

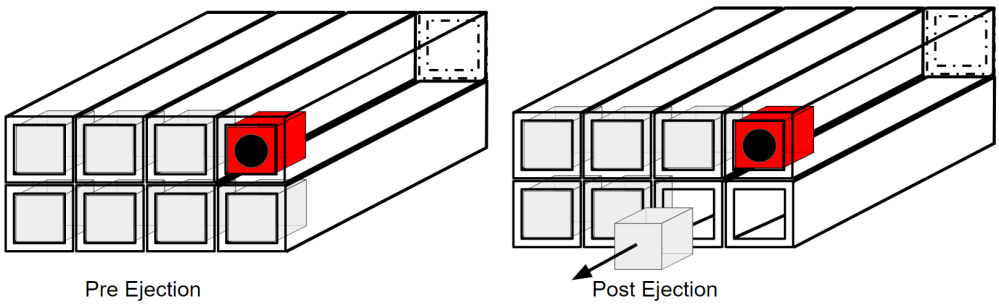

Figure 24. Vantage Position Inside NRCSD Alone

Pros	Cons
Largest available in-tube payload volume	Reduces the number of payload volume available for launch from 48U to 42U
Does not miss any payloads during ejection given that the field of view can see the rest of the NRCSDs	
Structural interface is clearly defined by the NRCSD ICD	

3.4.3. Interface Externally With Other NanoRacks CubeSat Deployers

In this configuration, the VANTAGE payload is acting as the entire NRCDS assembly. Figure 25 shows the position of the VANTAGE payload in the external configuration which is not internal to any deployment tube. The payload instead interfaces directly with the exteriors of the adjacent NRCSD deployment tubes. This allows VANTAGE to have a larger cross sectional area perpendicular to the boresight than when it is internal to the NRCSD tube. VANTAGE remains stationary in this configuration throughout the ejection sequence.

This design configuration satisfies the functional requirement **FR.4** but requires modification of the derived requirements. For this configuration communication with the NanoRacks team will be crucial to determine the parameters of the existing interfaces between NRCSD tubes. Since this interface already exists and is used, this configuration remains in the scope of the project. The interface design still occurs solely on VANTAGE hardware and not modification or engineering is to be done on the NanoRacks NRCSD bank. Some additional work is required to design and build an interface plate that simulates the interface between NRCSD tubes, but this type of design effort is expected for test equipment in all configurations so this slight change should no impact this effort significantly. This is a feasible option for the structural interface and position of the VANTAGE system warranting its inclusion in the trade study space.

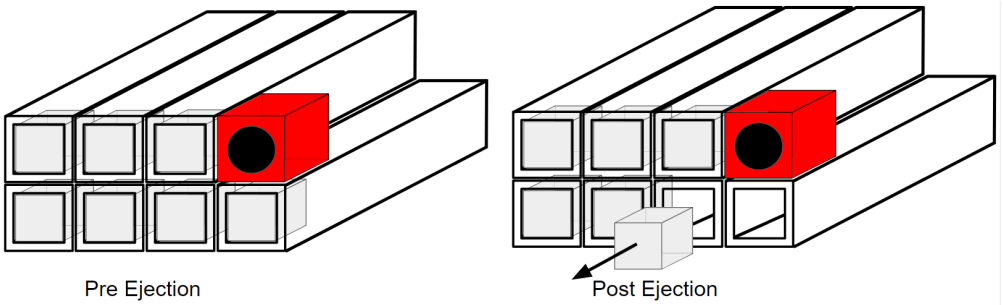

Figure 25. Vantage Position Outside the NRCSD

Pros	Cons
Largest available volume for VANTAGE payload	Need to define the structural interface with NanoRacks since no publicly released ICD currently exists
Largest single face surface area normal to the direction of deployment allowing for the most available surface mount sensor area	Unable to view all deployments with a single forward facing camera
	Reduces the number of payload space available for NanoRack's customers

4. Trade Study Process and Results

This section will present the process and results of the trade studies performed in order to make informed choices between valid key design options from Section 3.

It is pertinent to note why independent algorithms and test-rig trades were not included in the CDD. The reason that an independent algorithms trade was not conducted in the CDD can be seen in Appendix C. The reason that an independent test-rig trade was not conducted in the CDD can be seen in Section 6.

It is also worth pointing out that performing trade studies is more an art than an exact science. For example, the Measures of Success (MoS) selected to compare the various options in a given trade cannot be shown to perfectly compare every aspect of the design space, also while the weights for each MoS were chosen based on team-member experience and customer priority, no proven mathematical theory guarantees that these weights objectively relate the MoS to the success of the project. There is simply too much of a human element in the design process for this to be true. Therefore, a margin of uncertainty will be attributed to the trade study results. The team has elected to represent this margin in the following way: any option whose final trade study score is within 10% ((5-1)/10 = 0.4) of the highest scoring option can also be considered a valid design path for the project, and used as an "off-ramp" in case the first-choice option presents an unforeseen and insurmountable barrier during the preliminary design stage. **Any final scores which are more than 10% (0.4 points) above second-place are considered to be the clear winners of the trade study.** This point will be reiterated in Section 5.

4.1. Programming Language

4.1.1. Measures of Success and Weighting

All measures of success were presented to the customer and confirmed to be representative of both the team's and the customers priorities. Additionally, the relative importance of each measure of success was determined through discussion with the customer (i.e. for Software: "Quality and Availability of Packages/Libraries" ≥ "Implementation Difficulty" ≥ "Speed/Runtime", based on customer input). The exact weights were then determined based on VANTAGE team expertise, which is described in the "Rationale" column of Table 2. Each exact weight was again confirmed by the customer.

Table 2: Measures of Success and Weighting for Programming Languages

	Weight	Driving Req'ts	Rationale
Quality and Availability of Packages/Libraries	55%	DR.5.2, DR.6.1, DR.6.2, DR.7.2, DR.7.3, DR.7.4	It is highly unreasonable for our group to fulfill the software requirements in the given timeframe if designing all software from scratch. Availability of external software, and the quality of that software's documentation will be crucial to accomplishing requirements. If a language is chosen and developed with insufficient outside software, the project will likely fail.

Implementation Difficulty (Learning, Manhours)	30%	N/A	Difficulty in implementation is a crucial component of total manhours spent on a project. If the team spends a significant ratio of their time struggling with debugging/learning/syntax, the efficiency of the project will be unnecessarily decreased compared to choosing a more comfortable language.
Speed/Runtime [18]	15%	DR.7.1	Any given software language will have inherent differences from any other which can lead to significant variation in runtime of the same operations between different languages. A balance will be made between the optimization of our software and the available processing power provided from the electronics and sensors subteams. Ensuring the system is efficient is crucial, but there will likely be sufficient breathing room that allows us space to use more easily developed languages.

4.1.2. Non-Dimensional Scoring System

The measures of success listed above have been non-dimensionalized in Table 3 for the purpose of assigning a final numerical value (1-5) for each language being traded.

The values for Quality and Availability of Packages/Libraries was distributed evenly from a range of 0-100%, in intervals of 20%. Various languages may have no packages available whatsoever, while more of the popular and developed languages might have up to 100% of the overall packages that might be needed. Rank 1 was attributed as 80-100%, since this application will be somewhat specific, so a language lacking in 1 or 2 desired packages with sufficient documentation to be useable is still highly desirable. Conversely, if a language has only 1 or 2 packages that are sufficiently documented relevant to this project, it is likely to drastically increase the manhours necessary to place the software development outside the range of feasibility.

Implementation Difficulty was attributed a similar range, with a ratio of 0-1.0 for the manhours spent struggling with language implementation compared to actual development. A language in which the team struggles enough that 80-100% of the hours are wasted (i.e., learning a language completely foreign to the team, with difficult syntax and poor documentation) will be the poorest choice. Comparing this to a language the group is proficient in, to the point that no hours are lost, or up to 20%, indicates that implementing the language will be very straightforward.

For Speed/Runtime, these values were taken from the NASA Modeling Guru page [18] which compares the languages in reference to Problem 6 (Belief Propagation), which is indicative of applications similar to the VANTAGE project such as computer vision and image processing. To design the values from 1-5, the range of time solutions were examined, placing the reasonable worst time (ignoring the few languages with absurd completion times which were not in consideration) to completion as rank 5. The worst time was then approximated to 20s to allow for an even integer distribution for each of the 5 ranks.

Table 3: Software Non-dimensionalization Ranges for Measure of Success Scoring

Measure of Success	Metric	5	4	3	2	1
Quality and Availability of Packages/Libraries	Percentage (%) of necessary functions/requirements covered by available software	0-20	20-40	40-60	60-80	80-100
Implementation Difficulty (Learning, Manhours)	Ratio of expected manhours lost to implementation difficulty with respect to overall development	0.8-1.0	0.6-0.8	0.4-0.5	0.2-0.4	0.0-0.2
Speed/Runtime [18]	Time (sec) to completion of Image Processing/Computer Vision Algorithm for n=1000 iterations	>20	15-20	10-15	5-10	0-5

4.1.3. *Trade Study*

Table 4 presents the individual scores for each measure of success for each design option in the trade study. Each design option's score for each measure of success was determined by evaluating the characteristics of the design option, described in Section 3.1, using the non-dimensional scoring system presented in Section 4.1.2. It should be noted that this non-dimensional scoring system is set up such that higher numbers indicate high-challenge/low-feasibility while lower numbers indicate low-challenge/high-feasibility on a scale of 5 (bad) to 1 (good). This means that the design option with the lowest final score is the best, or most feasible option for the project.

The bottom row presents the results of the trade study and indicates relative feasibility of each design option based on the following color scheme: green (good), light green (within 10%, or 0.4 points, of first-place), yellow (neutral), red (bad). The results were ultimately calculated by taking the weighted average of the scores for each design option, but a deeper approach was taken to derive and validate this simple method (see Appendix A).

Table 4. Programming Languages Trade Study

	Weight	Python	C++	C	MATLAB
Quality and Availability of Packages/Libraries	55%	1	1	1	1
Implementation Difficulty (Learning, Manhours)	30%	2	2	3	1
Speed/Runtime	15%	3	3	3	2
Total/Result	100%	1.6	1.6	1.9	1.15

4.2. Sensor Suite

4.2.1. *Measures of Success and Weighting*

All measures of success were presented to the customer and confirmed to be representative of both the team's and the customers priorities. Additionally, the relative importance of each measure of success was determined through discussion with the customer (i.e. for the Sensor Suite: "Accuracy" ≥ "Software Development Time" ≥ "Cost" ≥ "Power" ≥ "Hardware Development Time" ≥ "Size", based on customer input). The exact weights were then determined based on VANTAGE team expertise, which is described in the "Rationale" column of Table 5. Each exact weight was again confirmed by the customer.

Table 5: Measures of Success and Weighting for the Sensor Suite

	Weight	Driving Req'ts	Rationale
Accuracy	23%	DR.6.1, DR.6.2	Accuracy is seen as the most important parameter for the sensor. Higher accuracy of depth measurements will lead to more accurate position and velocity measurements. The maximum acceptable error defined by requirement DR 6.1 is 1 m at 10 m range. This parameter is measured in the accuracy (number of sigmas) of range measurements possible with the given instrument. It is highly desirable to be able to improve accuracy of depth measurements because it will improve both relative and velocity measurements.
Software Development Time	20%	N/A	This refers to the amount of time and effort required by the software team to integrate with the sensor package. This parameter was designed to take into consideration the complexity of the data that needs to be processed in order to make range measurements. For example, the stereoscopic choice would require more software development time than the single-camera option. (Complexity of integrating the data from the two sensors). A great deal of team member experience in software development for DigitalGlobe and Harris as well as imaging, algorithm, and sensor-based software development on PolarCube (3U CubeSat) was leveraged when deciding on this weight.

Cost	18%	N/A	Preliminary analysis shows that the sensor system could cost more than half of the team's available budget. While the sensor was expected to be the largest cost to the team, purchasing a sensor for more than half the budget is seen as a risk.
Power	16%	DR.3.1, DR.3.2	Power was seen as a fairly significant driver of our design. Since VANTAGE will have limited power availability (DR 3.2) and a significant amount of power has to be left to the electronics system for processing and system monitoring, power was considered an important consideration. Some of the options considered consume 20 Watts power, a significant portion of the available, thus the power consumption of the sensor package was seen as a significant design driver.
Hardware Development Time	13%	N/A	This refers to the amount of time and effort required to integrate the sensor package into the system. Particularly, this refers to the integration with the electronics subsystem and calibration of the instruments. For example, dealing with two different types of sensors may be a challenge for the electronics subsystem because it may reduce rate of sensor measurements or require that both sensors be operated simultaneously.
Size	10%	DR.4.1	Size, measured in volume of the sensor package, was seen as the least significant consideration for our design choice. Given the fact that VANTAGE does not have to conform to the CubeSat form factor, size was seen as a minimally important consideration.

4.2.2. Non-Dimensional Scoring System

The measures of success for the sensing architecture are made non-dimensional to the standard 5-1 scale used for all of our trade studies. The full range of values between 5 and 1 are originally determined based on the maximum and minimum values for the considered design options (Section 3) for each measure of success. This full range is not spanned completely by the options in the actual trade because it was based on the entire set of design options considered even if they were not traded on. The one exclusion to this range determination is the cost. Despite being considered as a key design option, the cost for the most expensive sensor (flash LIDAR) was not used as the maximum for the cost measurement of success. This is because it was drastically larger than the project budget. With this in mind, the absolute maximum allowable cost was determined based on the cost ranges expected from other subsystems. Explicitly, the maximum structural cost was $1000 and all but one of the avionics options were also less than or equal to $1000. Based on those values and our budget of $5000, $3000 was selected as the upper range for cost in the trade study.

With the exception of accuracy, the bins for the non-dimensionalization were linearly distributed between these minimum and maximum values. For accuracy, it made more sense to do a non-linear attribution of bins because the actual distribution of values was not linearly distributed. A linear distribution would have been a poor representation for the varying accuracy values. For example: IR ToF sensors can have accuracy up to 20mm at close range where the next best, scanning LIDAR, has an accuracy of 3cm. A linear scaling would have put these sensors in adjacent bins, and that fails to realistically depict the difference between the sensor capabilities. The bins were chosen in such a way to fully distinguish the sensors based on depth accuracy while also loosely following a exponential scale because the distribution of sensor capabilities was better matched by this scaling.

Table 6: Sensor Suite Non-dimensionalization Ranges for Measure of Success Scoring

Measure of Success	Metric	5	4	3	2	1
Accuracy	Depth accuracy at 10 m (cm)	>10	5-10	2-5	1-2	0-1
Software Development Time	hours	>100	80-100	60-80	40-60	<40
Cost	USD ($)	>3000	2500-3000	2000-2500	1500-2000	<1500
Power	W	>20	15-20	10-15	5-10	0-5

Hardware Development Time	hours	>100	80-100	60-80	40-60	<40
Size	cm^3	600-800	400-600	200-400	100-200	0-100

4.2.3. Trade Study

Table 7 presents the individual scores for each measure of success for each design option in the trade study. Each design option's score for each measure of success was determined by evaluating the characteristics of the design option, described in Section 3.2, using the non-dimensional scoring system presented in Section 4.2.2. It should be noted that this non-dimensional scoring system is set up such that higher numbers indicate high-challenge/low-feasibility while lower numbers indicate low-challenge/high-feasibility on a scale of 5 (bad) to 1 (good). This means that the design option with the lowest final score is the best, or most feasible option for the project.

The bottom row presents the results of the trade study and indicates relative feasibility of each design option based on the following color scheme: green (good), light green (within 10%, or 0.4 points, of first-place), yellow (neutral), red (bad). The results were ultimately calculated by taking the weighted average of the scores for each design option, but a deeper approach was taken to derive and validate this simple method (see Appendix A).

In Table 7, "Stereoscopic" indicates a two-camera system with overlapping fields of view, "Cam+IR ToF" indicates a single camera and an infrared time-of-flight camera, "Cam + Diffuse LIDAR" indicates a single camera and a diffuse LIDAR system, "Cam+Scan LIDAR" indicates a single camera and a scanning LIDAR system.

Table 7. Sensor Suite Trade Study

	Weight	1 Camera	Stereoscopic	Cam+IR ToF	Cam + Diffuse LIDAR	Cam+Scan LIDAR
Accuracy	23%	5	4	1	4	3
Software Development Time	20%	2	5	3	3	4
Cost	18%	2	5	4	4	4
Power	16%	1	2	4	3	3
Hardware Development Time	13%	2	3	3	2	4
Size	10%	1	2	4	3	4
Total/Result	100%	1.46	2.76	1.95	2.24	2.64

4.3. Avionics

4.3.1. Measures of Success and Weighting

All measures of success were presented to the customer and confirmed to be representative of both the team's and the customers priorities. Additionally, the relative importance of each measure of success was determined through discussion with the customer (i.e. for Avionics hardware: "Space Certifiable" ≥ "Performance" ≥ "Development Time" ≥ "Cost" ≥ "Power", based on customer input). The exact weights were then determined based on VANTAGE team expertise, which is described in the "Rationale" column of Table 8. Each exact weight was again confirmed by the customer.

Table 8: Measures of Success and Weighting for Avionics

	Weight	Driving Req'ts	Rationale

Space Certifiable	30%	N/A	A major aspect of the project is to design the VANTAGE payload such that there is a clear path towards eventual (not this year) compatibility with the NanoRacks use-case system (FR.2, FR.3, FR.4). A related goal is therefore to select avionics hardware which satisfy one of three criteria: 1) avionics hardware has spaceflight heritage, 2) avionics hardware is a model similar to models with spaceflight heritage or 3) avionics hardware is space rated.
Performance	30%	DR.1.4, DR.8.1, DR.8.2	This is a lumped measure of success calculated by taking the average of scores for characteristics such as processor memory (RAM), data memory (ROM), Processor Speed (MIPS), and supported programming languages. This was assigned a relatively high weight because higher accuracy position and velocity measurements require a greater number of high-resolution sensor measurements which requires high throughput on the system processor. Additionally, faster processing speeds will reduce software development time by enabling faster iterations on software failures/debugging. Avionics performance will also enable the software team a wider range of options in development language choice.
Development Time	25%	N/A	Development time refers to the amount of time required to integrate avionics with power/data interfaces and the sensor suite. It was assigned a relatively high weight because avionics development must be finished quickly in order to allow for maximized software development time, which is expected to be the longest subsystem development time for the project.
Cost	10%	N/A	Cost refers to the effect of the avionics hardware on the team's finite budget. Avionics hardware is expected to take up a significant portion of the budget, but cost is still a factor and extravagance cannot be afforded. For these reasons cost was assigned a low-middling weight.
Power	5%	DR.3.1, DR.3.2	Avionics hardware is expected to be one of or even the greatest power draw for the VANTAGE payload. While officially NanoRacks has informed the team that 520 W is the maximum allowable power draw, thermal effects should be considered. This relatively low weight reflects the team's large power budget and the desire not to select avionics hardware which will make thermal control nearly impossible for future iterations of this project.

4.3.2. Non-Dimensional Scoring System

Similar to the scoring systems described in previous sections, the measures of success for electronics are given on a range from 1-5 using a non-dimensionalized scoring system. As seen in table 8, one of the most important factors in choosing an electronics board is its ability to be transferred to a space certifiable system for future iterations of this project. For this measure of success, the ideal board would be chosen such that it is already space-rated and does not need to be modified before integrating it into a future space ready system. The least desirable board for this measure of success would be one that meets all requirements but could not be converted to a space ready system.

The next most weighted measure of success for the electronics system is the board performance. This measure of success effects the supported software, memory size, and processor speed. Due to this dependence, the performance metric is broken down into 4 sub-categories that each contain their own scoring system. Each of these sub-categories is equally important to the effectiveness of the board performance and will therefore be averaged to obtain an overall performance score. The first sub-category ranges from less than one GB of RAM to more than 16 GB of RAM. 16 GB was chosen based on considering a 16 bit processor which is commonly seen among many PCs today. Each metric level from 5-1 is increased linearly by 4 GB of RAM to correspond with commonly produced Processor Memory. Another sub-category of the board performance is the Data Memory Size (ROM). This metric ranges from 32-256 GBs in size. The ROM is doubled for each new score to be consistent with common memory cards available on the market. Since the VANTAGE system is required to store at least 2 raw images of at most 6 CubeSats as well as processed trajectory data and maintain a decent overhead for flexibility, back-of-the-envelope calculations show that the electronics board needs at least 32 GB of ROM. This was derived from at least 24GB necessary to record 2

minutes of 4K video. Common production only builds memory at 16 or 32GB. An ideal board would contain more than 256 GB of ROM which would give the vantage team the design room to record and store high quality video over a 90 minute period. The third sub-category for board performance is the processor speed. The VANTAGE team has decided to record this metric in MIPS or Mega Instructions Per Second. This unit is chosen instead of Bits Per Second (BPS) based on the fact that two machines with the same BPS and different processor structures may have very different processing speeds. The VANTAGE electronics board must have at least 1000 MIPS of processing speed to run a low level program that processes at least 12 images in less than 90 minutes. A more ideal board would contain more than 10^5 MIPS which would allow for 90 minutes worth of high quality video footage to be processed in under 90 minutes. The numeric values for this category do not scale linearly with score. Instead they scale semi-exponentially to be consistent with the exponential increase in processor speed necessary to move from processing a few images to processing high-quality video footage. The final sub-category for electronics board performance is the programming language that the processor supports. The ideal supported software package is one that can run high level and low level languages such as MATLAB and Python on a Linux-based platform. This broad language support would provide our software team a large level of design flexibility and reduced development time due to using a familiar programming language. The minimum language support that the VANTAGE team could work with are low-level and compiled languages such as C and C++. These languages are necessary to maintain a reasonable software development time for processing high quality images.

After being Space Certifiable and having good Performance, the board Development Time is the most important measure of success. The metric chosen for this measure of success was based on looking at previous project schedules that are similar to the VANTAGE project. Successful teams began to develop software by the third week of the second semester of senior projects. Due to the needs of the software team, the Electronics board would ideally be ready for software development in less than two weeks. Worst case scenario, the software team could develop their program on another system and implement it on the electronics board a month and a half into the second semester of senior projects.

Another important measure of success for the electronics board is the cost. Worst case scenario the electronics board would cost more than 2500 dollars which is more than half of our project budget. This option, although feasible would leave our team little design room. A more ideal board cost would be less than 250 dollars which is a reasonable price range for a small PC. The metric values are scaled linearly increasing by 500 dollars for each score.

The last measure of success for the electronics board is the power consumption. This metric ranges from less than 12 to more than 100 watts. Most electronics boards run on 12 watts which would increase our design space to leave power for other devices such as our sensor. The board could consume more than 100 watts while still remaining under the 520 watt maximum load specified by NanoRacks. This metric does not scale linearly because a far greater number of devices operate at lower power consumption than at very high power consumption.

Table 9: Avionics Non-dimensionalization Ranges for Measure of Success Scoring

Measure of Success	Metric	5	4	3	2	1
Space Certifiable	N/A	Capabilities Known to be Inconsistent with Space-Rated hardware	No Known Space-Heritage	Similar Models have Space-Heritage	Space-Heritage	Space-Rated
Performance		Mean of following sub-categories				
Processor Memory (RAM)	GB	<1	1-4	4-8	8-16	>16
Data Memory Size (ROM)	GB	<32	32-64	64-128	128-256	>256
Processor Speed	MIPS*	<1000	1000-5000	5000-10^4	10^4-10^5	$>10^5$
Supported Languages	N/A	Compiled languages only (C++, C, etc) SoC-based and FPGA	Compiled languages only (C++, C, etc) Linux-based	High-level languages with MATLAB FPGA	High- and low-level languages can be supported with some changes	High- and low-level Language support: MATLAB, Python, C++, Linux-based
Development Time	weeks	>6	4-6	3-4	2-3	<2
Cost	USD ($)	>2500	1000-2500	500-1000	250-500	<250
Power	watts	>100	60-100	30-60	12-30	<12

*Millions of Instructions Per Second

4.3.3. Trade Study

Table 10 presents the individual scores for each measure of success for each design option in the trade study. Each design option's score for each measure of success was determined by evaluating the characteristics of the design option, described in Section 3.3, using the non-dimensional scoring system presented in Section 4.3.2. It should be noted that this non-dimensional scoring system is set up such that higher numbers indicate high-challenge/low-feasibility while lower numbers indicate low-challenge/high-feasibility on a scale of 5 (bad) to 1 (good). This means that the design option with the lowest final score is the best, or most feasible option for the project.

The bottom row presents the results of the trade study and indicates relative feasibility of each design option based on the following color scheme: green (good), light green (within 10%, or 0.4 points, of first-place), yellow (neutral), red (bad). The results were ultimately calculated by taking the weighted average of the scores for each design option, but a deeper approach was taken to derive and validate this simple method (see Appendix A).

In Table 10, "MpSoC" indicates a Multi-Processor System-on-Chip (e.g. Xilinx Zynq UltraScale+ MPSoC kit), "SoC" indicates a System-on-Chip (e.g. Xilinz Zynq-7000 SoC ZC702), and "NUC" indicates Next Unit of Computing (e.g. Intel NUC).

Table 10. Avionics Trade Study

	Weight	MPSoC	SoC	Cell Phone	NUC	Raspberry Pi
Space Certifiable	30%	1	1	4	3	2
Performance*	30%	2	3.3*	1.7*	1	3.3*
Processor Memory (RAM)	N/A	1	3	3	1	4
Data Memory Size (ROM)		1	4	1	1	2
Processor Speed		1	3	1	1	5
Supported Languages		5	3	2	1	2
Development Time	25%	5	5	1	1	3
Cost	10%	4	3	3	3	1
Power	5%	2	1	1	3	1
Total/Result	100%	2.65	2.88	2.32	1.90	2.48

*Non-integer values for the Avionics Measure of Success (MoS): Performance occur because this single MoS is the average of four performance parameters, see Section 4.3.2.

4.4. Design for Structural Interface with NanoRacks

4.4.1. Measures of Success and Weighting

All measures of success were presented to the customer and confirmed to be representative of both the team's and the customers priorities. Additionally, the relative importance of each measure of success was determined through discussion with the customer (i.e. for Structural Interface: "Algorithm Implications" ≥ "Development Time" ≥ "Cost" ≥ "Available Volume" ≥ "Maximum Tolerance", based on customer input). The exact weights were then determined based on VANTAGE team expertise, which is described in the "Rationale" column of Table 11. Each exact weight was again confirmed by the customer.

Table 11: Measures of Success and Weighting for Structural Interfaces

	Weight	Driving Req'ts	Rationale

Algorithm Implications	60%	DR.1.3, DR.5.2, DR.6.1, DR.6.2, DR.7.2, DR.7.3, DR.7.4	The algorithm implications of certain structural interfaces is a measure of success because five different high level requirements are connected to it. For this reason the algorithm implications category is given a significantly higher weighting , 60%, because it is highly dependent for the overall success of the project. The algorithm will depend heavily on the geometrical and time varying constraints of the structural interface with respect to the ground based NanoRacks mock hardware and thus should be the majority driver in the trade study of structural interfaces.
Development Time	20%	N/A	Development time is measure of success because a design needs to exist in a ready to manufacture state prior to the end of the Fall semester. This is the deadline enforced by CDR and when manufacturing is scheduled to begin. This schedule is the major driver for how development times for different structural interfaces were determined. In all the cases evaluated in the structural interface trade study the approximate dimensions and location of the VANTAGE payload are similar. Additional, experience on CUE3 and MAXWELL (both CubeSat missions being developed for flight through graduate projects) can be applied in the sense that bus design without complex mechanisms or complex loading and thermal considerations is straightforward. This anticipated similarity and lack of complex mechanisms or complex thermal and loading considerations informs the lower weighting percentage of 20%.
Cost	10%	N/A	Cost is also a performance measure but it is not a major factor contributing to the structural interface trade study. Based on previous experience on CUE3 and MAXWELL the anticipated material use and cost of VANTAGE is under $1000 which is the preliminary budget allocation for the structure. Especially since all machining is expected to take place in-house the only expected costs for the structure are stock material and fasteners. Based on these expectations the cost measure of success is only 10%.
Available Volume	5%	DR.4.1	The available volume is a success parameter because it deals directly with a requirement. The DR.4.1-STR requirement is strictly from a structural perspective and concerns the physical maximum physical dimensions of the payload dictated by an ICD. Available volume is only weighted at 5% because it represents a relatively less significant part of the design effort. The anticipated volume of the VANTAGE payload, no matter the structural interface chosen, will likely be close to 2U (10cm x 10cm x 20cm). Experience on CUE3 and MAXWELL (both 6U payloads) has shown that a significant amount of COTS components and custom design electronics can be fit into small volumes. This provides confidence in the 2U estimation of the VANTAGE payload no matter the component selection which leads to the 5% weighting in this category.

Maximum Tolerance	5%	DR.4.1, DR.4.2	The maximum tolerance is a measure of success because it deal directly with a requirement. The DR.4.1-STR requirement is strictly from a structural perspective and concerns the physical maximum physical dimensions of the payload dictated by an ICD. These dimensions are governed by a machining tolerance which can drive machining time and costs up. This relationship between maximum tolerance and manufacturing costs is important to consider when choosing the structural interface of the VANTAGE payload. This category is weighted minimally at 5% though because the most stringent tolerance expected is the standard machining tolerance of ±5 thousandths of an inch. Since this is a standard machining tolerance it is not anticipated that a significant amount of additional manufacturing time will be required to build the VANTAGE structure because milling with the CNC machines will easily meet this tolerance requirement.

4.4.2. Non-Dimensional Scoring System

Each of the measures of success for structures are non-dimensionalized based on a scaled range of values which are based on maximum and minimum expected values for each measure of success. The high end of the scale assumes 180 total hours of design. Over the next 10 weeks this results in about 18hrs/wk of design work. The low end of the scale is a total of 60 design hours equating to 6hrs/wk until the end of Fall semester. The non-dimensionalized ranges are included below in table 12. Non-dimensionalized cost measurements are based off a preliminary allocation of $1000 from the project budget. The available volume non-dimensionalization is based on the maximum and minimum available volume values for the three different structural cases where the maximum value is the maximum available volume and the minimum is the smallest. Maximum tolerance has a high value of 2 thousandths of an inch which represents the tightest tolerance that any structural component might have to meet. The lowest end of the range is 0.02 in which represents a very loose machining tolerance. The Non-dimensionalized algorithm manhours were scaled in a similar fashion to normal manhour metrics in this document, where it scales off of the low to high approximation of manhours needed to design and implement the corresponding algorithms needed to meet requirements given the structural choices. The overall software algorithm was broken into the independent sub-component algorithms required, and each of those sub-categories had their own manhour metrics. The sum of the manhour approximations of all the sub-component algorithms gave the general full range of the non dimensionalized metrics used. Most of the algorithms sub-components for all of the design options are archetypal to the problem itself, with heuristics being the most volatile to varying design options. For the design parameters that would require additional algorithms needed, the manhours for those additional algorithms to be implemented only affect the score of the design option, not the overall metric.

Table 12: Structural Interface Non-dimensionalization Ranges for Measure of Success Scoring

Measure of Success	Metric	5	4	3	2	1
Algorithm Implications	Expected algorithm related development manhours	164-190	132-156	108-132	84-108	60-84
Development Time	Expected manhours (hours)	156-180	132-156	108-132	84-108	60-84
Cost	USD ($)	800-1000	600-800	400-600	200-400	0-200
Available Volume	cm^3	7200-9000	5400-7200	3600-5400	1800-3600	0-1800
Maximum Tolerance	in	0-0.004	0.004-0.008	0.008-0.012	0.012-0.016	0.016-0.020

4.4.3. *Trade Study*

Table 13 presents the individual scores for each measure of success for each design option in the trade study. Each design option's score for each measure of success was determined by evaluating the characteristics of the design option, described in Section 3.4, using the non-dimensional scoring system presented in Section 4.4.2. It should be noted that this non-dimensional scoring system is set up such that higher numbers indicate high-challenge/low-feasibility while lower numbers indicate low-challenge/high-feasibility on a scale of 5 (bad) to 1 (good). This means that the design option with the lowest final score is the best, or most feasible option for the project.

The bottom row presents the results of the trade study and indicates relative feasibility of each design option based on the following color scheme: green (good), light green (within 10%, or 0.4 points, of first-place), yellow (neutral), red (bad). The results were ultimately calculated by taking the weighted average of the scores for each design option, but a deeper approach was taken to derive and validate this simple method (see Appendix A).

Table 13. Structural Interface Trade Study

	Weight	Inside NRCSD Behind CubeSats	Inside NRCSD Alone	External to NRCSD
Algorithm Implications	60%	4	3	3
Development Time	20%	3	5	5
Cost	10%	3	3	4
Available Volume	5%	5	3	1
Maximum Tolerance	5%	5	5	1
Total/Result	100%	3.80	3.50	3.30

5. Selection of Baseline Design

This section will analyze and summarize the baseline design for the VANTAGE project which has been determined through the trade studies in Section 4. Trade study scores are on a scale from 1 to 5, and in our case: **the lower the score, the better**.

It is worth pointing out that performing trade studies is more an art than an exact science. For example, the Measures of Success (MoS) selected to compare the various options in a given trade cannot be shown to perfectly compare every aspect of the design space. Also, while the weights for each MoS were chosen based on team-member experience and customer priority, no proven mathematical theory guarantees that these weights objectively relate the MoS to the success of the project. There is simply too much of a human element in the design process for this to be true. Therefore, a margin of uncertainty will be attributed to the trade study results. The team has elected to represent this margin in the following way: any option whose final trade study score is within 10% ((5-1)/10 = 0.4) of the highest scoring option can also be considered a valid design path for the project, and used as an "off-ramp" in case the first-choice option presents an unforeseen and insurmountable barrier during the preliminary design stage. **Any final scores which are more than 10% (0.4 points) above second-place are considered to be the clear winners of the trade study.**

5.1. Programming Language Selection and Analysis

The first-place option from the Programming Language trade study is **MATLAB with a score of 1.15**, compared to Python's score of 1.6, C++'s score of 1.6, and C's score of 1.9. The success of this option can be attributed to its all-around excellent scores in the trade study. MATLAB possesses excellent package documentation and existing code-base, has extremely low implementation difficulty, and is a high-speed language for image processing and object recognition. No other option came within 10% (0.4 points) of the first-place score. VANTAGE will move forward with MATLAB as the selection for a baseline programming language.

5.2. Sensor Suite Selection and Analysis

The first-place option from the Sensor Suite trade study is a **single-camera system with a score of 1.46**, compared to a single-camera and infrared time-of-flight camera's score of 1.95, single-camera and diffuse LIDAR's score of 2.24, single-camera and scanning LIDAR's score of 2.64, and stereoscopic dual-camera system's score of 2.76. The

success of this option can be attributed to its ease-of-use and resource requirements. While the camera scored in the lower-most bracket with respect to accuracy in depth-measurement, it is extremely low in cost, power consumption, size, and software/hardware development time. No other option came within 10% (0.4 points) of the first-place score. The VANTAGE team will move forward with a single-camera system as the selection for a baseline sensor suite.

5.3. Avionics Selection and Analysis

The first-place option from the Avionics trade study is a **Next Unit of Computing (NUC) device with a score of 1.90**, compared to a cell phone's score of 2.32, a Raspberry Pi's score of 2.48, a Multi-Processor System-on-Chip's score of 2.65, and a System-on-Chip's score of 2.88. The success of this option can be attributed to its all-around middle-to-high scores in the trade study. It possesses excellent performance and extremely low development time and is middle-of-the-road in cost, power consumption, and the ability to be space certified. No other option came within 10% (0.4 points) of the first-place score. VANTAGE will move forward with the NUC as the selection for a baseline design.

5.4. Design for Structural Interface with NanoRacks Selection and Analysis

The first-place option from the Design for Structural Interface with NanoRacks trade study is **mounting external to the NanoRacks CubeSat Deployer (NRCSD) tube with a score of 3.30**, compared to mounting inside the NRCSD alone with a score of 3.50 and mounting inside the NRCSD behind other CubeSats with a score of 3.80. The success of this option can be attributed to the external mounts very large available volume and relatively high maximum machining tolerance, which outweighed its relatively high cost and development time as well as its middling algorithm implications.

While mounting externally did win the trade study, the result for mounting inside the NRCSD alone is within the 10% (0.4 point) margin of uncertainty for these trade studies. Mounting alone internally possesses similar algorithm implications and development time at slightly reduced cost but with a smaller available volume and much tighter maximum machining tolerances. VANTAGE will move forward with designs to mount externally to the NRCSD as the selection for the design for structural interface with NanoRacks, but mounting alone internally to the NRCSD can be used as a viable "off-ramp" should mounting externally pose unforeseen and insurmountable barriers to the project during the preliminary design stage.

5.5. Summary

The baseline design consists of a Next Unit of Computing (NUC) central processing board for avionics, which will be running software written in MATLAB and operating a single-camera sensor suite. The VANTAGE payload structure will be designed to integrate externally to the NanoRacks use-case system in the place of one of the NanoRacks CubeSat Deployer (NRCSD) tubes in the eight-tube assembly. As VANTAGE enters the preliminary design stage, this baseline design will be analyzed for feasibility at a deeper level than at the scope of the CDD. If necessary, second-place trade study options can be used as "off-ramps" in the case of unforeseen and insurmountable barriers to first-place options.

6. Looking Forward to PDR

The team recognizes that a major aspect of this project's goals is to produce a test-rig capable of carrying out the functions described by the team CONOPS, Figure 2, and that this test-rig has not yet been clearly defined. **There are two main reasons why the test-rig was not included in the CDD trade studies:** 1) the explicit purpose of the CDD is to define the conceptual design of the VANTAGE system, and the test-rig is not explicitly a part of the VANTAGE design concept, and 2) the exact form of the test-rig and the test environment depend heavily on the type of sensor suite and design for structural interface selected. This second reason actually made it nearly impossible to develop a trade study for the test-rig in parallel with trade studies on the sensor suite and design for structural interface.

The success of this project *does* heavily depend on a well designed test-rig, and the team acknowledges that extensive design and/or trade studies must be performed in order to develop this rig. **Therefore, the test-rig will be investigated and discussed extensively in the PDR.**

References

[1] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., and Kim, T.-K., "Multiple Object Tracking: A Literature Review" Available: https://arxiv.org/pdf/1409.7618.pdf

[2] Matthies, L., Szeliski, R., and Kanade, T., "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences ," Jan. 1988.

[3] Clark, Stephen, "India lofts a record 104 spacecraft on a single rocket" Feb. 2017. Available: https://spaceflightnow.com/2017/02/15/india-lofts-a-record-104-spacecraft-on-a-single-rocket/

[4] Yu, S.-I., Meng, D., Zuo, W., and Hauptmann, A., "The Solution Path Algorithm for Identity-Aware Multi-Object Tracking" Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/app/S16-09.pdf

[5] Huang, X., Wu, F., and Huang, P., "Moving-object Detection Based on Sparse Representation and Dictionary Learning" Available:

[6] Ahmed, Nisar. Ann and H.J. Smead Aerospace Engineering Sciences, College of Engineering and Applied Science, Retrieved September 13, 2018, from https://www.colorado.edu/aerospace/nisar-ahmed.

[7] Riesing, K., and Cahoy, K., "Orbit Determination from Two Line Element Sets of ISS-Deployed CubeSats," 2015.

[8] Fitzgerald, J., "Joe Fitzgerald," Re: Sputnik-1 orbit parameters Available: https://www.amsat.org/why-is-there-so-much-tle-confusion-when-new-cubesats-are-launched/.

[9] Embedded Vision and Control Solutions with the Zynq UltraScale MPSoC [Digital image]. (n.d.). Retrieved September 23, 2018, from https://www.xilinx.com/video/soc/embedded-vision-and-control-solutions-with-the-zynq-mpsoc.html

[10] Xilinx Zynq-7000 SoC ZC702 Evaluation Kit [Digital image]. (n.d.). Retrieved September 23, 2018, from https://www.xilinx.com/products/boards-and-kits/ek-z7-zc702-g.html

[11] Raspberry Pi 3 B single-board computer, supplied as a press sample by the Raspberry Pi Foundation. [Digital image]. (n.d.). Retrieved September 23, 2018. from https://en.wikipedia.org/wiki/Raspberry_Pi#/media/File:Raspberry_Pi_3_B_(39906369025).png

[12] Diagram of time of flight sensor [Digital image]. (n.d.). Retrieved September 27, 2018, from https://www.stemmer-imaging.co.uk/media/cache/default_image_505/uploads/cameras/sis/gl/glossary-3D-time-of-flight-cameras-1-en.jpg

[13] Diagram of scanning LIDAR operation [Digital image]. (n.d.). Retrieved September 27, 2018, from https://www.infineon.com/export/sites/default/media/products/Sensors/Solid-state-Lidar-concept-of-Infineon_klein.jpg_1898503090.jpg

[14] LeddarTech, "LEDDARVU 8-SEGMENT SOLID-STATE LiDAR SENSOR MODULES" [datasheet]. Retrieved September 27, 2018, from https://leddartech.com/app/uploads/dlm_uploads/2017/12/Spec-Sheets-LeddarVu-12avril2018-ENG-web.pdf

[15] Comparison of flash and scanning LIDAR [Digital image]. (n.d.). Retrieved September 27, 2018, from https://www.researchgate.net/profile/Suddhasheel_Ghosh/publication/261333968/figure/fig3/AS:667860351328267@1536241724587/LiDAR-vs-Flash-LiDAR-technologies-Courtesy-wwwfosternavnet.jpg

[16] "GoldenEye 3D Flash LIDAR™ Space Camera" Retrieved September 27, 2018, from http://www.advancedscientificconcepts.com/products/portable.html

[17] High level diagram of diffused photoelectric sensor fundamentals [Digital image]. (n.d.). Retrieved September 29, 2018, from https://www.automation.com/library/articles-white-papers/sensors-sensing-technologies/fundamentals-of-photoelectric-sensors

[18] Kouatchou, J., "Basic Comparison of Python, Julia, Matlab, IDL and Java (2018 Edition)," NASA Available: https://modelingguru.nasa.gov/docs/DOC-2676

[19] "Stack Overflow Trends," Stack Overflow Available: https://insights.stackoverflow.com/trends?tags=python,c,c,matlab

[20] "MATLAB vs. Python: Top Reasons to Choose MATLAB," Reconstructing an Image from Projection Data - MATLAB & Simulink Example Available: https://www.mathworks.com/products/matlab/matlab-vs-python.html

[21] jerangkungdarat, "Automatic License Plate Recognition System - Matlab (Image Processing Algorithm)," YouTube Available: https://www.youtube.com/watch?v=bnQp7cLk7O0

[22] "Simple Object Detection in 3 lines of Code (OpenCV/Python)," machinelearning1 Available: https://machinelearning1.wordpress.com/2014/08/05/simple-object-detection-in-3-lines-of-code-opencvpython/

[23] Urationc, "Hand Recognition & Finger Counter – OpenCV and C ," Moustapha Saad Available: https://moustaphasaad.wordpress.com/2015/02/10/hand-recognition-finger-counter-opencv-and-c/

[24] Paz, C., Conde, M., Porteiro, J., and Concheiro, M., "On the Application of Image Processing Methods for Bubble Recognition to the Study of Subcooled Flow Boiling of Water in Rectangular Channels," MDPI Available: https://www.mdpi.com/1424-8220/17/6/1448/htm

[25] Dunbar, B., "Technology Readiness Level," NASA Available: https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_accordion1.html.

[26] "ADAS Solutions Powered by Xilinx and the Road to Autonomous Vehicles" Available: https://www.xilinx.com/applications/megatrends/automotive-driver-assist.html.

[27] Vallado, D. A.,McClain, W. D. "Fundamentals of astrodynamics and applications". New York: McGraw-Hill Companies, Inc. 1997, pp.351

[28] NanoRacks CubeSat Deployer Clearance Cone Definition https://nanoracks.egnyte.com/dl/dvVFF3Dguk

[29] "C library for linear algebra & scientific computing," armadillo Available: http://arma.sourceforge.net

[30] "ccv," ccv - a modern computer vision library Available: http://libccv.org

[31] "Computer Vision System Toolbox," Reconstructing an Image from Projection Data - MATLAB & Simulink Example Available: https://www.mathworks.com/products/computer-vision.html

[32] "Image Processing Toolbox," Reconstructing an Image from Projection Data - MATLAB & Simulink Example Available: https://www.mathworks.com/products/image.html

[33] Martone, M., "librsb : A shared memory parallel sparse matrix computations library for the Recursive Sparse Blocks format," librsb Home Page Available: http://librsb.sourceforge.net

[34] "NumPy¶," NumPy - NumPy Available: http://www.numpy.org

[35] "OpenCV library," About - OpenCV library Available: https://opencv.org

[36] "Python Data Analysis Library¶," pandas: powerful Python data analysis toolkit - pandas 0.23.4 documentation Available: https://pandas.pydata.org

[37] Nakayama, H., "MULTI-OBJECTIVE OPTIMIZATION AND ITS ENGINEERING APPLICATIONS," *Dept. of Information Science and Systems Engineering, Faculty of Science and Engineering, Konan University*, URL: https://core.ac.uk/download/pdf/62911127.pdf

[38] Olver, P., *Applied Linear Algebra*, Pearson, London, 2005.

Appendix A Trade Study Theory

In order to properly discuss the methodology behind the trade study, some mathematical background must be developed. First, each major trade decision consists of a set of options available to the group. This is written in set notation as:

$$\mathbb{L} \equiv \{l_i\}_{i=1}^{N_L} \qquad \mathbb{W} \equiv \{w_i\}_{i=1}^{N_W} \qquad \mathbb{S} \equiv \{s_i\}_{i=1}^{N_S} \qquad \mathbb{E} \equiv \{e_i\}_{i=1}^{N_E}$$

where l_i represents the ith Structural **L**ocation choice out of N_L choices, w_i represents the ith **S**oftware choice out of N_W choices, s_i represents the ith **S**ensor choice out of N_S choices, and e_i represents the ith **E**lectronics choice out of N_E choices.

From these design choices, a number of important Performance Measures are determined. The values of the Performance Measures are determined by the characteristics of the design choice. A set of functions may be defined that rates the design choice for each Performance Measure:

$$\mathbb{F}_L \equiv \left\{f_j^L\right\}_{j=1}^{M_L} \qquad \mathbb{F}_W \equiv \left\{f_j^W\right\}_{j=1}^{M_W} \qquad \mathbb{F}_S \equiv \left\{f_j^S\right\}_{j=1}^{M_S} \qquad \mathbb{F}_E \equiv \left\{f_j^E\right\}_{j=1}^{M_E}$$
$$f_j^L : \mathbb{L} \to \mathbb{R} \qquad f_j^W : \mathbb{W} \to \mathbb{R} \qquad f_j^S : \mathbb{S} \to \mathbb{R} \qquad f_j^E : \mathbb{E} \to \mathbb{R}$$

where $f_j^L(l_i)$ represents the jth Performance Measure out of M_L Performance Measures of the ith element of $\mathbb{L}$, $f_j^W(w_i)$ represents the jth Performance Measure out of M_W Performance Measures of the ith element of $\mathbb{W}$, $f_j^S(s_i)$ represents the jth Performance Measure out of M_S Performance Measures of the ith element of $\mathbb{S}$, and $f_j^E(e_i)$ represents the jth Performance Measure out of M_E Performance Measures of the ith element of $\mathbb{E}$.

The purpose of the Trade Study is to choose the most feasible complete set of design choices. A complete set of design choices is known as a Design Architecture and is defined as:

$$\Omega \equiv \left\{\{l, w, s, e\} \,\middle|\, \forall l \in \mathbb{L}, \forall w \in \mathbb{W}, \forall s \in \mathbb{S}, \forall e \in \mathbb{E}\right\}$$

This set of Design Architectures represents all combinations of design choices available and must hold the most feasible Design Architecture.

In order to examine and rate the feasibility of these Design Architectures, they must be transformed from a set of choices to a set of Performance Measures. This set of Performance Measures allows the characteristics of the Design Architectures to be mapped into a Trade Space defined as:

$$\forall \vec{v} \in \Omega, \exists! \vec{u} : \vec{u} \equiv \{\mathbb{F}_L(l), \mathbb{F}_W(w), \mathbb{F}_S(s), \mathbb{F}_E(e)\} \in \mathbb{T} \qquad \mathbb{T} \in \mathbb{R}^{M_L+M_W+M_S+M_E}$$

where $\mathbb{T}$ is the Trade Space consisting of the Performance Measures of the elements of Ω. This Trade Space is where Design Architectures shall be compared. In order to perform this comparison, some of the properties of the Trade Space must be developed. The first of these is the inner product. The Trade Space, $\mathbb{T}$, has an inner product defined as:

$$\left\langle \vec{v}_i, \vec{v}_j \right\rangle = \vec{v}_i^T \Psi \vec{v}_j \qquad \vec{v}_i, \vec{v}_j \in \mathbb{T} \qquad \Psi \in \mathbb{M}_{M_L+M_W+M_S+M_E}$$

This defines the Trade Space as an inner product space. This allows us to define a norm on $\mathbb{T}$ as:

$$\left\|\vec{v}_i\right\| = \sqrt{\langle \vec{v}_i, \vec{v}_i \rangle}$$

It is this norm that will allow us to compare and rate Design Architectures.

In order to fairly judge the feasibility of one Design Architecture over another, a special object, known as a "Utopia Point" must be defined.[37] A "Utopia Point" represents the best case scenario of each individual Performance Measure, something which is usually unobtainable. By using the $\mathbb{T}$ norm to measure the distance between the "Utopia Point" and a certain Design Architecture, the difficulty of said Design Architecture relative to the "Utopia Point" may be determined.[37] It is this distance by which Design Architectures are judged.

In general, the Design Architecture closest to the "Utopia Point" is not always the most preferable choice. In fact, many different Design Architectures will surround this "Utopia Point." In order to quantify this envelope of Design Architectures, an object known as a Pareto Frontier, P, is developed:[37]

$$P(\mathbb{T}) \equiv \{\vec{u}' \in \mathbb{T} : \{\vec{u}'' \in \mathbb{T} : \vec{u}'' \succ \vec{u}', \vec{u}'' \neq \vec{u}'\} = \emptyset\}$$

This is the definition of the Pareto Frontier on Trade Space $\mathbb{T}$. This definition may be read as: "the Pareto Frontier on Trade Space $\mathbb{T}$, $P(\mathbb{T})$, is defined as the set of all elements, $\vec{u}'$ of $\mathbb{T}$ such that the set of all elements, $\vec{u}''$, of $\mathbb{T}$ such that $\vec{u}''$ strictly dominates $\vec{u}'$ where $\vec{u}''$ is not a member of set $\vec{u}'$, is the null set, $\emptyset$." Strictly dominating is defined as having a feature or set of features that, compared to another feature or set of features, is preferable. In this case, preferable is being defined by the $\mathbb{T}$ norm distance of the Design Architecture from the "Utopia Point." Thus, the entire envelope of points immediately surrounding the "Utopia Point" will compose the Pareto Frontier.[37]

In classical Multi-Objective Optimization theory, any solution on the Pareto Frontier may be considered to be a "good decision," and it is generally left to the discretion of the subjective preferences of a human decision maker to choose a final solution from this set.[37]

The first step in performing the process presented above is to define the Performance Measure function sets: $\mathbb{F}_L$, $\mathbb{F}_W$, $\mathbb{F}_S$, and $\mathbb{F}_E$. These functions are each defined by a set of measurable characteristics of a design choice which is normalized and placed on a feasibility scale.[37] One of the problems inherent to this method are the coupled effects within the design choices. For example: the type of Sensor being used might limit the types of Electronics that may be used. Thus, the Sensor choices have an effect on the Electronic choices, and the characteristic of the Electronics should contribute to the selection of Sensors. As it turns out, Location, Electronics, Sensors, and Software are all interdependent. Thus, the exact forms of the functions within each functional set should be a function of every element of a Design Architecture.

This complete interdependence presents a large problem for the Trade Study. Basic combinatorics states that there are $N_L \cdot N_W \cdot N_S \cdot N_E$ unique elements within Ω and, by consequence, $\mathbb{T}$.[38] Obviously, evaluating such a large number of Design Architectures is far too computationally expensive. One of the many ways of decreasing the number of computations required is to project the Trade Space onto a specific set of Performance Measures and determine the Pareto Frontier of this smaller subspace. This can greatly reduce the computational time of the Trade Study at the cost of only considering a subset of points of the full Pareto Frontier. Since points on the Pareto Frontier are still chosen with this method, it technically represents a "good decision." A common practice used to recover the effects of the full dimensionality of the original problem is to choose a region of points in this subspace and evaluate the original Trade Space about this smaller domain to determine a final design.[38] Another method of capturing the effects of the full set of Performance Measures is to include the mean effect of the excluded parameters when considering points within the subspace.[37] This second method shall be used for the purposes of this Trade Study.

There exists a set of heuristics that may be used to determine suitable subsets of Performance Parameters onto which to project the Trade Space. First, any independent set of parameters can always be used as a subspace without any losses.[38] This is due to the fact that, in being independent of the rest of the system, picking a set of preferable options on this subspace still allows any set of options not included within this subspace to be chosen. Should no set of independent parameters exist, as is the case in this particular Trade Study, Performance Measures with direct dependencies should be grouped together.[37] Using this set of Performance Measures, the average effects of the indirect dependencies may be included to capture the effects of the larger system with less losses. This method is extensively used within this Trade Study.

A total of four projections shall be used, corresponding to the four main component decisions of the VANTAGE project. These projections are shown below:

$$V_L = \text{proj}_{B_L}(\mathbb{T}) \qquad B_L = \left\{f^L_{\text{time}}, f^L_{\text{cost}}, f^L_{\text{volume}}, f^L_{\text{tolerance}}\right\} \tag{1}$$

$$V_W = \text{proj}_{B_W}(\mathbb{T}) \qquad B_W = \left\{f^W_{\text{time}}, f^W_{\text{support}}, f^W_{\text{optimization}}, f^W_{\text{maintainability}}\right\} \tag{2}$$

$$V_S = \text{proj}_{B_S}(\mathbb{T}) \qquad B_S = \left\{f^S_{\text{time}}, f^S_{\text{cost}}, f^S_{\text{power}}, f^S_{\text{accuracy}}\right\} \tag{3}$$

$$V_E = \text{proj}_{B_E}(\mathbb{T}) \qquad B_E = \left\{f^E_{\text{time}}, f^E_{\text{cost}}, f^E_{\text{power}}, f^E_{\text{certification}}\right\} \tag{4}$$

This first projection, done with the purpose of making a decision on $\mathbb{L}$, projects $\mathbb{T}$ onto the set of Performance Measures within the set B_L. As may be seen, the functions within B_L are members of $\mathbb{F}_L$. Thus, they only depend on the element of $\mathbb{L}$ being considered. By projecting $\mathbb{T}$ onto B_L, the elements of $\mathbb{L}$ make up the whole of the Pareto Frontier of this subspace. However, the higher order effects of the original Trade Space have yet to be captured. In order to include

these effects, if $\vec{u}$ represents the "Utopia Point", $\forall \vec{v}_l \in \mathbb{T}$ corresponding to some $l \in \mathbb{L}$, the following function is used to define the feasibility of $\vec{v}$:

$$\text{Feasibility}_L(\vec{v}) = \left\| \left\{ \text{proj}_{B_L}(\vec{v}), \text{mean}\left(\left\| \text{proj}_l(V_W) \right\| \right) \right\} - \left\{ \text{proj}_{B_L}(\vec{u}), \text{mean}\left(\left\| \text{proj}_l\left(\text{proj}_{B_W}(\vec{u}) \right) \right\| \right) \right\} \right\| \tag{5}$$

Noting that the weighted inner product transforms naturally under the projection operators. The local minimum of this Feasibility function is chosen as the design choice for $\mathbb{L}$.

Through similar arguments, the following three Feasibility functions are defined and used to select designs for $\mathbb{W}$, $\mathbb{S}$, and $\mathbb{E}$:

$$\text{Feasibility}_W(\vec{v}) = \left\| \text{proj}_{B_W}(\vec{v}) - \text{proj}_{B_W}(\vec{u}) \right\| \tag{6}$$

$$\text{Feasibility}_S(\vec{v}) = \left\| \left\{ \text{proj}_{B_S}(\vec{v}), \text{mean}\left(\left\| \text{proj}_s(V_W) \right\| \right) \right\} - \left\{ \text{proj}_{B_S}(\vec{u}), \text{mean}\left(\left\| \text{proj}_s\left(\text{proj}_{B_W}(\vec{u}) \right) \right\| \right) \right\} \right\| \tag{7}$$

$$\begin{aligned} \text{Feasibility}_E(\vec{v}) = \Bigg\| & \left\{ \text{proj}_{B_E}(\vec{v}), \text{mean}\left(\left\| \text{proj}_e(V_W) \right\| \right), \text{mean}\left(\left\| \text{proj}_e(V_S) \right\| \right) \right\} \\ & - \left\{ \text{proj}_{B_E}(\vec{u}), \text{mean}\left(\left\| \text{proj}_e\left(\text{proj}_{B_W}(\vec{u}) \right) \right\| \right), \text{mean}\left(\left\| \text{proj}_e\left(\text{proj}_{B_S}(\vec{u}) \right) \right\| \right) \right\} \Bigg\| \end{aligned} \tag{8}$$

It is worth noting that the elements of B_W were determined to be independent of the remaining Performance Measures. Thus, the decision on $\mathbb{W}$ is completely representative of the decision on $\mathbb{T}$.

Appendix B Field of View Design Considerations

There are a few drivers for the choice of the sensor field of view. One is the expected drift of the CubeSat from the boresight axis due to relative motion between the CubeSat and the ISS. Another is desired resolution at long range, if a narrow FOV is chosen, a higher spatial measurement resolution will be possible. Finally, the desired range at first sighting is an important consideration. If a narrow FOV is chosen and the camera is positioned so that the deployer is out of the FOV, the CubeSat may not enter the FOV until it is tens of meters away.

We'll begin with camera lens focal length calculation based on the desired FOV and the camera parameters. Given a desired FOV and size variable related to the sensor size (could be the x, y, or diagonal length of the sensor), the focal length is given by the following

$$f = \frac{d}{tan(FOV/2)} \tag{9}$$

This relation is derived geometrically by using a pinhole camera assumption and assuming that no cropping occurs due to a sensor that is too small. See the diagram below for details.

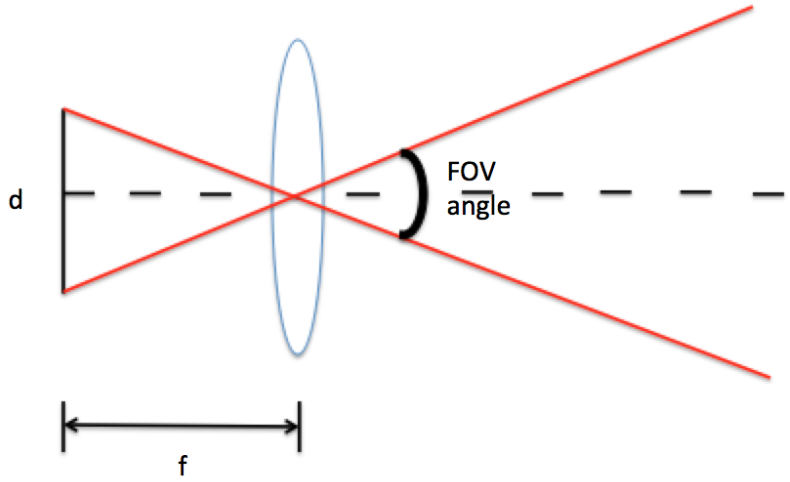

Figure 26. Pinhole approximation for FOV calculations.

Clohessy-Wiltshire:

The motion of a CubeSat relative to the ISS after it's been deployed was estimated to inform the FOV selection. This motion can be modeled by the Clohessy-Wiltshire or Hill equations, taken here from Vallado[27].

ALGORITHM 44: Hill's Equations $(x_o, y_o, z_o, \dot{x}_o, \dot{y}_o, \dot{z}_o$ of the interceptor,

$\omega_{tgt}, \Delta t \Rightarrow x, y, z, \dot{x}, \dot{y}, \dot{z}$ of the interceptor)

$$x(t) = \frac{\dot{x}_o}{\omega}\text{SIN}(\omega t) - \left(3x_o + \frac{2\dot{y}_o}{\omega}\right)\text{COS}(\omega t) + \left(4x_o + \frac{2\dot{y}_o}{\omega}\right)$$

$$y(t) = \left(6x_o + \frac{4\dot{y}_o}{\omega}\right)\text{SIN}(\omega t) + \frac{2\dot{x}_o}{\omega}\text{COS}(\omega t) - (6\omega x_o + 3\dot{y}_o)t + \left(y_o - \frac{2\dot{x}_o}{\omega}\right)$$

$$z(t) = z_o\text{COS}(\omega t) + \frac{\dot{z}_o}{\omega}\text{SIN}(\omega t)$$

$$\dot{x}(t) = \dot{x}_o\text{COS}(\omega t) + (3\omega x_o + 2\dot{y}_o)\text{SIN}(\omega t)$$

$$\dot{y}(t) = (6\omega x_o + 4\dot{y}_o)\text{COS}(\omega t) - 2\dot{x}_o\text{SIN}(\omega t) - (6\omega x_o + 3\dot{y}_o)$$

$$\dot{z}(t) = -z_o\omega\text{SIN}(\omega t) + \dot{z}_o\text{COS}(\omega t)$$

Figure 27. Clohessy-Wiltshire Equations, courtesy of Vallado.

where x,y, and z are the components of the relative position vector in the radial (zenith), velocity, and out-of-plane directions, forming a right-handed coordinate system. ω is the angular rate of the ISS's orbit. These equations assume a near-circular orbit, which is appropriate for the ISS's circular (e=0.0004) orbit. The NanoRacks Clearance Cone Definition Document[28] exactly defines the direction CubeSats are to be deployed from the ISS to be 45 degrees from Nadir in the anti-velocity direction in the plane of the orbit. In the frame defined for the equations above, the corresponding unit vector can be defined as $-(\sqrt{2}/2)\hat{x} - (\sqrt{2}/2)\hat{y}$. This was taken as the initial velocity vector for the CubeSat. Over time, the CubeSat will drift from this initial direction relative to the ISS. This drift is depicted in the graph below. We see that drift is about 3 degrees at 100 meters, indicating that relative motion is not very significant driver for VANTAGE's design because the FOV will be chosen to be much larger than 3 degrees see CubeSats before they get too far down range.

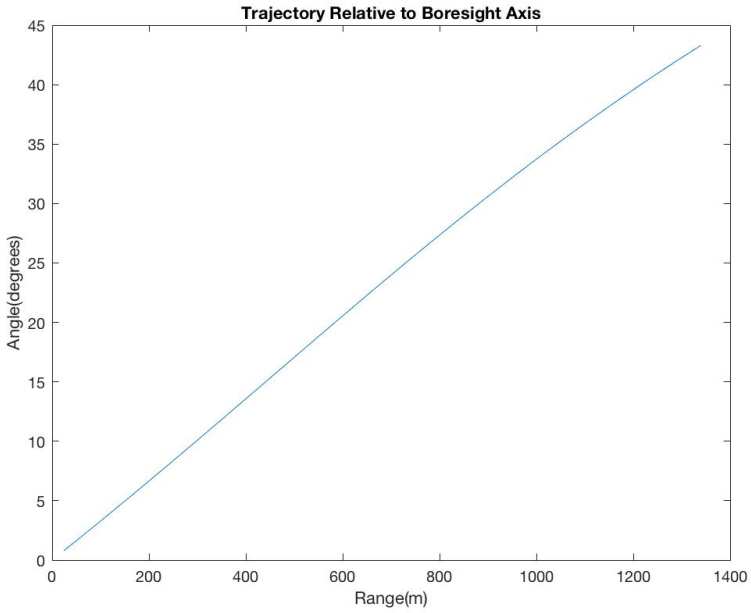

Figure 28. Drift from the boresight axis due to relative motion vs range. As modeled by the CW equations.

Another important consideration when determining the FOV constraints is the distance at which the system will first see the cubesats. This is determined by the FOV as well as the offset distance between VANTAGE and the deployment tube. This relationship is shown in figure 29 below and is characterized by the following function:

$$L = \frac{d}{tan(\text{FOV}/2)} \tag{10}$$

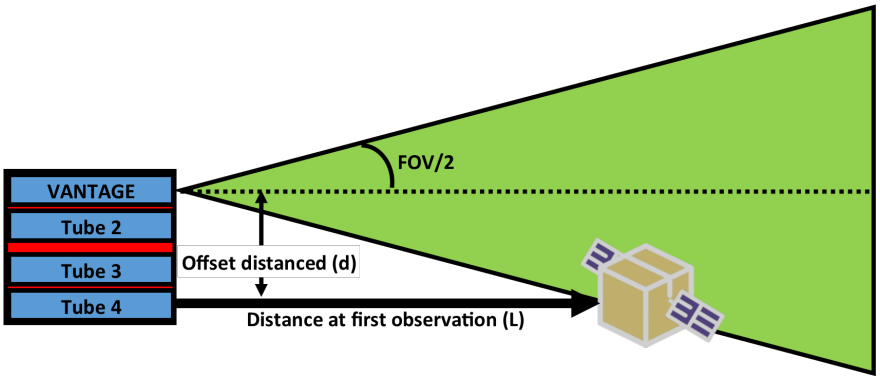

Figure 29. Diagram of first observation geometry.

Plotting this function for for each deployment tube (assuming VANTAGE is at the position of tube 1) yields the plot shown in figure 30 below. This plot can be used to determine the FOV required to observe the most offset cubesat at some customer specified minimum distance.

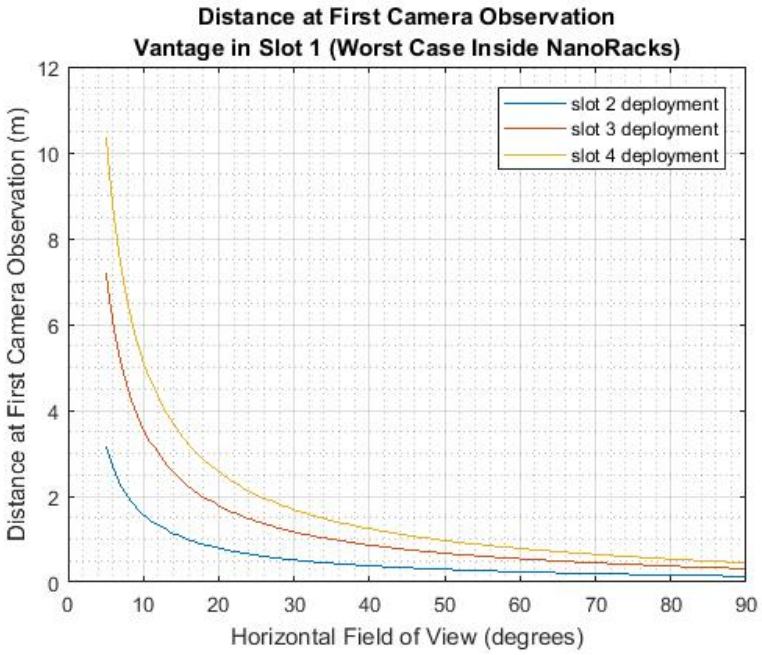

Figure 30. Distance at first observation for maximum offset position (VANTAGE in tube 1).

Appendix C Note on Algorithms Trade

As to why algorithms did not have it's own trade study in the main body of the Conceptual Design Document; among all the interdependent trade study sections, algorithms is the most flexible, given the constraints. Both avionics and structures' designs are bounded by the soft cap customer's given constraints, and the sensors' choices are bounded by a soft cap of finance. An algorithms' design choices are bounded by a soft cap of does at least one research paper exist that demonstrates the practicality or existence of the algorithm we desire. Algorithm choices can also change multiple times given our time constraint, without much impact to the other systems of trade. If two known algorithms perform the same function but with minor variations of side effects, and both are open source, then the impact of choosing one over the other is minor, as there is little cost to switch back at any point of time later. As such it was deemed that the other topics should be traded on, and should have their own section on the trade studies portion of the document, and the impacts on algorithms should be grouped with the structural interface trades, as the structures and algorithms are very intertwined for this project.

Although Structural interface has the most direct impact on the algorithms needed to accomplish the requirements, it is important to note that the algorithms have an implicit impact on all of the design trades. For example, the availability of existing libraries in the programming language section, the software development time in the sensors suite section, and the performance category in the avionics section are all weighted to account for their impact on the algorithm development and implementation.

One could argue that sensors would have just as much impact on structures, if not more, on this project. While structural design options impacts the ability to use many heuristics from the given problem, the sensors would impact what algorithms we use to convert the data into values we could use, and the shape of the values as it progresses through the other sub-component algorithms. We decided to group up algorithms with structures and have it implicit to the rest of the sections due to the impact of the algorithms relative to the overall algorithms used.

While sensors do impact the algorithms needed to take the sensor data and abstract it for the remaining processes, each of those algorithms are tied to the respective sensors, and are not directly interchangeable with each other. Extracting the information from the sensor is only a sub-component overall, as tracking multiple objects, with a good prediction algorithm to maintain the object tracking individually between the every frame (or next time-step where data is collected) will be the same independent of sensor chosen. And at least for the time being, the latter half of the algorithms are treated as the more difficult set.

Structural design choices do not have a direct one to one algorithm tied to each of the choices; although the algorithms will differ among the structures options, this is mainly a derivative effect that location of the sensor will dictate the heuristics available to be used. For example, if the sensor is mounted externally rather than internally, heuristics could be used to only search the entire image resolution for Cubesats in a specific quadrant, and only until the number of designated Cubesats are accounted for. Then the algorithm could use the predictive model algorithm set to minimize the amount of computations used in scanning through the entire field of view for every data set. This also would allow us to not have to use a filter algorithm or a conditional check in the object defining algorithm (like an image classification algorithm for optical sensors) to avoid having stars or any other false positives from being declared a Cubesat, which in turn allows us to bypass having some form of using some form of optimization algorithm to define between which two points is a Cubesat and which is a star, as the internal sensor would not have access to the Cubesats until after they have all deployed, and by the time they no long eclipse each other, they could be mistaken as background noise etc.