University of Colorado Boulder
Department of Aerospace Engineering Sciences
ASEN 4018

Project Definition Document (PDD)

Mobile Astronautic Ranging and Control for Positioning and Location (MARCo PoLo)

Approvals

	Name	Affiliation	Approved	Date
Customer	Taylor Mauer	Lockheed Martin		
Course Coordinator	Kathryn Wingate	CU/AES		

2.1. Project Customer

Name: Taylor Mauer Email: taylor.maurer@lmco.com	Name: Joshua Nelson Email: joshua.d.nelson@lmco.com Phone: N/A

2.2. Team Members

Name: Nadia Abuharus Email: naab7238@colorado.edu	Name: Ben Arnold Email: bear0768@colorado.edu
Name: Brian Byrne (SE) Email: brby2549@colorado.edu	Name: Tyler Candler (CFO) Email: tyca6175@colorado.edu
Name: Austin Coleman Email: auco8437@colorado.edu	Name: Kevin Cook Email: keco2766@colorado.edu
Name: Matt Davis Email: mada9614@colorado.edu	Name: Jackson DePenning Email: jade5047@colorado.edu
Name: Justin Pedersen Email: jupe4558@colorado.edu	Name: Ray Stine Email: rast9345@colorado.edu
Name: Tyler Sterrett Email: tyst6455@colorado.edu	Name: Kintan Surghani (PM) Email: kisu8917@colorado.edu

3. Problem or Need

Expanding government and commercial accessibility into space requires that new technologies and methods for satellite servicing be investigated. On-orbit satellites have a limited time frame to execute their designed mission objectives. Factors such as fuel to sustain orbit, on-board technology limits, and hardware deterioration present as constant obstacles for satellites to remain useful. A design life, which incorporates these factors, is assigned to a spacecraft and establishes the expected duration that the system will remain operational. A satellite's design-life can be relatively short when compared to the price required to place such a system in orbit. Despite recent cost saving advancements in satellite delivery systems, the expenses can still run in the hundreds of millions of dollars. New innovations in satellite servicing and repair will offer solutions to extend the design-life and add more capabilities to on-orbit satellites.

Legacy solutions are prohibitively expensive and cumbersome. The Hubble telescope is a prime example of how small but critical repairs required enormous expenditures, years of astronaut training, and the use of the space shuttle, a notoriously complex vehicle. Regardless of the setbacks, the servicing provided has allowed the Hubble telescope to continue operating for over 30 years and has enhanced capabilities due to the addition of upgraded hardware. The proliferation of CubeSats has offered a new potential solution to providing satellite servicing. With the right compact sensor suite a small platform space vehicle can be capable of identifying and reliably navigating to a target space vehicle to provide services. In addition, new methods of edge-processing can allow greater automation in the system or serve to work in tandem with pre-existing navigation methods.

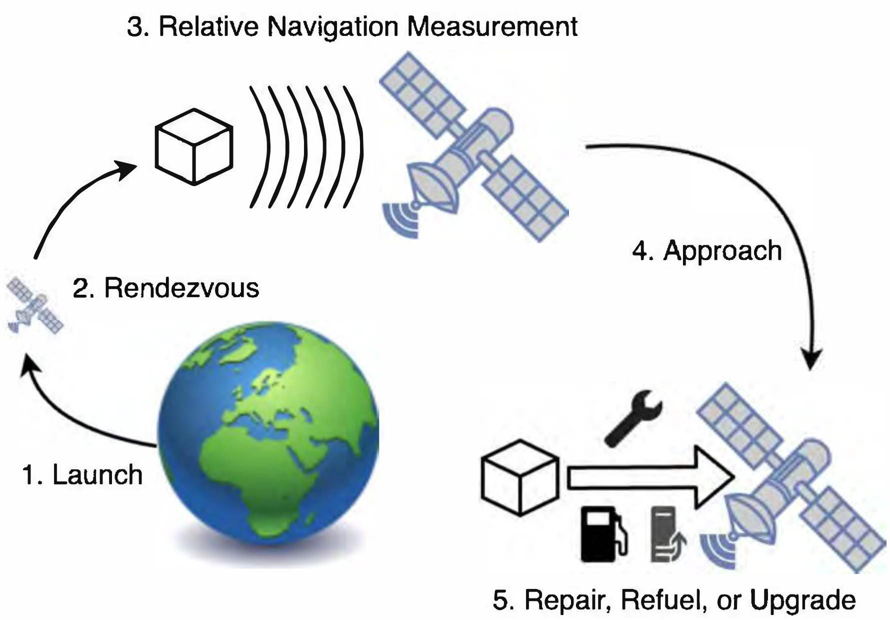

Figure 1: CubeSat Servicing an On-Orbit Satellite

A CubeSat is a cost-effective platform that can be used for servicing and upgrading on-orbit satellites. The small form factor allows for large number of servicing satellites to be deployed in a single launch. With a robust, inter-satellite sensor package the system can guide itself to the target satellite to provide repairs, upgrades, or refueling. Figure 1 is an example of how the CubeSat will carry out a servicing operation. Such a solution will directly increase a satellite design life thus providing a greater return on investment for a mission profile.

4. Previous Work

Autonomous on-orbit satellite servicing has been a longtime goal of the aerospace industry, as it would allow for longer range and less expensive/resource intensive missions than manned satellite servicing. However, the technology to achieve this has only very recently come to fruition, and as such there is significant opportunity for innovation.

The first successful autonomous on-orbit satellite servicing mission was carried out by the Northrop Grumman Mission Extension Vehicle (MEV-1) satellite in April 2020 [1]. The target of the mission was the Intelsat-901 satellite, which was unable to provide its own sufficient propulsion, and would have otherwise been retired in a graveyard orbit. Instead of retiring then, the MEV-1 docked with the craft and returned it to geosynchronous orbit, where it will provide propulsion and attitude control for the Intelsat-901 for 5 years. To achieve this the MEV uses 6 DOF operational control and a Redundant Rendezvous Proximity Operation and Docking (RPOD) sensor suite [2]. Future goals for the MEV craft include refueling, repair, and replacement of parts.

NASA has also begun development of autonomous on-orbit satellite servicing craft. Engineers at the Goddard Space Center have patented a craft (GSC-TOPS-182) that will utilize robotic arms executing pre-programmed instructions or using artificial intelligence to repair satellites [3]. LIDAR is employed for the ranging and optical seeking of a satellite, and docking is achieved through grappling by a robotic arm or berthing pins secured to the satellite structure.

NASA's Exploration and In-Space Services (NEXIS) completed its critical design review for the OSAM-1 spacecraft in March 2022, its mission will have it service an unspecified government spacecraft in low earth orbit [4]. OSAM-1 will utilize various sensors and algorithms, as well as a robotic arm, in order to dock with the client craft. The specific tools utilized for its relative navigation system are called Raven and Argon, Raven is equipped with visible infrared and LIDAR sensors paired with a high speed processor and advanced algorithms [5]. Argon has a vision navigation sensor, long/short range optical cameras, a situational awareness camera, and a SpaceCube flight processor [6]. OSAM-1's key mission is to utilize a unique propellant transfer system to refuel the craft, and afterwards it will relocate the client craft. NASA plans to share this technology with commercial entities to help further develop the autonomous on-orbit satellite servicing industry.

Our project will focus on the relative navigation system of a CubeSat developed by Lockheed Martin to repair satellites, notably distinguished from the above projects by being far smaller and less expensive. The systems used in the above projects for guidance and tracking of the target satellite will inform the development of our project.

5. Specific Objectives

The specific objective of this project is to design, build, and test a prototype of a relative navigation processing system. The core of the system will include a sensor suite to gather the raw data, a processor to translate the raw data into usable information for guidance, and a storage system to store measurements. This system will be tested using Lockheed Martin's Space Operations Simulation Center (SOSC) by autonomously tracking an arbitrary point for the duration of a three hour test. The design will have to mount onto existing hardware and meet objectives provided. Below are the specific objectives broken down by level and category:

	Level 1	Level 2
Measurements	• The onboard sensor suite shall begin measurements of the relative position of the target spacecraft once it is within 30 meters. • Data shall be sampled at a rate of 1 measurement per second.	• The sensor suite shall take measurements against any arbitrary object beyond 30 meters. • Data shall sample at a TBD measurements per second.
Accuracy	• The sensor shall measure 3 degrees of freedom [x, y, z] to an accuracy of 1 meter. • The sensor shall measure 6 degrees of freedom [x, y, z and rotation] to an accuracy of 1 meter and 15 degrees.	• The sensor shall measure 3 degrees of freedom to an accuracy of 0.5 meters. • The sensor shall measure 6 degrees of freedom to an accuracy of 0.5 meters and 5 degrees.
Software	• The onboard processor shall utilize algorithms to process raw data into a GNC readable format • The processor shall store TBD amount of raw data for the duration of 3 hours • The system shall be autonomous except for activation and termination of the sensor suite.	• The processor shall store 1 TB amount of raw data for the duration of 24 hours. • The system shall be completely autonomous.
Construction	• The sensor suite shall be able to mount onto a LINUSS CubeSat (≈ 20cm x 20cm x 30cm) • The sensor suite shall accept a 120 VAC power supply • The sensor suite shall weigh less than 15kg	• The sensor suite shall be able to mount to any CubeSat system.

Table 1: Specific Objectives

6. High Level Functional Requirements

The following functional requirements, represented in table 2, were explicitly written and delivered by Lockheed Martin. Using these requirements and conversations with the customer, the MARCo PoLo team developed a high level and project scope CONOPs shown in figures 2 and 3 respectively.

FR.1	Shall be capable of taking and storing relative measurements within 30m of the test object during testing at SOSC.
FR.2	Shall measure 3DoF (x,y,z) to an accuracy of 1m at a rate of 1 per second.
FR.3	Shall measure relative 6DoF rotation and (x,y,z) to an accuracy of 15 degrees and 1m at a rate of 1 measurement per every 1 second.
FR.4	Shall be capable of measuring and identifying non-fiducial items (ie. can make measurements against an arbitrary test object).
FR.5	Shall be autonomous except for human intervention to start data collection during testing at SOSC.
FR.6	Shall run and store data for the duration of the 3 hour test at SOSC.

Table 2: High Level Functional Requirements

6.1 Analysis

Functional Requirement 1 (FR.1) ensures measurement data necessary for the incoming CubeSat to locate its docking station can be taken and provided to the spacecraft bus for processing.

FR.2 considers three translational degrees of freedom, and ensures processed test measurements are sufficiently accurate for the prototype.

FR.3 is similar to FR.2, but also considers three rotational degrees of freedom, while continuing to ensure processed test measurements are sufficiently accurate for the prototype.

FR.4 ensures the CubeSat can identify the docking station of its target satellite, and utilize relative location to successfully navigate to and dock on board the target satellite.

FR.5 ensures the CubeSat is capable of locating the target satellite and navigating to its docking station without the need for intervention. For the prototype, human intervention will be required to start the imaging.

FR.6 ensures the prototype is capable of maximizing test value by being functional throughout the entire duration of the entire allocated testing window.

6.2 Concept of Operations

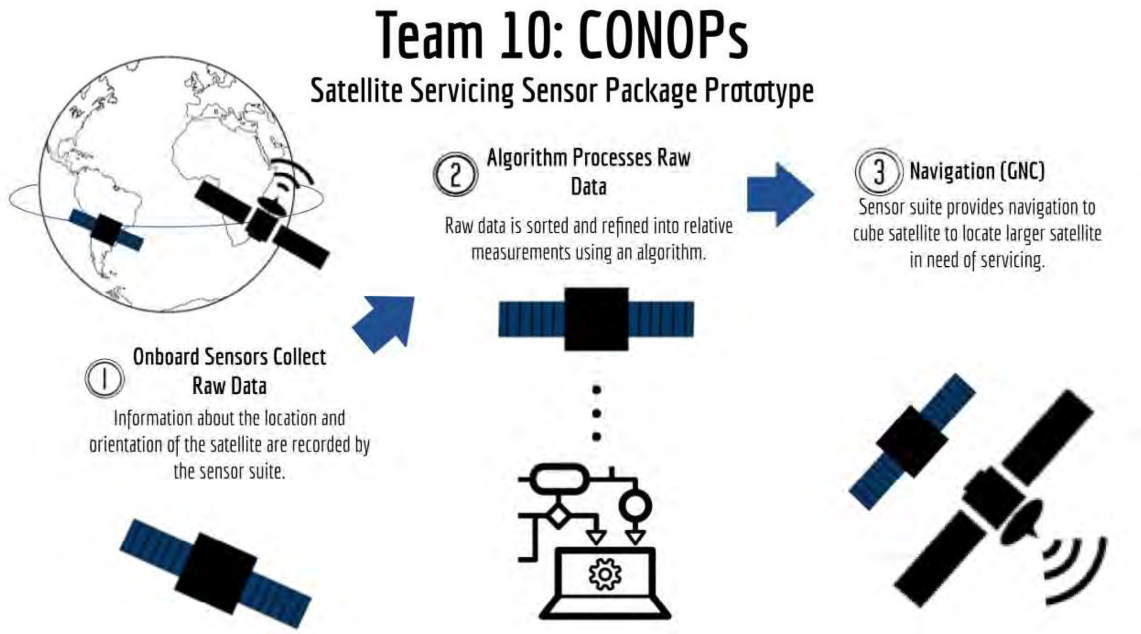

Figure 2: High Level Concept of Operations

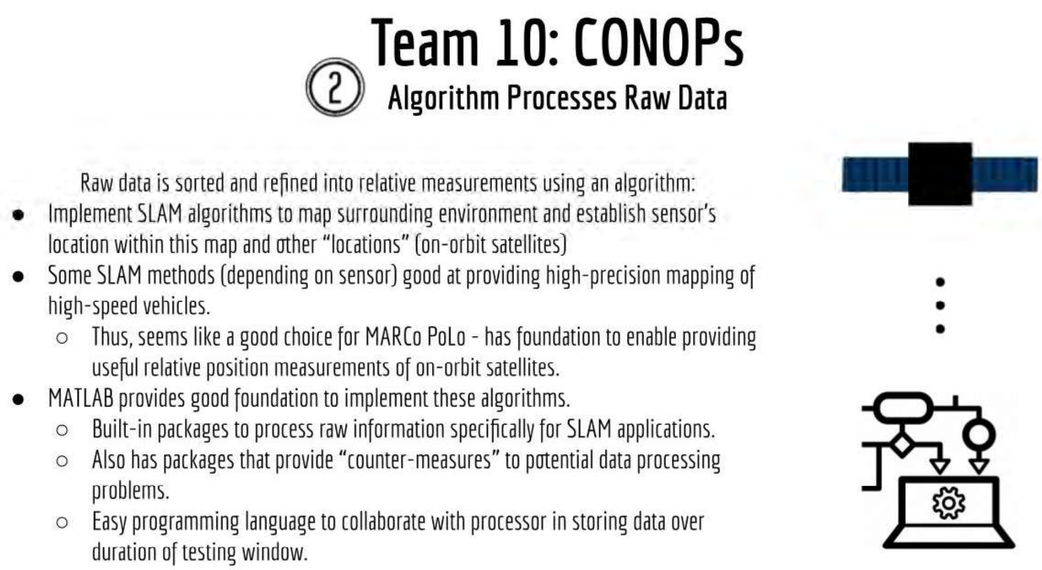

Figure 3: Project Scope Concept of Operations

7. Critical Project Elements

There are eight factors of the project that are critical for success, referred to as Critical Project Elements (CPE). Each CPE is limited by technical, logistical, financial, and/or potential testing/validation constraints. The eights CPEs are tabulated below.

Critical Element		**Constraint Rationale**
CP1	Sensors	Sensors shall measure the CubeSat's relative position with 3 and 6 degrees of freedom. The accuracy of these measurements shall be 1 meter, a rate of 1 meter/second, and 15 degrees. Accuracy in sensor measurements is important for determining navigation.
CP2	Power Source	The sensor suite shall have a power source on the CubeSat. Electronics onboard the sensor suite shall comply with this power source.
CP3	Data Storage	The processing system shall be capable of storing relative measurements made by the sensors throughout the duration of the mission. Data shall also be stored using a to be determined format. Failed data storage means the CubeSat cannot receive the necessary GNC information needed to navigate.
CP4	System Integration	The processing system shall interface with the electronics/sensors properly. This means that the processor is receiving all of the measurements made by the sensor suite in the correct format.
CP5	Data Processing	The processing system shall interpret the sensor data and create a reference frame to measure relative distance between the sensor system test stand.
CP6	Duration	The sensor suite shall function properly for a duration of 3 hours autonomously.
CP7	Mounting	The sensor suite shall be mounted to a test-mount that mimics a CubeSat. This test-mount will be utilized in testing procedures.
CP8	Testing/Validation	Each sensor shall be individually tested to ensure proper functioning and sensitivity. The sensor suite as a whole shall be mounted securely to a testing structure that resembles a CubeSat and tested at Lockheed Martin's SOSC facility.

Table 3: Critical Project Elements

8. Subsystem Breakdown and Interdependence

8.1 Mechanical System

The mechanical system is responsible for designing and building the sensor suite that will house each of the sensors necessary to the navigation system. This system will also be responsible for designing a test-mount that resembles/mimics a CubeSat and will contain the sensor suite. The mechanical system will need to ensure that the test-mount and sensor suite remain structurally sound during the entire 3-hour duration of the test.

The test-mount will be attached to the test-arm provided by Lockheed Martin at the SOSC facility. In order to do this this subsystem will need to reach out to the sponsor for information such as the mounting pattern, weight concerns, and how power will be connected. The suite will be interdependent with the electronics selections. Specifically, the suite will house the sensors, processors, and power supply components. The design of this suite will depend on how power is connected to the test-mount as well as if there are any components that need to be near one another.

8.2 Electrical System

The electrical system has three main purposes and is highly interdependent on the other systems. It will take in 120 volts alternating current and must buck and regulate this to supply the correct direct current to the processor and the sensor system. The design of this dual power supply system is dependant on the voltage and current requirements of the sensor chosen based on the measurement requirements, and the processor chosen based on the requirements set by the software and electrical teams. Lastly, the electrical system must connect the sensor to the processor for sending and storing data. Making sure the sensor data is compatible with the processor and code is interdependent on all systems.

The processor is a major part of the electrical system but is dependent on software requirements, the main requirements being storage and computation. The processor must store three hours of data for the 3DOF and 6DOF measurements being taken at 1 Hz, while still having memory to operate. The computational requirement of the processor is entirely dependant on the algorithm complexity of the software system.

8.3 Software System

The software system's main purpose is to take in raw data from the system's sensors and convert them algorithmically into relative position measurements. This system has a heavy interdependency with the electrical subsystem, as the software has to interface with the sensors, processor, and data storage. The developed software will have to run efficiently enough to generate a usable measurement framerate of 1 Hz without exceeding the maximum clock speed of the processor. The software will then have to store these measurements in the data storage without write errors or exceeding the maximum data storage capacity. There is an additional interdependency with the mechanical system, as the relative positions of the sensors in the system need to be known with a degree of accuracy. The software will need to be able to accept small disturbances in the position of the sensors and any possible misalignments while still converting the raw sensor data into accurate measurements.

A possible avenue to turn this raw data into usable measurements is to implement algorithms that solve the computational problem coined "Simultaneous Localization and Mapping" (SLAM). At a high-level, these algorithms - as the name suggests - uses given measurement information to construct a map of the current, and unknown, environment without any a priori information of the environment. While this mapping is going on, the algorithm is also able to determine the location of the instrument within this mapped environment. Mathematically, the problem is a statistical one at heart governed by a set of probability equations. The measurements used to define/solve the problem are each given a weight of sorts, and statistical techniques such as Kalman filters and particle filters are used to find approximate solutions to the probabilistic equations (map of environment/position within map) that are comprised of these weighted quantities (measurements). In terms of the sensors that provide the information to be processed by the SLAM algorithms, many different ones can be utilized to do so - each with their own level of effectiveness for given applications. LiDAR is a sensor that has been commonly used in SLAM applications and is capable of making very precise measurements of high-speed moving vehicles (on-orbit satellites). Thus, LiDAR SLAM specifically seems like a good starting point to develop the software for this project.

Output measurements from LiDAR typically come in the format of 2D or 3D point cloud data. For this design problem in question, a sensor capable of returning 3D point cloud data is desirable. Movement is then estimated by looking sequentially at the point cloud data, and matching where locations were in each point cloud, and using the displacement to discern how much the sensor has moved in between each "frame". It is worth noting that this point cloud matching process can be expensive in terms of processing power. High-power processors can be necessary depending on the application, and optimization of the processes being used will likely be required to improve the speed of the computations. Clearly, SLAM is not without its setbacks/challenges that may need to be addressed moving forward, some of which include: accumulation of localization errors, loss of position due to localization failing, computational cost, etc. However, in the midst of these potential drawbacks, using SLAM algorithms to process sensor information (to then in turn determine useful relative position measurements) is a very good starting point for MARCo PoLo.

To implement these algorithms, a number of different programming languages could be utilized. For LiDAR SLAM applications (even just general SLAM applications), MATLAB seems to be the best choice at a surface level. Not only does MATLAB come with many built-in packages that aid in the implementation of SLAM applications, but these packages also address many of the drawbacks that can present themselves and offer countermeasures in response. MATLAB has the capability of processing and scan matching 3D LiDAR data via the Lidar Toolbox and Navigation Toolbox, can use output from SLAM algorithms to follow planned paths with obstacle avoidance (with capability of sending real-time commands), and can utilize the Parallel Computing Toolbox to speed of the computational expense by running different branches of code in parallel.

In summary, the software side of this design project must take in raw sensor data and implement algorithms to extract usable relative position measurements. These algorithms will more than likely require their own, unique processor, and the chosen programming language should be compatible with this processor. When being implemented in a satellite mission, this information will then be fed to the broader GNC system to aid in the CubeSat's docking process. At these early stages of development, it seems that LiDAR SLAM algorithms are a good avenue to explore to accomplish the above tasks.

9. Team Skills and Interests

Critical Project Elements	Team member(s) and associated skills/interests
CP1 - Sensors	**Kevin Cook:** Interest and some experience in remote sensing methods and equipment **Ray Stine:** Experience and interest in remote sensing. **Tyler Candler:** Interest in learning about sensors and electronics.
CP2 - Power Source	**Ben Arnold:** Experience working with power regulation systems, interest in designing from the start. **Justin Pedersen:** Interest and experience in signal power transfer via radio frequency
CP3 - Data Storage	**Ben Arnold:** Ensure that processor meets computer systems requirements for storage and computation. **Ray Stine:** Interested in data correction algorithms for data storage.
CP4 - System Integration	**Austin Coleman:** Necessary that software integrates with other subsystems, if not project fails. **Tyler Sterrett:** Interest in gaining exposure to system integration processes. **Jackson DePenning:** Interest in understanding system overlap and interaction . **Tyler Candler:** Interest in gaining exposure to system integration processes. **Matt Davis:** Interest in system integration, experience with electrical and hardware systems.
CP5 - Data Processing	**Austin Coleman:** Interest and experience developing data processing software. **Tyler Sterrett:** Experience and interest in data processing and modeling. **Jackson DePenning:** Experienced in coding data processing software. **Brian Byrne:** Experience and interesting in coding and data processing **Matt Davis:** Interested in data processing element, experience analyzing and processing large amounts of spacecraft data. **Nadia Abuharus:** Interested in developing the processor. Some experience in coding. **Justin Pedersen:** Interest and experience in signal processing and GPS.
CP6 - Duration	**Austin Coleman:** Must ensure software can run duration of testing window. **Ray Stine:** Interested in watchdog timer and error correction algorithm while running tests. **Ben Arnold:** Ensure electronic hardware will stay functional for the whole test.
CP7 - Mounting	**Kintan Surghani:** Interest in manufacturing/testing/prototyping. **Nadia Abuharus:** Experience in design and prototyping (manufacturing and testing). **Tyler Candler:** Interest and experience in design/prototyping. **Kevin Cook:** Both AutoCAD and Solidworks experience that can be used in the manufacturing of the mount
CP8 - Testing/Validation	**Tyler Sterrett:** Experience in design and implementation of software tests. **Kintan Surghani:** Experience with running vibration, electrical, and software testing **Nadia Abuharus:** Experience in simulation (Ansys) and physical testing. **Ray Stine:** Experience in planning and running full system tests. **Brian Byrne:** Experience and interesting in testing and analyzing data from system's test.

Table 4: Team Skills and Interests

10. Resources

Critical Project Elements	Resource/Source
CP1 - Sensors	ASEN Electronics Shop, Adafruit, Newark, Sparkfun (local)
CP2 - Power Source	ASEN Electronics Shop, Adafruit, Newark, Sparkfun (local)
CP3 - Data Storage	Alexandra LeMoine, Josh Mellin
CP4 - System Integration	Robert Marshall, Jade Morton, Scott Palo
CP5 - Data Processing	Alexandra LeMoine, Trudy Schwartz
CP6 - Duration	Alexandra LeMoine, Bobby Hodgkinson
CP7 - Mounting	Matt Rhode, ASEN Machine Shop
CP8 - Testing/Validation	Taylor Mauer, Joshua Nelson, KatieRae Williamson

Table 5: Resources

11. References

[1] Henry, C., 2020. Intelsat-901 satellite, with MEV-1 servicer attached, resumes service - SpaceNews. [online] SpaceNews. Available at: https://spacenews.com/intelsat-901-satellite-with-mev-1-servicer-attached-resumes-service/

[2] eoPortal. 2020. MEV-1 (Mission Extension Vehicle-1) and MEV-2. [online] Available at: https://www.eoportal.org/satellite-missions/mev-1mev-1-mission-extension-vehicle-1-and-mev-2

[3] Technology.nasa.gov. n.d. Method and Associated Apparatus for Capturing, Servicing, and De-Orbiting Earth Satellites Using Robotics. [online] Available at: https://technology.nasa.gov/patent/GSC-TOPS-182

[4] NExIS. n.d. OSAM-1 Mission. [online] Available at: https://nexis.gsfc.nasa.gov/osam-1.html

[5] NExIS. n.d. Raven. [online] Available at: https://nexis.gsfc.nasa.gov/Raven.html

[6] NExIS. n.d. Argon. [online] Available at: https://nexis.gsfc.nasa.gov/argon.html

[7] "What is SLAM (simultaneous localization and mapping) – matlab simulink," What Is SLAM (Simultaneous Localization and Mapping) – MATLAB Simulink - MATLAB Simulink Available: https://www.mathworks.com/discovery/slam.html.

[8] "What is slam algorithm and why slam matters?," GIS Resources Available: https://gisresources.com/what-is-slam-algorithm-and-why-slam-matters/: :text=SLAM