www.adeepakpublishing.com

Uzo-Okoro. E. et al. (2022): JoSS, Vol. 11, No. 2, pp. 1143–1163
(Peer-reviewed article available at www.jossonline.com)

www. JoSSonline.com

Ground-Based 1U CubeSat Robotic Assembly Demonstration

Ezinne Uzo-Okoro, Mary Dahl, and Kerri Cahoy

Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA US

Abstract

In-space assembly of small satellites could enable rapid response and reconfiguration of swarms and constellations. Key gaps currently limiting their in-space assembly are: (1) the lack of standardization of electromechanical CubeSat modules for compatibility with commercial robotic assembly hardware; and (2) testing and modifying commercial robotic assembly hardware to enable successful small satellite assembly on-orbit. Standardization of electromechanical CubeSat modules for on-orbit assembly requires compatibility with low-cost end-effectors. These end-effectors must successfully grasp and manipulate CubeSat parts, engaging component attachment mechanisms, to assemble them into a CubeSat. In this proof-of-concept assembly demonstration, we use modular CubeSat components and two commercial off-the-shelf (COTS) 16 in x 7 in x 7 in (40.6 cm x 17.8 cm x 17.8 cm) dexterous robot arms, weighing two kg each. Working to close both the standardization and testing gaps, we demonstrate the robotic assembly of a 1U CubeSat without humans-in-the-loop in less than eight minutes with a lab prototype. This paper describes both the successful aspects of this work, and lessons learned about issues such as thermal and power for overheated motors and positioning errors. We additionally discuss a plan toward a flight-like system. Robotic assembly of CubeSats in space can increase response time, enable customization, on-orbit replacement and refueling, decrease time and cost spent on Earth-based environmental and vibration testing, and can be more effectively packed for better use of mass and volume given launch costs.

1. Introduction

As space becomes more accessible, there is a need for cost-effective on-orbit assembly and servicing capability to address multiple government (Doggrell, 2006) and commercial constellation needs. We first consider state-of-the- art robotic servicing and assembly capability in general, and then discuss small satellite applications. Examples of state-of-the-art robotic servicing systems include Northrop Grumman's Mission Extension Vehicle-1 (MEV-1), which completed its first docking to a client satellite, Intelsat IS-901, on February 25, 2020. MEV-1 is designed to dock to geostationary satellites whose fuel is nearly depleted, and it does not make use of robot arms (Northrop Grumman, 2020). There are several on-orbit robotic assembly missions that are geared toward use with traditional spacecraft platforms with budgets typically on

Corresponding Author: Ezinne Uzo-Okoro: ezinne@mit.edu
Publication History: Submitted – 02/08/21; Revision Accepted – 02/22/22; Published – 06/12/22

Copyright © A. Deepak Publishing. All rights reserved.

the order of hundreds of millions of dollars. (Flores-Abad et al., 2014). For example, the Defense Advanced Research Projects Agency's (DARPA's) $400M Robotic Servicing of Geosynchronous Satellites (RSGS) program aims to demonstrate that a robotic servicing vehicle can perform safe, reliable, useful, and efficient operations in or near the Geosynchronous Earth Orbit (GEO) environment. RSGS is using the custom-developed Front-end Robotics Enabling Near-term Demonstration (FREND) robot arm, which is a 1.8 m arm from shoulder pitch to wrist pitch weighing 78 kg, with an additional 10 kg for electronics (Kelm et al., 2008).

With large numbers of space assets already in GEO and Low Earth Orbit (LEO), including the International Space Station (ISS) orbit and sun-synchronous orbits, and with the growing number of on-orbit and proposed small satellite constellations and swarms, there is an opportunity for developing space infrastructure that can quickly and cost-effectively deploy, refuel, or modify nodes. An on-orbit service based on robotic assembly of modularized components into small satellites can provide such a cost-effective and rapid-response solution. Advances in satellite cellularization (Kerzhner, 2013; Hill, 2013; Barnhart, 2012; Jaeger and Mirczak, 2013) have made some improvements in the modularization of small satellite subsystems. In this work, however, we develop a new approach to CubeSat production, based on the robotic assembly of functional spacecraft components in space.

A proposed implementation of on-orbit assembly of CubeSat form factor spacecraft is shown in Figure 1 (Uzo-Okoro, 2020). In this example, two commercial off-the-shelf (COTS) robot arms are enclosed in a free-flying small spacecraft "locker" of approximately 24 in x 36 in x 12.5 in (61 cm x 91 cm x 32 cm), sized to fit through the ISS Japanese Experiment Module Exposed Facility (JEM-EF) Airlock (Kawasaki, 2008), inside which CubeSats are assembled and from which they can be deployed. In addition to the two robotic arms for assembly, the spacecraft locker contains modular components such as sensors, power, and propulsion modules. The initial test demonstration spacecraft locker contains components for up to five 1U CubeSats. After an analysis of the test demonstration,

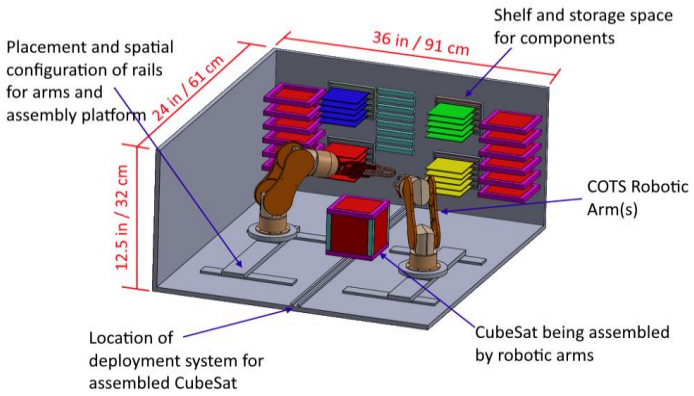

Figure 1. Conceptual system design of the interior of the spacecraft locker.

lockers in the future may contain components for more or for larger CubeSats. These future spacecraft lockers can have propulsion capability or can be compatible with a space-tug in order to deploy on-demand, robot-assembled CubeSats where needed.

The minimum documented launch-on-demand response time is NASA's ISS crew rescue in an emergency at 35 days. (Ceccacci and Dye, 2005). More feasible for CubeSat deployment, however, is using a ride-share program such as Nanoracks. Using Nanoracks, the nominal turnover is three months, and deployment can occur three-to-six months post launch (Nanoracks, 2021). If such a ride-share were not available, the full cost of a rocket would be necessary. With on-orbit assembly, it could instead take on the order of a few hours for a small satellite build and deployment cycle.

This work presents four phases toward on-orbit robotic assembly of CubeSats:

1. Ground Phase. Ground demonstration of robotic assembly of a CubeSat using two dexterous arms and modular CubeSat electromechanical components. In this first phase, there is also design and development of the CubeSat robot-compatible power, propulsion, and sensing modules, and evaluation of their utility for smallsat constellation or swarm architectures.
2. ISS Locker Phase. Development and demonstration on the ISS of a spacecraft locker containing robot arms and CubeSat modular components, possibly including propulsion. The spacecraft

Copyright © A. Deepak Publishing. All rights reserved.

locker could be hosted at the ISS JEM-EF (Kawasaki, 2008; Steimle et al., 2014; Steimle and Pape, 2014), and house enough components to demonstrate the on-orbit assembly of five 1U CubeSats. The first prototype CubeSat would be assembled on earth and deployed to test the structure and deployment system. The four remaining CubeSats—two with Radio Science Experiments (RSE) and magnetometers, and with visible (VIS) sensors—will be robotically assembled on-orbit. The ISS Phase 2 technology demonstration is expected to prove the on-orbit assembly of modular reconfigurable CubeSats, increase Technology Readiness Level (TRL), and assess response time quantitatively.

3. Free-Flying Locker Phase. This phase develops the agile free-flyer spacecraft locker with robotic arms to assemble and deploy rapid-response CubeSats. Response time is further reduced and options to mount on existing satellites are considered. Showing response time that improves ground development time by 10x would be a key objective.
4. Constellation of Lockers Phase. The development of a strategic constellation of free-flyer locker satellites with robotic arms to autonomously assemble and deploy CubeSats in select orbits. The goal is to demonstrate a response that improves ground time by 100x (from a minimum of three months to launch, using Nanoracks).

This paper addresses progress in Phase 1 and outlines plans toward Phases 2 through 4.

1.1. Organization

Section 2 briefly summarizes the state-of-the-art for robot arms in space. Section 3 describes the approach for our lab prototype demonstration of the robotic assembly of a 1U CubeSat by two dexterous COTS robot arms. Section 4 discusses results from using two COTS robot arms to assemble modular CubeSat boards fastened with snaps into a small satellite. The assembly steps use closed-loop control and a Python program. We show that the robotic arm assembly of modular components is a viable option for a CubeSat. In Section 5, we provide a summary of the work, and introduce the next steps for space qualification of the system.

2. Feasibility of Robot Arms in Space

2.1. Flight Heritage

On-orbit robotics missions typically support ISS experiments, exploration, and servicing missions (e.g., to refuel or repair existing satellites) (Katz and Some, 2003; Putz, 1998; Weisbin and Rodriguez, 2000). Previous missions include the DARPA Orbital Express program (Whelan, 2000), the DARPA Phoenix Program (Barnhart, 2013), and the Jet Propulsion Laboratory's (JPL) Mars Insight Mission (Smrekar and Banerdt, 2014). There have been examples of on-orbit robotic assembly, such as the Shuttle Remote Manipulator System (SRMS) (Sallaberger, 1997), also known as Canadarm, which is a 16.9 m, seven degree of freedom (DOF) manipulator with a relocatable base, the National Space Development Agency of Japan's (NASDA) Japanese Experiment Module (JEM) Remote Manipulator System (JEMRMS), which is a 9.91 m, six-DOF manipulator, and lastly, the European Robotic Arm (ERA), which is an 11 m, seven DOF manipulator (Laryssa et al., 2002).

Additional space robotics examples include Hirzinger's multisensory robot, which was tested aboard the Columbia shuttle, operated successfully in autonomous mode, teleoperated by astronauts, and used in telerobotic ground control modes (Hirzinger et al., 1993 and 2002). Some notable autonomy efforts include the SPHERES Universal Docking Port (UDP), which demonstrates autonomous docking maneuvers using small satellites inside the ISS in microgravity (Rodgers, Nolet, and Miller, 2006). Following on from SPHERES, AstroBees is a free-flying robot research platform also used inside the ISS (Bualat et al., 2015).

Table 1 lists examples of space missions using robotic arms for servicing and exploration, and an example of how CubeSat missions can achieve mass and volume savings with modularized COTS components. In this work, we combine COTS, modularity, and robotic assembly. Free-flying robotic servicing missions include the MEV-1, as described in Section 1, and DARPA RSGS programs (Parrish,

Copyright © A. Deepak Publishing. All rights reserved.

Table 1. Capability Gaps in an Example Subset of On-Orbit Servicing Missions

Subset of Robotic and CubeSat Constellation Mission Examples	COTS Robot Arm	Standard Modularized Components	Robotic Assembly / Servicing	Mass / Volume Savings
JPL Mars Insight	✓			
NG MEV-1, DARPA RSGS, GSFC OSAM			✓	
Made In Space Archinaut	✓		✓	
ARC EDSN CubeSats Constellation		✓		✓
This Work	✓	✓	✓	✓

2016). RSGS aims to demonstrate satellite servicing mission operations on GEO satellites. RSGS leverages the FREND project, which developed the state-of-the-art in autonomous rendezvous and docking approaches for satellites not pre-designed for servicing. FREND was the precursor and inspiration for the DARPA RSGS program (Kelm et al., 2008).

NASA Goddard Space Flight Center's (GSFC) On-Orbit Servicing, Assembly, and Manufacturing (OSAM)-1 servicing mission (Reed et al., 2016) is a robotic spacecraft equipped with the tools to rendezvous with, grasp, refuel, and relocate satellites to extend their lifespan. OSAM-1 is intended for a technology demonstration mission to refuel the Landsat 7 satellite in LEO. NASA's Dragonfly robotic servicing project, not to be confused with the Titan mission, has also recently demonstrated a ground-based test of robotic satellite assembly with objectives that include manipulation and placement of large satellite antennas (Piskorz and Jones, 2018; Lymer et al., 2016).

Made In Space's (MIS) Archinaut mission is working with NASA to demonstrate on-orbit assembly using three robot arms to assemble 3-D printed structures in space (Patane, Schomer, and Snyder, 2018). The Archinaut platform is designed to eliminate the need to over-engineer structures to survive launch conditions by constructing them in space. There have been several proposed structures, such as extended booms, sunshields, solar arrays, and large aperture telescopes. This technology will be leveraged by 2022 to construct two beams for the OSAM-2 mission (Harbaugh, 2020).

2.2. The Case for On-orbit SmallSat Assembly

The main concept for this paper is robotic assembly of standardized CubeSat components for mechanical, electrical, power, and thermal subsystems. These form a common bus that can be paired with modular sensors and/or propulsion units. On-orbit assembly allows efficient packing of components (compared with less efficient use of volume for already assembled units). It also enables precise robotic placement and calibration of sensors without risk of being disturbed by the launch process, through the use of custom board housings that utilize snap connectors. The small robotic snap-assembly technique is discussed in Section 3.3.

On-orbit assembly also supports rapid deployment. The custom-built robotic assembly spacecraft locker can be placed in LEO and GEO orbits. For example, if there exists an issue with a LEO asset, and an inspection is required quickly, the spacecraft locker in LEO could robotically assemble a

Copyright © A. Deepak Publishing. All rights reserved.

CubeSat with an RF sensor to listen, a compact lidar or visible or infrared optical capability to image, along with either electric or chemical propulsion, to investigate. The spacecraft locker in LEO, a smart locker with all components ready, would require no wait-time or launch from the ground; it could assemble and deploy the needed CubeSat solution within hours. Similarly, constellations involve a network of many nodes. If a node goes down, it either needs to be replaced by an on-orbit spare, or the constellation needs to be reconfigured. On-orbit spares would need to wait for a launch to the desired location, and may not be configured to investigate or address the anomaly. The spacecraft locker instead could robotically assemble a CubeSat to investigate or replace the node within hours.

On-orbit assembly has performance advantages compared to launching ready-made CubeSats, both for volume usage and satellite modularity. A spacecraft locker filled with densely packed components for on-orbit assembly can be twice as volume efficient as a spacecraft locker filled with ready-made CubeSats (see Figure 2 and Table 2). As the spacecraft locker size grows, we can fit more flat-packed CubeSats components that are robotically assembled versus a static deployer with ready-made CubeSats (Uzo-Okoro, 2020).

For all designs, the robotic arms take up 0.01 m^3, and 0.02 m^3 is used for the assembly area. The mini-fridge-sized locker is 0.07 m^3, which was scaled for this analysis. It is assumed the robot arms can always reach the satellite parts needed, likely through use of a timing belt to transfer parts. The analysis uses

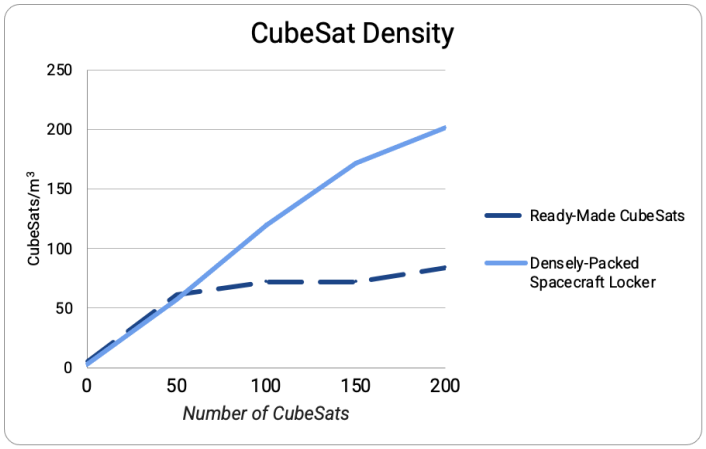

Figure 2. CubeSat density shows a growth in the number of CubeSats.

Table 2. Comparison of Ready-Made CubeSats in Smart Deployer vs. a Flat-Packed Spacecraft Free-Flyer Locker

Capability	Free-Flyer Locker (On-Orbit Robotic Assembly)	Smart Deployer (no Robotic Assembly)
Development Timeline	12-24 months	24-48 months
Launch Timeline	Minimum 35-day launch manifest (One launch)	Minimum 35-day launch manifest (One launch)
Volume (number of Satellites in 177U Free-Flyer)	120 3U CubeSats (shelved/flat packed structures; includes 10U volume for robot arms)	60 3U CubeSats (pre-assembled with structures and rails in upright configuration)
Spacecraft Configurations	Various right-sized power and propulsion modules	Limited (determined before launch)
Propulsion	Various	Limited (determined before launch)
Payload Options	Various purpose-driven sensor types and configurations	Limited (determined before launch)
Deployment to Target	Hours (from on-orbit location)	Hours (from on-orbit location)

Copyright © A. Deepak Publishing. All rights reserved.

conservative flat-pack assumptions and can be further improved with optimized modules. Based on the mini-fridge spacecraft locker, with a mass between 20 kg when empty and 140 kg when loaded with components for 120 1U-sized CubeSats, one can assume a one-time launch cost of between $895K and $1,350K (Spaceflight, 2021). Figure 2 compares the launch costs of a locker in this price range with the trend given by Equation (1), depicted below. Based on these trends, the initial CubeSat launch costs for on-orbit assembly become more favorable than standard terrestrial assembly and launch after a break-even point of 51 CubeSats, as indicated in Figure 2.

An additional consideration between robotic assembly and pre-integrated CubeSats is the mass. The ISS Japanese Experiment Module (JEM) locker's mass constraint is approximately 45 kg for the JEM airlock RMS arm, which serves as an upper bound for the ~177 U of volume. Hand calculations show that the advantage of assembly at the ISS is that the Figures of Merit would be 120 possible CubeSats (1-3U) per unit volume (locker) if components were stacked vertically and assembled robotically versus 72 pre-integrated stored CubeSats, which require a satellite frame and dispenser per CubeSat and cost approximately 10-20% of volume. Additional mass is saved by the locker design by not requiring a dedicated deployer for each individual (or pair) of CubeSats.

Pre-integrated CubeSats have the benefit of lower complexity activities on-orbit (if on ISS, unpacking and maneuvering the deployer rack external to ISS, or a straightforward deployment from a rocket fairing), but lack the flexibility of on-orbit manufacture and assembly; all satellite configurations must be determined prior to launch and must be designed and tested to survive the launch environment. The loss of flexibility, responsivity, repair, and re-use are the trades to compare against the added complexity of robotic assembly on-orbit.

2.3. Cost Considerations

For the purposes of this work, we assume the system is in the ISS orbit, which is between 370 km and 460 km, and has a 51.6-degree inclination. The ISS demonstration concept for the locker could qualify for a research (free-of-charge) ride from the Air Force or NASA (NASA, 2016). Otherwise, we would work through a payload integrator, which bundles several CubeSats into a single launch. We address the flight cost per unit, launch opportunity changes as CubeSat volumes change, and changes to the CubeSat cost based on satellite complexity, in relation to the following four scenarios:

1. Traditional CubeSat assembly (full I&T, ground, no robots) with ground launch or rideshare;
2. Robotic assembly on the ground and then securing a launch;
3. Pre-assembled CubeSats deployed from an on-orbit locker;
4. Robotically assembled CubeSats deployed from an on-orbit locker.

Launch costs are significantly more impactful than the cost of materials and components, although they can vary. For example, low-cost opportunities such as NASA's CubeSat Launch Initiative can cover launch costs for academic applications up to $300K (NASA, 2016). However, launch-only costs can exceed $300K for a 3U CubeSat and higher, depending on the size and weight of the satellite. In 2019, one major integration vendor, Spaceflight Industries (formerly Spaceflight Services) offered bundles beginning with 3U CubeSats for $295K to LEO (Spaceflight, 2021). Another vendor, NanoRacks, quotes $40K per CubeSat unit (for academia) through $85K per CubeSat unit (Nanoracks, 2021).

Although most launch opportunities grow at approximately $80K per CubeSat Unit, the price per unit eventually begins to decrease. Despite this, launch costs of $45K for a 1U CubeSat could be more than 20 times greater than the costs of the hardware components alone as shown by Williams in their approximately $2,200 CubeSat hardware-only cost (Williams, 2016). Using currently available pricing information from Spaceflight, the launch cost for a 1U CubeSat increases linearly (Spaceflight, 2021). The standard launch cost per 1U CubeSat can be represented according to Equation (1):

$$COST_{Std.LV} = (50{,}000 * n) - 5{,}000, \quad (1)$$

Copyright © A. Deepak Publishing. All rights reserved.

where COSTStd.LV is the cost to launch and deploy n-quantity of standard 1U CubeSats in USD. While the cost of large-scale commercial constellations, such as SpaceX's Starlink and the OneWeb constellations, are likely much lower per unit, this project focuses on capabilities for entrepreneurs or research-focused developers in search of low-cost standardized CubeSat options. Without the advantage of also having access to in-house discounted launch services, these developers, often launching a very small number of units at a time, instead manage costs based on the linear scale in Equation (1).

In addition to differences in vehicle costs, flight cost per unit depends heavily on the destination and the configuration of the payload stack. For example, assuming 40% capacity loss to the adapter and CubeSat dispensers, a dedicated rideshare flight on Virgin Orbit LauncherOne to 500 km SSO could be estimated as $110K per CubeSat Unit, while the same vehicle to 230 km equatorial orbit could be $65K per CubeSat Unit (Virgin Orbit, 2019).

3. Robotic Assembly of CubeSats

3.1. Human vs. Robot Assembly Time

To assess the feasibility of robotic CubeSat assembly, we procured low-cost[1] commercial robotic arms and components, as seen in Figure 3. These servo motors were not robust enough to support the concept demonstration due to motor burnout and mechanical inaccuracies. We procured more robust but still inexpensive HiWonder servo motors, used on the LewanSoul xArm robots.

Given that one of our performance metrics is the time it takes for robotic assembly of a CubeSat prototype, we consider example values of both human and robotic assembly in Table 3. Of interest is MakerSat-1, a 1U CubeSat built using an approach with modular components. MakerSat-1 was developed to be constructed by a single astronaut on the ISS by having them plug modular boards together and 3D print rails that snap on without fasteners. MakerSat-1 was ultimately ground assembled and deployed in February 2020, and has been collecting-ionizing-radiation particle counts on-orbit and assessing polymer degradation ever since (Campbell et al., 2020)[2]. MakerSat-1 motivates our use of five minutes as a goal for robotic CubeSat assembly. Configuring our robotic assembly prototype to be similar to the MakerSat-1 assembly, we use two robot arms: one arm to hold the partly assembled satellite, while the other arm inserts and clicks together parts gathered from a shelf.

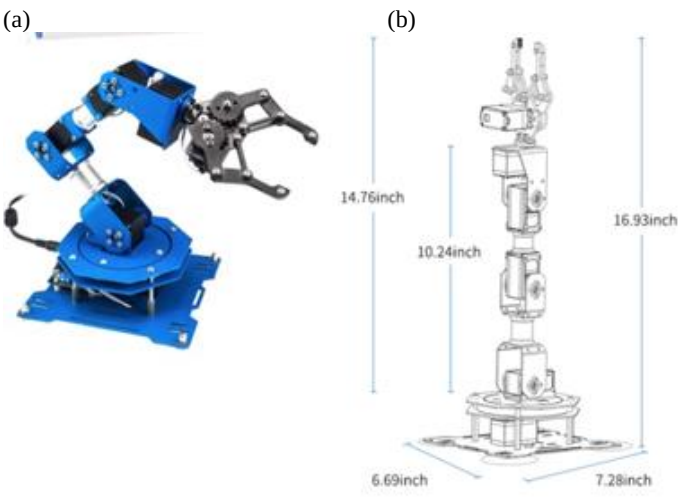

Figure 3. (a) LewanSoul xArm Robot with 6-DOF; (b) Robot arm with dimensions. (Source: LewanSoul xArm User Manual).

Table 3. Example CubeSat Assembly Times (Uzo-Okoro, 2020)

Satellite	Assembly time by human teams
NASA MarCo CubeSat	Several months by a large team (Asmar et al., 2016)
Planet Small Satellite	One spacecraft per day (Soulage et al., 2019)
MakerSat-1	5 minutes in International Space Station by 1 astronaut (Grim, 2016)

[1] Low-cost in this work means less than USD $500 for all robots and CubeSat mock-up components.

[2] A video demonstration of assembly under five minutes is available (NNU, 2017).

Copyright © A. Deepak Publishing. All rights reserved.

3.2. Requirements for Robotic CubeSat Assembly

A subset of some key Level 1 requirements are shown in Appendix A. There are additional requirements, not shown, such as material outgassing. Parts will be examined for resilience in a space-relevant environment by running a thermal vacuum test. Parts which are not suitable for space operation will be modified or replaced as necessary. One Level 1 requirement of note is moving a maximum payload of two kg. In selecting robotic system components, the sensors were evaluated against this requirement.

The total mass of the system is a typical resource limitation for space missions. Working toward the "spacecraft locker" concept, we developed a requirement for the robot kit to weigh less than three kg. This led to lower mass sensors and actuators being preferred. In this work, force-torque (FT) sensors are used at the end-effector (gripper). Many robotic applications require a multi-axis or six-axis (three translations and three rotations) FT sensor to give feedback to the robot about the end-effectors (Li and Chen, 1998). To measure the effort in all six axes, the FT sensor usually combines information from a minimum of six unitary measuring elements such as strain gauges. Using the geometry of the measuring elements, the force and torque are computed along the axes and used in the robot control loop. FT sensors can also be leveraged for sensitive tasks including spiral and linear search, rotational insertion, and path recording (Tsujimura and Yabuta, 1989; Liu et al., 1998). Brushless motors, which are appropriate for space operations, are also selected (Murugesan, 1981).

We considered lower DOF approaches by testing a model simulation of a 2-DOF robot in the PyBullet physics engine. We performed a design optimization for assembly time using the 2-DOF and 6-DOF robots to perform the same task. Simulation results lead to the selection of a 6-DOF robot arm for this work (Uzo-Okoro et al., 2020). Having six or more degrees of freedom meets the topological requirements for grasping components.

The initial laboratory prototyping sensors and servo motors had:

- A six-axis wrist force-torque sensor that measures the wrench (three forces and three torques) at the end-effector.
- Redundant strain gauge bridges for each of the joint torque sensors, on the robot arm in Figure 3, shall measure the output torque of each of the joints of the arm.
- End-effector link strain gauges that measure bending and twist strains for each of the links (the rigid parts of the robots connecting the joints).
- Each servo motor has its own control board and a motor current sensor.

3.3. CubeSat Modularity Approach

Commercially available CubeSat boards and structures are not ideal for on orbit assembly for multiple reasons. These structures are optimized for traditional launch, and thus must survive extreme vibration environments while assembled. This requires lifecycle and environmental testing, even if CubeSats use the exact same design. By flat-packing components and assembling the CubeSat after launch, the structure does not need to withstand these forces. This leads to a design that is both simpler for a robot to construct and requires less mass. Three primary criteria are used in development of modular CubeSats for robotic assembly. (1) The structure has to be compatible with the majority of pre-existing CubeSat deployers and components, adhering to the Cal Poly CubeSat standard (Cal Poly SLO, 2014). While the proposed locker will not be using standard CubeSat deployers, there were several reasons for still adhering closely to the standard. First, it allows the use of commercially available CubeSat boards instead of requiring custom ones, making pre-existing modules easier to introduce to the system. It additionally allows for a slightly-modified version of the structure to be launched in a traditional method as a proof-of-concept demonstration. As the structural design matures, this requirement will be reconsidered. (2) All pieces need to be large enough for the robotic arms to manipulate. (3) Typical fasteners and connectors need to be replaced by robotic-assembly-compatible approaches. Standard mechanical fasteners, such as screws are

Copyright © A. Deepak Publishing. All rights reserved.

both too small and require too fine precision for a low-cost robot to use. The low gravity environment also precludes them, as small detached objects can be hazardous from both a handling and debris perspective. Limitations on the speed and precision of robots may prevent recovery of a fastener if it was improperly placed or released. This would cause time delays in assembly, a waste of resources trying to recover lost fasteners, and having to carry spare fasteners. Additionally, traditional rails for connecting boards were determined infeasible, as they require a high level of precision that the low-cost robot arms cannot supply.

3.4. Mechanical Design Approach

Several iterations of CubeSat component structural designs were needed. The key element in the different structural designs was the method of attachment. Two alternatives to screws were considered, as shown in Figure 4.

The first design, Option 1, uses "latches," or small outcroppings, in the top and bottoms of the rails, shown in purple in Figure 4b. These latches can slide into the base and top of the CubeSat, but cannot be pulled back out without applying pressure to the center of the rail to retract them, allowing the robot to adjust for mistakes. Solar panels would be slid into grooves on the sides of the rails and held in place by the bases, shown in green. Each board would be stacked within the structure. However, after further consideration, his design was ultimately rejected for two reasons: lack of space, and the excessive precision required. The size of the latch required to secure the rails in place was infeasible while allowing adequate surface area for solar panels and adhering to the CubeSat Standard. Even decreasing the latch size to the minimum required latch size was determined still too large for efficient packing. The second reason was the precision needed to place a rail in its position in the base was too great for the low-cost robot arms; there was little room for error when placing rails in the base. Additional structural pieces such as hard stops for guiding the pieces into place will be considered for future work.

The second design, Option 2, uses snaps. A prototype version of this design is seen in Figure 5. For this design, the holes in CubeSat boards typically used for rails are utilized for attaching boards. Bases, such as the one shown in white, have snaps. The robot can press boards into the snaps to attach them. Side solar panels, shown in red, are placed into grooves in the base and held in place with friction. Rails, not pictured, would then snap into knobs on the outside of the structure to hold it together. This design was used for the demonstration robotic assembly, although it encountered problems. There were several instances when the robot arms could not provide enough force to snap parts into place, so after empirical testing, the snaps were shaved down by 35%. Additionally, the side solar panels were often not able to be placed into

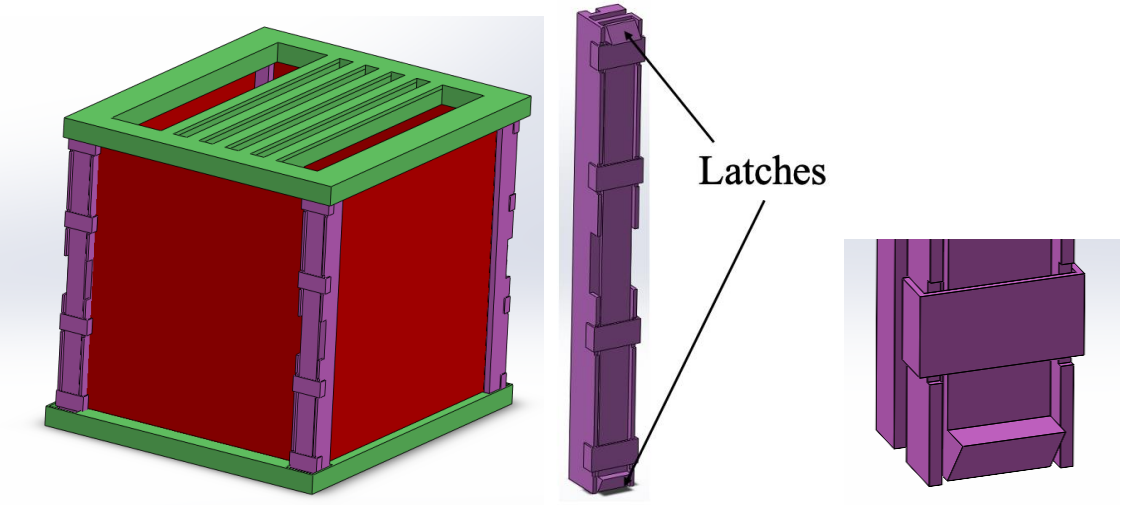

Figure 4. The first structure considered in its entirety (a); with its rails and latches (b); and with detail on its latches (c).

Copyright © A. Deepak Publishing. All rights reserved.

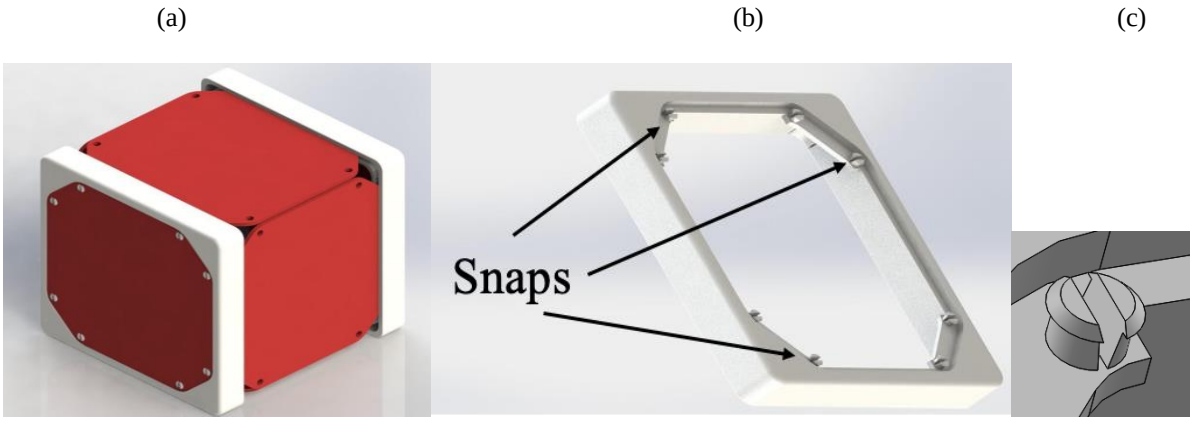

Figure 5. Option 2, (a) assembled without side rails, (b) the top and base with snaps, and (c) detail on the snaps (Uzo-Okoro, 2020).

the grooves with enough precision. Several tests were required to improve precision, as well as adding a camera. Future designs will include hard guides to aid in this.

In addition to the mechanical mating required, the electronic mating of the boards is considered. As one of the primary design constraints is to make the structure compatible with existing CubeSat boards, the electronic mating either needed to use the board-to-board connectors already on the boards, or these connectors needed a simple modification to work. At present, the structures are designed to be compatible with the existing pin to pin connections. An important consideration in assembling with robots is alignment, so it is paramount to design structures that enforce alignment as the structure is assembled. More complex design choices such as modifying boards to have edge card connections or electronic touch-connections were considered, and may be implemented if pin alignment is deemed infeasible for the robot.

To ensure there are no shorts during assembly, no live connections will ever be made during the robotic assembly process. Additional hardware and protocols will be included for the robot to perform a voltage check on each board before the next one is mated. The electrically active hardware, such as the battery, will have inhibits that can be pulled after assembly. Electronic connections were not included in the demonstration discussed in this paper, but will be carefully tested and monitored in future work.

The parts for design iteration were made with Acrylonitrile Butadiene Styrene (ABS) using a Fused Filament Fabrication (FFF) 3D printer, as it is effective for laboratory prototyping purposes. We anticipate the final design will be 3D-printed using a low outgassing material such as Ultem 9085 (Chuang et al., 2015) using FFF, or Windform XT 2.0 (CRP Technology, 2015) using a Selective Laser Sintering (SLS) printer. 3D-printing provides us with multiple advantages over machining. 3D-printing is both faster and cheaper than traditionally machined parts. 3D-printing also supports fine detail and features that would be challenging or costly to have machined, such as the snaps shown in Figure 5(a).

3.5. Robotic Assembly Approach

The block diagram in Figure 6 shows the laboratory prototype setup. The diagram shows key elements of the setup and the data connections from each servo motor to the controller board. All servo motors serve the same function, which is to power the robot arm, and possess the same characteristics (Uzo-Okoro, 2020).

The FT Sensor receives software commands from a MacBook Pro 2015 laptop, which are passed to the Servo Controller board. The Raspberry Pi 3 (RPi3) camera provides the pose of the 1U CubeSat components to the software to steer the capture trajectory and to determine when the component is within the robot arm's capture envelope. The Servo Controller sends a command to the servo motors using

Copyright © A. Deepak Publishing. All rights reserved.

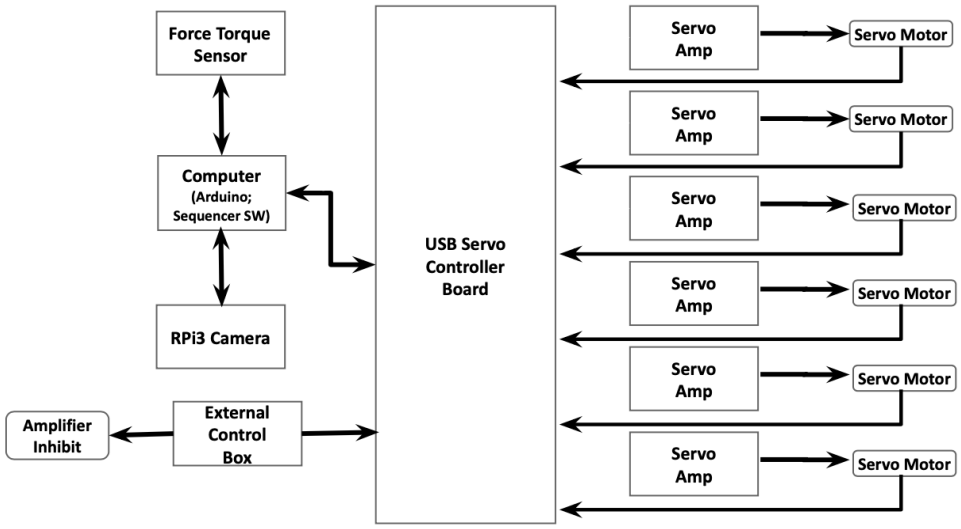

Figure 6. Robot assembly block diagram (Uzo-Okoro, 2020).

amplifiers. The servo motors execute the command on the arm joint (shoulder, elbow or wrist) or sensor head rotation. The encoded action is sent back to the Servo Controller board, through the serial port, and to the computer. The computer interprets the action and sends additional commands to the FT Sensor. While brushless direct current (DC) motors will be used for the space-qualified test, the LewanSoul pre-packaged kit came with servo motors which are used for the initial laboratory prototype.

3.6. Motion Planning and Control

3.6.1. Overview

Motion planning is a key component for robotic autonomy. It accomplishes two goals: ensuring collision-free trajectories and allowing the robot to reach the goal location as fast as possible. The space of all possible configurations is assessed to determine a two- or three-dimensional configuration space. Control operations can be either closed-loop or open-loop, depending on the desire for feedback. The challenge for this project is to ensure the assembly is rapid enough to meet on-demand needs, but limits the errors in assembly, as broken parts are difficult to diagnose and dispose of on orbit.

3.6.2. Control Selection

In this work, to ensure the robot arms reach the target boards with a single calculation, open-loop control, which is faster than closed-loop control, is initially used. An open-loop control system performs based on the input, and the output has no effect on the control action. With open-loop control, outputs rarely change and process disturbances are not the norm. In a controlled environment, such as the laboratory or potentially the spacecraft locker, it is possible to use open-loop control, as there are few disturbances to affect the unknowns in the system. Closed-loop control is best used when the environment has unknowns and measurements are required. By utilizing the predicted responses to input control, the process is set on certain points within a given accuracy and automates correction to process disturbances.

After experiencing the wear and tear of components, and part reliability issues, we selected a closed-loop control as the better choice given its ability to self-correct. Figure 7 shows a block diagram of the final closed-feedback, control-loop robot software system, which consists of a robot system and a control unit. The robot system represents the mechanical components, and the control unit represents the electronics components. The desired position and other control parameters are supplied by the user to the control unit. The control parameters

Copyright © A. Deepak Publishing. All rights reserved.

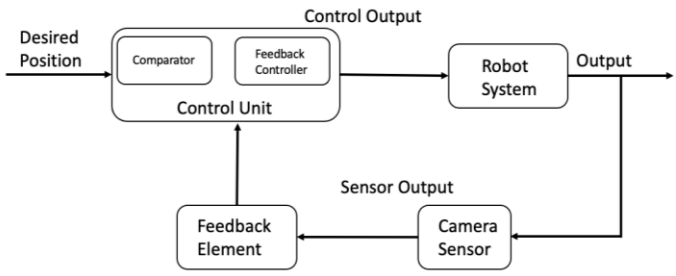

Figure 7. Closed-feedback control diagram.

describe the characteristics of the robot unit. The control unit collects data and calculates the needed force. The interface between the robot unit and the control unit includes comparators and feedback controllers, which result in control output parameters. The controllers track the current position of the mechanical component and transform the current position based on sensor outputs. The sensor output is converted in the feedback element, and sent to the control unit. The calculated output value is converted to an analog signal, and that signal drives the motor to move the robot.

3.6.3. Task Planning Software

We implement all software of the robot arm assembly system in Python 3.7.7, using an Inverse Kinematics (IK) library and a test function for the robot arm. The task plan consists of two precise subtasks: grasping an object and moving it to the drop-off location, where satellite assembly takes place. The robot arm end effector has to be located within two mm of the pickup and drop-off locations for at least five seconds to perform the required pickup and assembly operations. The task planning module is hard-coded. A power consumption module satisfies the need for control of the servo motor power consumption. Servo motors are specified to have a no-load current and a maximum load. The software used to control robotic arms can typically be customized for control system parameters. The interaction between the control parameters and the physical dexterity can be complex due to communication latencies and multi-tasking using the operating system.

For task planning in two-dimensional workspaces, we determine the viewpoints for the entire target surface using a randomized sampling. This randomized sampling, as defined in the Traveling Salesman Problem (TSP) (Applegate, 2007), optimizes task planning by reducing the computational effort and memory load associated with determining and arriving at the target viewpoints. The TSP enables three tasks. The tasks are the selection of components, assembly of components in required time (<50 seconds), and connecting the components. Test runs show that task planning between specific goals dominates the runtime cost compared to the computation of approximate solutions to the TSP. A lower bound estimation of the task length between goals is used to calculate candidate TSP solutions. The complete task planner is used for edges in candidate solutions.

The robotic arm control algorithm (Paus et al., 2017) finds a solution to two key problems: path planning and robot arm placement. This is accomplished by using a divide and conquer strategy and optimization heuristic planning approaches to the reachability and the coverage problem. The algorithm shows how we sample points from the target surface and use the points TStarget to estimate the progress of the coverage planning. We store all the points in the solutions in the set Tcoverage, allT and align each pose (from the global set) pA, with reachable target points from the predefined map. The main loop continues until R, a function used to determine reach, is empty or all target points are covered. Next, we find the pose pmax, which includes the largest subset of the rest of the target points. We also use the coverage planner to find a trajectory t in as many points as possible in R(pmax), which are stored in Tcoverage.

Constraints like stability requirements are taken into account by the coverage planner. Tcoverage and pmax are removed from R by updating every entry(p',T')R to remove the covered points (p', T'\Tcoverage). Entries with no reachable points are emptied during each timestep, which is 10 ms. Given the multi-step process required, we use the Python time() function, to measure time and create a function to configure the clock and evaluate the microcontroller at 100 Hz. And for the last steps in the loop, we update the target points Tcoverage, all by adding the points in Tcoverage and adding (pmax, t) to the list of solutions.

Copyright © A. Deepak Publishing. All rights reserved.

Upon completion of the while loop, we find the degree of coverage.

3.7. Prototype Laboratory Implementation

LewanSoul robots were used, as shown in Figure 3. An RPi camera is set atop a 2.5 ft tall post, with an Arduino attached behind it. The Arduino served as the controller to the RPi camera, which took images. As shown in Figure 8, red prototype CubeSat boards are set up in front of the two robot arms. The process begins with the RPi camera capturing an image of the platform. OpenCV object detection software libraries are used in a Python software program to identify the color-coded boards, calculating the center of the boards for grasp accuracy (by converting pixels to meters). After an image capture of the field, the pixel position of the boards' center is calculated, resulting in two sets of (x, y) points in meters and pixels. The maximum range for the LewanSoul robot arms is +0.15 m to -0.15 m in the y-direction and 0 m to 0.3 m in the x-direction, which determines the initial placement of each arm and board stacks.

Using Inverse Kinematics (Rodgers, Nolet, and Miller, 2006), the board location values are converted into a set of six angles. Since there are six servos on each arm, the Raspberry Pi 4 (RPi4) sends six values for each target location, one angle for each servo, to the Arduino for control of the arm via the USB serial port. The serial port is used for the RPi4 communication with the Arduino. It controls the six different servos on each robot arm by first restricting the rotating limits for each servo. The rotation range is between 0 and 240 degrees and the minimum increment for each is 13.8 degrees. Using each servo's unique ID number, their rotating duration and rotating position are controlled. In the Arduino code, we pre-defined several functions that move the arm to the vertical initial position, move it to the target location based on input arguments, and move the arm to the "parts bin" location, namely move_to_initial(), move_to() and move_to_bin().

The Arduino has only one serial port, and needs to communicate with both the RPi4 and the six servos, so an additional hardware serial port was set up. A protocol requiring the RPi4 and Arduino to communicate and confirm messages was added to ensure all six values were sent. This acknowledgment approach is useful for error detection. Once complete, the camera would capture a new image of the boards

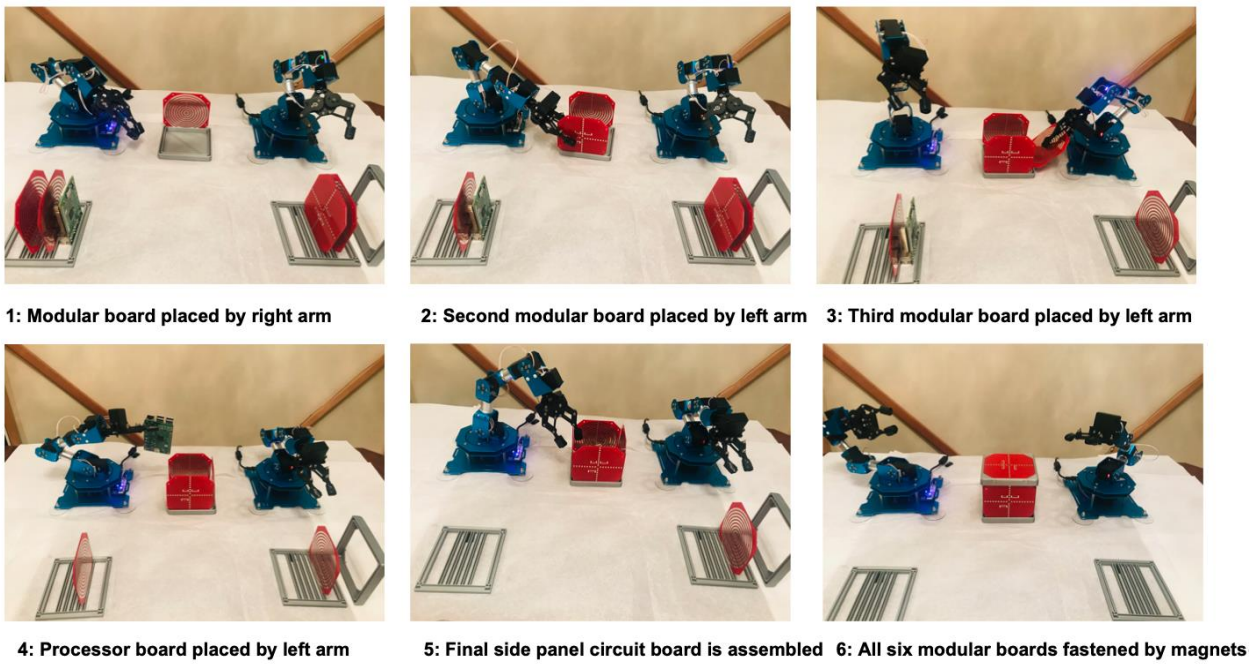

Figure 8. Sequence of robot arm assembly of a 1U CubeSat in under eight minutes (Uzo-Okoro, 2020).

Copyright © A. Deepak Publishing. All rights reserved.

to be processed. The arm proceeds to perform movements to grasp each board at target locations and begin assembly using specified location values. The process is repeated until a CubeSat has been assembled. The steps of the assembly process are captured in Figure 8.

4. Results

4.1. Observations

The LewanSoul arms assembled the structures and six prototype boards in under eight minutes. The robot arms were subjected to 170 hours of tests; all servo motors and rotation angles were tested to determine stability, accuracy, and feasibility of operation. The accuracy plots are shown in Figure 9. We automated repetitive tests of each servo motor for over 120 hours, while the software for the satellite assembly was being programmed.

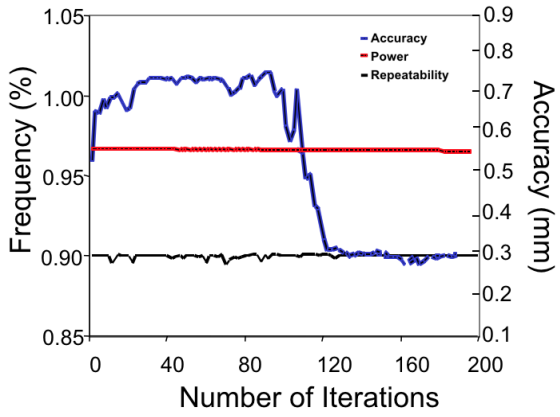

Figure 9. Robot arm power consumption and accuracy.

There were initial challenges in assembly as the robot arms kept missing the assembly area, the structure spaces for board placement, and the correct angle for side boards. While the boards were grasped within the first week of programming, we learned after five weeks of errors in placement that slowing the speed of the arm movement by a factor of two as the robot arm approached the satellite assembly area addressed the problem. For instance, if the board was picked up by the robot arm, the board traveled the 0.2 m distance to the assembly area in 1,480 ms. We also moved each servo motor (robot arm joint) in 1480 ms—across the same distance—as the board approached its final destination. When the board arrived at the assembly area, the board was lowered carefully into its intended position in 740 ms. Despite this slowdown, the robotic assembly of each component took approximately 22.25 seconds. It took the same amount of time to grasp the other structures as it did to grasp boards.

Additional issues arose during the assembly process, such as loose grippers. The grippers became loose after over 100 hours of use and were not able to pick up the boards, as they were sliding off the gripper pads. The grippers were subsequently tightened. On occasion, electrical tape was used on the gripper pads to retrain the gripper in a gripping position.

The camera lighting is controlled by a Python script. We imported the picamera package, which serves as a Python interface to the camera module for Python 3.7.7. Pi Cam lighting control was used to ensure adequate lighting at all times. Sometimes the LED for lighting did not work. To resolve this, we needed to tighten the bolts of the LED and camera assembly and then restart the camera.

Ultimately, all challenges were resolved, and the entire 1U CubeSat was assembled 57 times with no humans-in-the-loop, ranging from 19 minutes and five seconds to seven minutes and 39 seconds with optimization. The second best assembly time was eight minutes and 40 seconds.

Using the Inverse Kinematics approach made for less intensive programming; however, using a robot arm as part of a larger system required mastering a learning curve in robot automation and robotics programming. The Python code resulted in several hundreds of lines of code, which was human-intensive to create. As shown in Figure 9, while robot arm power consumption and repeatability of movements are predictable, there was a decline in the robot arm's 95% accuracy requirement after 120 iterations. However, at this time, this cost-effective prototype was unable to repair each other or the setup. We observed the robots become physically shaky and technically imprecise. For instance, although the robot arm was programmed with the correct coordinates, it kept missing the structure by 2 cm when installing a board because the

Copyright © A. Deepak Publishing. All rights reserved.

servos were worn out from excessive use. We added some additional error detection to the code and adjusted the software to offset the coordinates to match the functional offset over time. All programming and CAD work was completed on an Apple MacBook Pro 2015 laptop with access to and use of standard Python 3.7.7 libraries. The standard libraries used are pybullet, which includes calculateinversekinematics(), pybullet_data, math, time, datetime, and numpy.

5. Summary and Future Work

5.1. Summary

We have shown that even low-cost commercial robot arms can be used to assemble a 1U lab prototype CubeSat in under eight minutes. We note that improved motors would be necessary for an ISS demonstration, as servo motors burn out due to degradation after fewer than 200 hours of testing and use. We additionally found that the end-effector (gripper) accuracy diminishes with time. In addition to considering more precise and rugged robot end effectors, development of a self-calibration routine will be necessary for on-orbit use. For space environment use, we are carefully assessing more rugged and moderate cost robotic arm systems, as well as their materials.

While we successfully developed structural 1U CubeSat form factors for robotic assembly, we anticipate expanding this research prototype from 1U CubeSats to larger small satellites of varying forms factors and designs to meet current industry needs. Another next step for future work is incorporating and testing post-assembly function. Power and data connectors that can be successfully robotically mated need to be developed and tested. We are investigating the use of optical data transfer and magnetic power transfer.

Robotic assembly on-orbit could reduce schedule and integration and test costs. The benchmarked small satellite assembly time with a human-in-the-loop requires 12 to 90 months of component assembly and integration time on Earth. We anticipate that on-orbit assembly capability optimized for a 1U functional CubeSat with 30 W of total power, would reduce the assembly time by an order of magnitude. Using robotic arm simulation models, for a 1 U CubeSat assembly, we showed that sub-minute assembly of a flight component is possible with only a few iterations, and that with more iterations, the approximate assembly time result is achieved (Uzo-Okoro, 2020).

As small satellites and constellation missions continue to evolve, demand for precise and rapid CubeSat assembly with no humans-in-the-loop will grow. Benefits of robotic assembly include precision alignment and calibration of sensors on-orbit, without having to survive the launch environment. Electrical and mechanical design of components that are robotically assembled in space can also be lighter and less complex than those assembled on the ground and subjected to the launch environment.

5.2. Future Work

We are performing trade studies on moderate-cost robotic assembly systems that can address the reliability concerns with the low-cost systems, and that are compatible with environmental testing (thermal vacuum, and vibration). We are developing more refined power budgets and thermal management plans for the robotic assembly system.

We are developing more flight-like mechanical structures and electrically functional CubeSat boards and modules with data and power connectors that are able to be robotically mated. Development of a new CubeSat standard for robotic assembly would be beneficial to encouraging vendors to produce low-cost, robotic-assembly compatible components. Electronic connections will be carefully tested and monitored in future work.

We are exploring other robot options such as cartesian robots, and continuing the use of closed-loop control systems to compensate for reliability issues. We consider additional path planning and localization algorithms as well to add to the robustness of the system.

We are planning for lifetime and build precision and quality testing, for a mission lifetime of two years, as well as periodic self-calibration testing and self-maintenance.

Copyright © A. Deepak Publishing. All rights reserved.

We are also developing more detailed designs for the spacecraft locker, and its subsystems, including thermal management and attitude determination and control (particularly with the active robots moving mass around inside the locker and then deploying CubeSats). The spacecraft locker also needs to be developed to try to reduce effects of the launch environment on the packed and stored CubeSat modular components.

Acknowledgements

We thank Christian Haughwout and Emily Kiley for valuable insights and support throughout this work.

References

Applegate, David L., Bixby, Robert E., Chvátal, Vašek, and Cook, William J. (2007): The Traveling Salesman Problem: A Computational Study, Princeton, NJ: Princeton University Press. doi: 10.1515/9781400841103.

Barnhart, D., et al. (2012): Changing Satellite Morphology through Cellularization, in *Proc. AIAA SPACE 2012 Conf. and Exposition*, Pasadena, CA. Sept. 11–13. Paper 2012-5262. doi: 10.2514/6.2012-5262.

Barnhart, D., et al. (2013): Phoenix Program Status-2013, in *Proc. AIAA SPACE 2013 Conf. and Exposition*, San Diego, CA. Sept 10–12. Paper 2013-5341. doi: 10.2514/6.2013-534.

Bualat, M., et al. (2015): Astrobee: Developing A Free-Flying Robot for the International Space Station, presented at *AIAA SPACE 2015 Conf. and Exposition*, Pasadena, CA. Aug. 31–Sept. 2. Paper 2015-4643. doi: 10.2514/6.2015-4643.

Cal PolySLO (2014): CubeSat Design Specification, Rev. 14.1. Available at: https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/62193b7fc9e72e0053f00910/16458208097 79/CDS+REV14_1+2022-02-09.pdf (accessed Feb. 22, 2022).

Campbell B. et al. (2020): On-Orbit Polymer Degradation Results from MakerSat-1: First Satellite Designed to be Additively Manufactured in Space, in *Proc. Conf. on Small Satellites*, Aug. 1–6. Online Event, Paper SSC20–WKVII-04. Available at: https://digitalcommons.usu.edu/smallsat/2020/all2020/45/ (accessed Apr. 28, 2022).

Ceccacci, A. and Dye, P. (2005): Contingency Shuttle Crew Support (CSCS)/Rescue Flight Resource Book, National Aeronautics and Space Administration, pp. 89. Available at: https://www.nasa.gov/pdf/153444main_CSCS_Resource_%20Book.pdf (accessed Apr. 28, 2022).

Chuang, K, et al. (2015): Additive Manufacturing and Characterization of Ultem Polymers and Composites, in *Proc., Composites and Advanced Materials Expo (CAMX) Conf.*, Dallas, TX, October 26–29. Available at: https://ntrs.nasa./api/citations/20160001352/downloads/20160001352.pdf (accessed Apr. 25, 2022).

CRP Technology (2015): Windform Materials for Advanced 3D Printing. Available at: http://www.windform.com/news/new-achievements-windform-materials.html (accessed Jan. 25, 2021).

Doggrell, L. (2006): Operationally Responsive Space: A Vision for the Future of Military Space. Air Univ Maxwell AFB, AL Airpower J., Vol. l., pp. 42–49. Available at: https://apps.dtic.mil/sti/pdfs/ADP023958.pdf (accessed Apr. 25, 2022).

Fairchild, Carol, and Thomas L. Harman (2016): ROS Robotics by Example: Bring Life to Your Robot Using ROS Robotic Applications. Houston: Packt Publishing Ltd. doi: http://hdl.handle.net/10657.1/888.

Flores-Abad, A., et al. (2014): A Review of Space Robotics Technologies for On-orbit Servicing. Progress in Aerospace Sciences, Vol. 68, pp. 1–26. doi: 10.1016/j.paerosei.201403.002.

Grim, B., et al. (2016): MakerSat: A CubeSat Designed for In-Space Assembly, in *Proc., Conf. on Small Satellites*, Logan, UT, Aug. 6–11. Paper SSC16–WK-29. Available at: https://pdfs.semanticscholar.org/1d7a/9a0fe71ef4fd6559ec93624de20b0df6986f.pdf (accessed Apr. 28, 2022).

Harbaugh, Jennifer (2020): OSAM-2 | NASA. Available at: https://www.nasa.gov/mission_pages/tdm/osam-2.html (accessed Jan. 27, 2021).

Copyright © A. Deepak Publishing. All rights reserved.

Hill, L., et al. (2014): The Market for Satellite Cellularization: A Historical View of the Impact of the Satlet Morphology on the Space Industry, in *Proc., AIAA SPACE 2013 Conf. and Exposition*, San Diego, CA. Sept. 10–12. Paper 2013–5486. doi: 10.2514/6.2013-5486.

Hirzinger, G., et al. (2002): Robotics and Mechatronics in Aerospace, in Proc., 7th Int. Workshop on Advanced Motion Control. Maribor, Slovenia. pp. 19–27. doi: 10.1109/AMC.2002.1026885.

Hirzinger, G., et al. (1993): Sensor-based Space Robotics-ROTEX and its Telerobotic Features. IEEE Transactions on Robotics and Automation, Vol. 9, No. 5, pp. 649–663. doi: 10.101109/70.258056.

Jaeger, T., and Mirczak W. (2013): Satlets - The Building Blocks of Future Satellites - and Which Mold do You Use?, in *Proc., AIAA SPACE 2013 Conf. and Exposition*, San Diego, CA, Sept.10–12, 2013. Paper 2013–5485. doi: 10.3514/6.2013-5485.

Katz, D. S. and Some, R. R. (2003): NASA Advances Robotic Space Exploration. Computer, Vol. 36, No. 1, pp. 52–61. doi: 10.1109/MC.2003.1160056.

Kawasaki, K. (2008): Overview of JEM-EF on ISS, in *Proc. of the RIKEN Symposium*, Saitama, Japan, pp. 4–7. doi: 10.1.1.1056.9284.

Kelm, B. E. et al. (2008): FREND: Pushing the Envelope of Space Robotics. Washington, DC: NRL Review—Space Res., pp. 239-241. Available at: https://apps.dtic.mil/sti/pdfs/ADA517473.pdf (accessed on Jun 4, 2020).

Kerzhner, A., et al. (2013): Architecting Cellularized Space Systems using Model-based Design Exploration, in *Proc., AIAA SPACE 2013 Conf. and Exposition*, San Diego, CA. Sept. 10–12. Paper 2013–5371. doi: 10.3514/6.2013-5371.

Laryssa, P., et al. (2002): International Space Station Robotics: A Comparative Study of ERA, JEMRMS and MSS, in *Proc., 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation, ESTEC*, Noordwijk, The Netherlands, Nov. 19–21. Available at: http://robotics.estec.esa.int/ASTA.Astra2002/Papers/astra2002_1.3-1.pdf. (accessed April 25, 2022).

Li, Y. F., and Chen, X. B. (1998): On the Dynamic Behavior of a Force/torque Sensor for Robots. IEEE Transactions on Instrumentation and Measurement, Vol. 47, No.1, pp. 304–308. doi: 10.1109/19.728839.

Liu, G. et al. (1998): A Base Force/torque Sensor Approach to Robot Manipulator Inertial Parameter Estimation, in *Proc., 1998 IEEE Int. Conf. on Robotics and Automation*. Leuven, Belgium, Vol. 4, pp. 3316–3321. doi: 10.1109/ROBOT.1998.680950.

Lymer, J. et al. (2016): Commercial Application of Inspace Assembly, in *Proc., AIAA SPACE 2016 Conf. and Exposition*, Long Beach, CA, Sept. 13–16, Paper 2016–5236. doi: 10.2514/6.2016-5236.

Murugesan, S. (1981): An Overview of Electric Motors for Space Applications. IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol. IECI–28, No. 4, pp. 260–265. doi: 10.1109/TIEIC.1981.351050.

Nanoracks (2021): Nanoracks ISS Deployment. Available at: https://nanoracks.com/products/iss-deployment/ (accessed Jan. 25, 2021).

Nanoracks (2017): Frequently Asked Questions. Available: https://nanoracks.com/resources/faq/ (accessed May 18, 2021).

NASA (2017): Additive Manufacturing and Characterization of Ultem Polymers and Composites. Available at: https://ntrs.nasa.gov/api/citations/20160001352/downloads/20160001352.pdf (accessed April 28, 2022).

NNU. (Jan. 19, 2017): YouTube: NNU's MakerSat-1 CubeSat Assembly. Available at: https://youtu.be/shLPETczsF4 (accessed Jan. 25, 2021).

Northrop Grumman (Feb. 26, 2020): Northrop Grumman Newsroom: Companies demonstrate groundbreaking satellite life-extension service. Available at: https://news.northropgrumman.com/news/releases/northrop-grumman-successfully-completes-historic-first-docking-of-mission-extension-vehicle-with-intelsat-901-satellite (accessed Jan. 25, 2021).

Parrish, J.: DARPA: Robotic Servicing of Geosynchronous Satellites (RSGS). Available at: https://www.darpa.mil/program/robotic-servicing-of-geosynchronous-satellites (accessed Jan. 25, 2021).

Copyright © A. Deepak Publishing. All rights reserved.

Patane, S., Schomer, J., and Snyder M. (2018): Design Reference Missions for Archinaut: A Roadmap for In-Space Robotic Manufacturing and Assembly, in Proc., 2018 AIAA SPACE and Astronautics Forum and Exposition, Orlando, FL, Sept. 17–19. Paper 2018–5188. doi: 10.2514/6.2018-5188.

Paus, F., et al. (2017): A Combined Approach for Robot Placement and Coverage Path Planning for Mobile Manipulation, in Proc., IEEE/RS J. Int. Conf. on Intelligent Robots and Systems, Vancouver, BC, CA, Sept. 24–28, pp 6285–6292. doi: 10.1109/IROS.2017.82o6531.

Piskorz, D. and K. Jones. (2018): On-Orbit Assembly of Space Assets: A Path to Affordable and Adaptable Space Infrastructure. The Aerospace Corporation. Available at: https://csps.aerospace.org/sites/default/files/2021-08/OnOrbitAssembly_0.pdf (accessed Apr. 25, 2022).

Putz, P. (1998): Space Robotics in Europe: A Survey. Robotics and Autonomous Systems, Vol. 23, Nos. 1–2, pp. 3–16. doi: 10.1016/So921-8890(97) 00053-5.

Reed, B., et al. (2016): The Restore-L Servicing Mission, in Proc., AIAA SPACE, Long Beach, California, Sept. 13–16. Paper 2016–5478. doi: 10.2514/6.2016-5478.

Rodgers, L., Nolet S., and Miller D. (2006): Development of the Miniature Video Docking Sensor, in *Proc. SPIE 6221, Modeling, Simulation, and Verification of Space-based Systems III*, Orlando, FL. May 31. doi: 10.1117/12.665258.

Sallaberger, C. (1997): Canadian Space Robotic Activities. *Acta Astronautica*, Vol. 41, Nos. 4–10, pp. 239–246. doi: 10.1016/S0094-5765(98)00082-4.

Smrekar, S. and Banerdt B. (2014): The InSight Mission to Mars. presented at the 8th Mars Conf., Pasadena, CA, Jul. 14–18. Available at: https:// www.hou.usra.edu/meetings/8th mars2014/presentations/Smrekar_InSight_8thMars_Missions.pdf (accessed Apr. 25, 2022).

Soulage M. et al. (2019): Flexible, High Speed, Small Satellite Production, in *Proc. Conf. on Small Satellites*, Logan, UT, Aug. 3–8. Paper SSC19–I–05. Available at: https://digitalcommons.usu.edu/smallsat/2019/all2019/268/ (accessed Apr. 25, 2022).

Spaceflight (2021): "Schedule and Pricing. Available at: https://spaceflight.com/pricing/ (accessed Mar. 13).

Steimle, C. and Pape, U. (2014): ISS External Payload Platform - A New Opportunity for Research in the Space Environment, in Proc. 40th COSPAR Scientific Assembly, Moscow, Russia, Aug. 2–10. Available at: https://ui.adsabs.harvard.edu/abs/2014cosp...40E3195S/abstract (accessed April 25, 2022).

Steimle, Per C. et al. (2014): Commercial Approach to Research Outside the International Space Station - A Small Size Precursor Service for Future In-Orbit Testing, in Proc., AIAA SPACE 2014 Conf. and Exposition, San Diego, CA, Aug. 4–7. Paper 2014–4255. doi: 10.2514/6.2014-4255.

Tsujimura, T., and Yabuta, T. (1989): Object Detection by Tactile Sensing Method Employing Force/torque Information. IEEE Trans. on Robotics and Automation, Vol. 5, No. 4, pp 444–450. doi: 10.1109/70.88059.

Uzo-Okoro, E. et al. (2020): Optimization of On-Orbit Robotic Assembly of Small Satellites, in *Proc. ASCEND 2020*, Online, Nov. 16–18. Paper 2020–4195. doi: 10.2514/6.2020-4195.

Uzo-Okoro, E. (2020): Characterization of On-Orbit Robotic Assembly of Small Satellites. Thesis: SM, Massachusetts Institute of Technology, Media Lab, May. Available at: https://hdl.handle.net/1721.1/130212 (accessed Apr. 25, 2022).

Virgin Orbit (2019): "Service Guide." Available at: https://virginorbit.com/wp-content/uploads/2019/09/ServiceGuide_Sept2019.pdf (accessed Apr. 25, 2022).

Weisbin, C. and Rodriguez, G. (2001): NASA Robotics Research for Planetary Surface Exploration. IEEE Robotics & Automation Magazine, Vol. 7, pp. 25–34. doi: 10.1109/100.894030.

Whelan, D., et al. (2000) DARPA Orbital Express Program: Effecting a Revolution in Space-based Systems, in *PROC. SPIE 4136, Small Payloads in Space*, San Diego, CA, pp. 48–56. doi: 10.1117/12.406656.

Copyright © A. Deepak Publishing. All rights reserved.

Williams, K. Elizabeth, B. Bossler, R. Hubbell, A. Tsang, D. Whitman, and S. Wirth (2021): Commercial Off-the-Shelf Infrastructure for a 1U CubeSat. Available: https://repository.arizona.edu/bitstream/handle/10150/613784/azu_etd_mr_2016_0238_sip1_m.pdf?sequence=1&isAllowed=y (accessed Apr. 25, 2022).

Copyright © A. Deepak Publishing. All rights reserved.

APPENDIX 1: Key Level 1 Requirements for Robotic Assembly

Requirements	Rationale
The robot arms shall perform CubeSat assembly functions in a volume of 0.17 m^3	Robotic arms and parts will be in a constrained environment with temperature control.
The robot arms shall sense, grasp, and assemble CubeSat components	Goal is functional 1U CubeSat assembly
The robot arms shall be able to maneuver forward/backward, up/down, left/right, in three perpendicular axes, combined with rotation about three perpendicular axes.	Six degree-of-freedom (DOF) arms with a kinematic configuration of yaw-pitch-pitch-pitch-yaw-roll. Desire 95% accuracy to satisfy sense and grasp requirements, including partial single-fault tolerance.
The robotic arm motors shall operate remotely without generating FOD or requiring maintenance in the space environment.	Brushless motors and simple sensors. In this work, we use a 30:1 gear ratio and 256-count magneto-resistive encoders.
Each robot arm shall be able to move a mass of 2 kg	The mass of a 1U CubeSat, which is the final assembled object, weighs a maximum of 1.33 kg (Cal PolySLO, 2014)
Each robot arm shall have a minimum arm length of 1 m and a maximum arm length of 2 3.5 m	The spacecraft locker volume is 61 cm x 92 cm x 31 cm. (Figure 1) accommodates 3.5 m in length
The robot arms shall use Inverse Kinematics algorithms to sense and reach components.	Inverse Kinematics is used to initialize a rotating angle for each servo. (Fairchild, 2006)
The robot arms shall use Velocity Kinematics for target position error correction	After comparing the current target position and goal position to output an error, Velocity Kinematics is used to calculate the updated rotating angles.
The robot arms shall be mounted on a static platform that is three-axis stabilized.	The spacecraft locker should be three-axis stabilized. Operation on a tumbling or uncooperative platform are not in the scope of this work.
The robot arms shall passively sense target position.	The system will use a camera to provides the pose of the 1U CubeSat
Requirements	Rationale
The robot arms shall perform CubeSat assembly functions in a volume of 0.17 m^3	Robotic arms and parts will be in a constrained environment with temperature control.
The robot arms shall sense, grasp, and assemble CubeSat components	Goal is functional 1U CubeSat assembly
The robot arms shall be able to maneuver forward/backward, up/down, left/right, in three perpendicular axes, combined with rotation about three perpendicular axes.	Six degree-of-freedom (DOF) arms with a kinematic configuration of yaw-pitch-pitch-pitch-yaw-roll. Desire 95% accuracy to satisfy sense and grasp requirements, including partial single-fault tolerance.
The robotic arm motors shall operate remotely without generating FOD or requiring maintenance in the space environment.	Brushless motors and simple sensors. In this work, we use a 30:1 gear ratio and 256-count magneto-resistive encoders.
Each robot arm shall be able to move a mass of 2 kg	The mass of a 1U CubeSat, which is the final assembled object, weighs a maximum of 1.33 kg (Cal PolySLO, 2014)

Copyright © A. Deepak Publishing. All rights reserved.

Requirements	Rationale
Each robot arm shall have a minimum arm length of 1 m and a maximum arm length of 2 3.5 m	The spacecraft locker volume is 61 cm x 92 cm x 31 cm. (Figure 1) accommodates 3.5 m in length
The robot arms shall use Inverse Kinematics algorithms to sense and reach components.	Inverse Kinematics is used to initialize a rotating angle for each servo. (Fairchild, 2006)
The robot arms shall use Velocity Kinematics for target position error correction	After comparing the current target position and goal position to output an error, Velocity Kinematics is used to calculate the updated rotating angles.
The robot arms shall be mounted on a static platform that is three-axis stabilized.	The spacecraft locker should be three-axis stabilized. Operation on a tumbling or uncooperative platform are not in the scope of this work.
The robot arms shall passively sense target position.	The system will use a camera to provides the pose of the 1U CubeSat

Copyright © A. Deepak Publishing. All rights reserved.