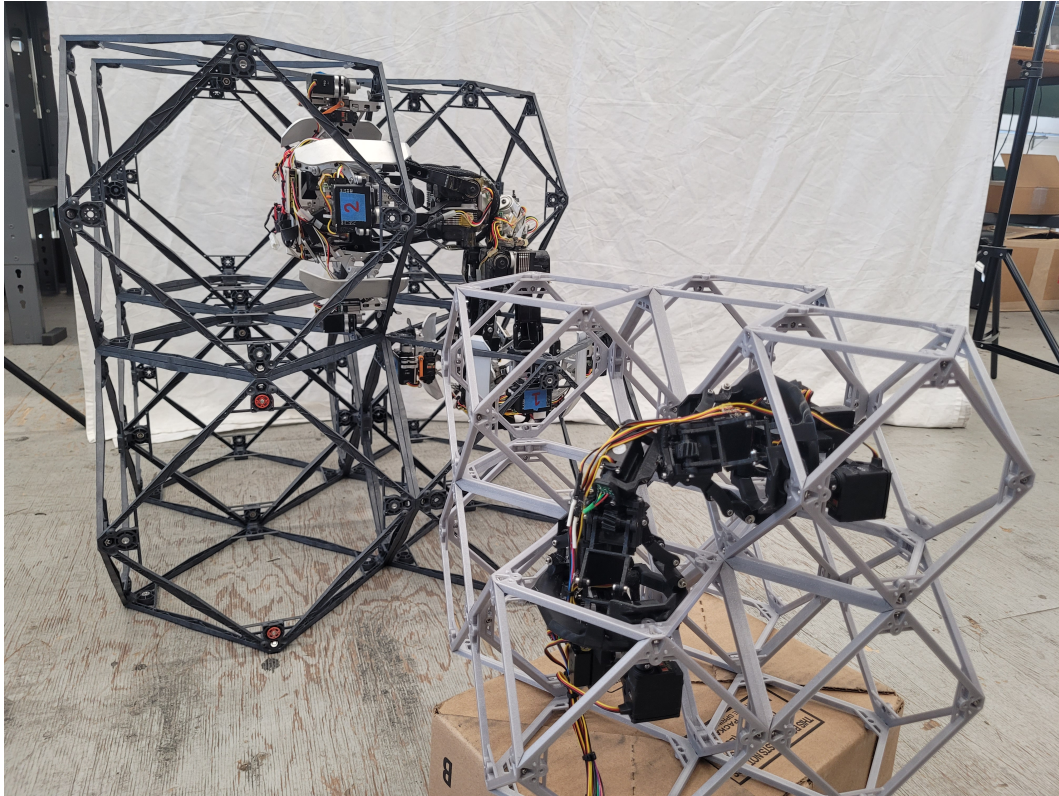




CHALMERS
UNIVERSITY OF TECHNOLOGY



Development of Downsized Bolter Robot for Lunar Structures

Iteration of MMIC-I Robot to Increase TRL for Autonomous Assembly and Locomotion through Meta-Material Structures

Product development, MSc

JAKOB MADESTAM

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

MASTER'S THESIS 2025

Development of Downsized Bolter Robot for Lunar Structures

Iteration of MMIC-I Robot to Increase TRL for Autonomous
Assembly and Locomotion through Meta-Material Structures

JAKOB MADESTAM



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Materials Science
Division of Product Development

IMSX30

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025

Development of Downsized Bolter Robot for Lunar Structures
Iteration of MMIC-I Robot to Increase TRL for Autonomous Assembly and Locomotion through Meta-Material Structures
JAKOB MADESTAM

© JAKOB MADESTAM, 2025.

On-site Supervisor: Dr. Kenneth C. Cheung, NASA
Supervisor: Prof. Johan Ahlström, Department of Industrial and Materials Science
Examiner: Dr. Lars Almfelt, Department of Industrial and Materials Science

Master's Thesis 2025
Department of Industrial and Materials Science
Division of Product Development
IMSX30
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Original MMIC-I (left) and newly developed mMMIC-I (right).

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Development of Downsized Bolter Robot for Lunar Structures
Iteration of MMIC-I Robot to Increase TRL for Autonomous Assembly and Locomotion through Meta-Material Structures
JAKOB MADESTAM
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

This master's thesis project in Product Development at Chalmers University of Technology presents the development of a downsized version of the MMIC-I robot, called mini MMIC-I (mMMIC-I), within NASA's ARMADAS project. The project was conducted during the spring of 2025 at NASA Ames Research Center in San Francisco, California. ARMADAS focuses on autonomous assembly of large structures in space using modular building blocks (voxels). With the goal of raising the technology readiness level (TRL) of the ARMADAS system and preparing for a potential CubeSat mission, this work investigates the feasibility of scaling down the MMIC-I robot to operate within a 1.5 U voxel size (150 mm x 150 mm x 150 mm).

The project followed an agile-inspired product development methodology, combining iterative prototyping, structured evaluation, and reflective learning processes. A two-week sprint structure guided the development timeline, enabling continual feedback and design refinement. The methodology included prototyping with increasing fidelity. Emphasis was placed on balancing fast prototyping cycles with sufficient analytical groundwork to avoid over-engineering within the project's limited four-month duration.

During the project, two iterations of mMMIC-I were prototyped and evaluated. While the latest prototype does not perform with the same reliability as MMIC-I, it can be concluded that through further iterations, a version that can perform reliably in 1.5 U voxels is possible. The project also discusses how different aspects of robotic simplification can be applied to the mMMIC-I design and what the required upcoming development steps before flight are.

Keywords: MMIC-I, ARMADAS, Robotics, Lunar, Down-Scaling, Redesign, Voxels, mMMIC-I, NASA, Design-simplification.

Acknowledgements

I would like to express my gratitude to several individuals and entities who supported me throughout my internship and this project.

First and foremost, I would like to thank NASA Ames Research Center, the Swedish National Space Agency, and the International Internship Program for giving me the opportunity to travel to the United States of America as a student intern. Of course, I also want to thank Dr. Kenneth C. Cheung, head of the Coded Structures Laboratory, for accepting me into his lab during my internship. His guidance, encouragement, and expertise were vital to both my professional and personal growth throughout this period.

I would also like to thank the Coded Structures Lab members who provided me with significant guidance and support during my research. Special thanks to Olivia Formoso, Irina Kostitsyna, and Frank Sebastianelli, who worked on the original MMIC-I design and could provide a lot of insight and information that was helpful to the process. Additionally, I am grateful to the other researchers and employees in the ARMADAS project group who provided valuable insights and assistance. Their collaborative efforts and willingness to share their expertise greatly enriched my learning experience.

Finally, I want to acknowledge my academic supervisor for this project, Johan Ahlström. Despite the time difference, his guidance supported me in the academic aspects of this project.

Jakob Madestam, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, listed in alphabetical order:

ARC	Ames Research Center
ARMADAS	Automated Reconfigurable Mission Adaptive Digital Assembly Systems
CAD	Computer Aided Design
CSL	Coded Structures Laboratory
DOF	Degrees of Freedom
LEO	Low Earth Orbit
MMIC-I	Mobile Meta-Material Interior Co-Integrator
mMMIC-I	Mini Mobile Meta-Material Interior Co-Integrator
NASA	National Aeronautics and Space Administration
PCB	Printed Circuit Board
SOLL-E	Scaling Omni-directional Lattice Locomoting Explorer
TRL	Technology Readiness Level

Contents

List of Acronyms	ix
1 Introduction	1
1.1 Background	1
1.2 Aim	3
1.3 Limitations	4
1.4 Specification of the Issue Being Investigated	4
2 Technical Overview	5
2.1 Technology Readiness Level	5
2.2 The CubeSat Standard	6
2.3 MMIC-I	7
3 Approach	11
3.1 Schedule	11
3.2 Background Research and Plan	12
3.3 Idea Generation and Idea Evaluation	13
3.4 Prototyping	14
3.5 Evaluation	15
3.6 Project Outcome	16
4 Implementation	19
4.1 Background and Requirement Specification	19
4.2 Idea and Concept Selection	21
4.2.1 Arm Module	21
4.2.2 Hip Module	23
4.2.3 Gripper and Bolter Module	24
4.3 First Prototype	25
4.3.1 Mechanism Calculations	26
4.3.2 Assembly	27
4.3.3 Control and Software	27
4.4 Evaluation, Improvements and Adjustments	28
4.4.1 Testing	28
4.4.2 Making Changes	29
4.5 Final Evaluation	30
4.5.1 Adding ESP32	30

5	Results	33
5.1	The Robot	33
5.1.1	Hip Module	34
5.1.2	Arm Module	35
5.1.3	Bolter Module	36
5.1.4	Gripper Module	37
5.2	Latest Evaluation	38
6	Discussion and Conclusion	41
6.1	mMMIC-I Design	41
6.1.1	Module Design	41
6.1.2	Result Interpretation	42
6.1.3	Next Steps	42
6.2	Simplification	43
6.3	Symmetry Dilemma	44
6.4	Methodology Evaluation	45
6.4.1	Agile Sprint Duration	45
6.4.2	Prototyping vs Evaluating	46
6.5	Concluding Thoughts	47
	Bibliography	49
A	Specification of Requirements	I
B	Evaluation Log 1	III
C	mMMIC-I Schematic	V
D	mMMIC-I Bill of Materials	XI
E	Evaluation Log 2	XV

1

Introduction

This is the final master's thesis report for: Development of Downsized Bolter Robot for Lunar Structures, which was conducted during the spring semester of 2025 at Chalmers University of Technology within the Product Development Master's Program. The thesis project was conducted at NASA Ames Research Center (ARC), located in Mountain View, California. The Coded Structures Laboratory (CSL), which hosts the project, focuses primarily on the Automated Reconfigurable Mission Adaptive Digital Assembly Systems (ARMADAS) project.

1.1 Background

National Aeronautics and Space Administration (NASA), established in 1958, is a leading force in space exploration, aeronautics, and scientific research (NASA, 2025). Over the past 65 years, NASA has contributed to the development of human and robotic spaceflight while advancing technology and improving our understanding of Earth, the Sun, and the broader universe. Through initiatives like the Artemis program, NASA aims to return humans to the Moon and pave the way for exploration of Mars. The agency also supports a low-Earth orbit economy, operates the International Space Station, and develops innovative technologies to improve air travel and space exploration. NASA's mission remains focused on expanding knowledge and advancing technology to support scientific discovery and exploration for the benefit of humanity.

The ARMADAS project, developed by NASA, addresses the growing need for sustainable infrastructure in space to support future deep space exploration, such as missions to the Moon and Mars (Figliozzi, 2024). Currently, sending large, pre-assembled hardware from Earth is costly and inefficient. ARMADAS aims to solve this by creating autonomous systems that can build large structures in space using small, modular building blocks called voxels (12 in x 12 in x 12 in) (Gregg et al., 2024). These robots work together to assemble a variety of functional structures, including habitats, antennas, and spaceports, without the need for human intervention. By using a modular approach and simple, reliable robots, ARMADAS offers a flexible and cost-effective solution to build and reconfigure infrastructure in space, making it crucial for long-term space exploration and missions that require scalable, reconfigurable, and adaptive structures.

1. Introduction

The autonomy and robotics group in CSL focuses on two main robots: Scaling Omnidirectional Lattice Locomoting Explorer (SOLL-E) and Mobile Meta-Material Internal Co-Integrator (MMIC-I). SOLL-E is a bipedal robot designed to navigate voxel structures (Park et al., 2023). It uses a combination of foot modules, joint actuators, and a cargo backpack to grip the surface of a voxel and move while carrying other voxels as payload. MMIC-I is a robot that moves through voxel structures, aligning and bolting new voxels to the existing build (Formoso et al., 2023). SOLL-E delivers the voxels while MMIC-I handles the exact placement and bolting to each adjacent voxel. A swarm of these robots would work together to complete large structures.

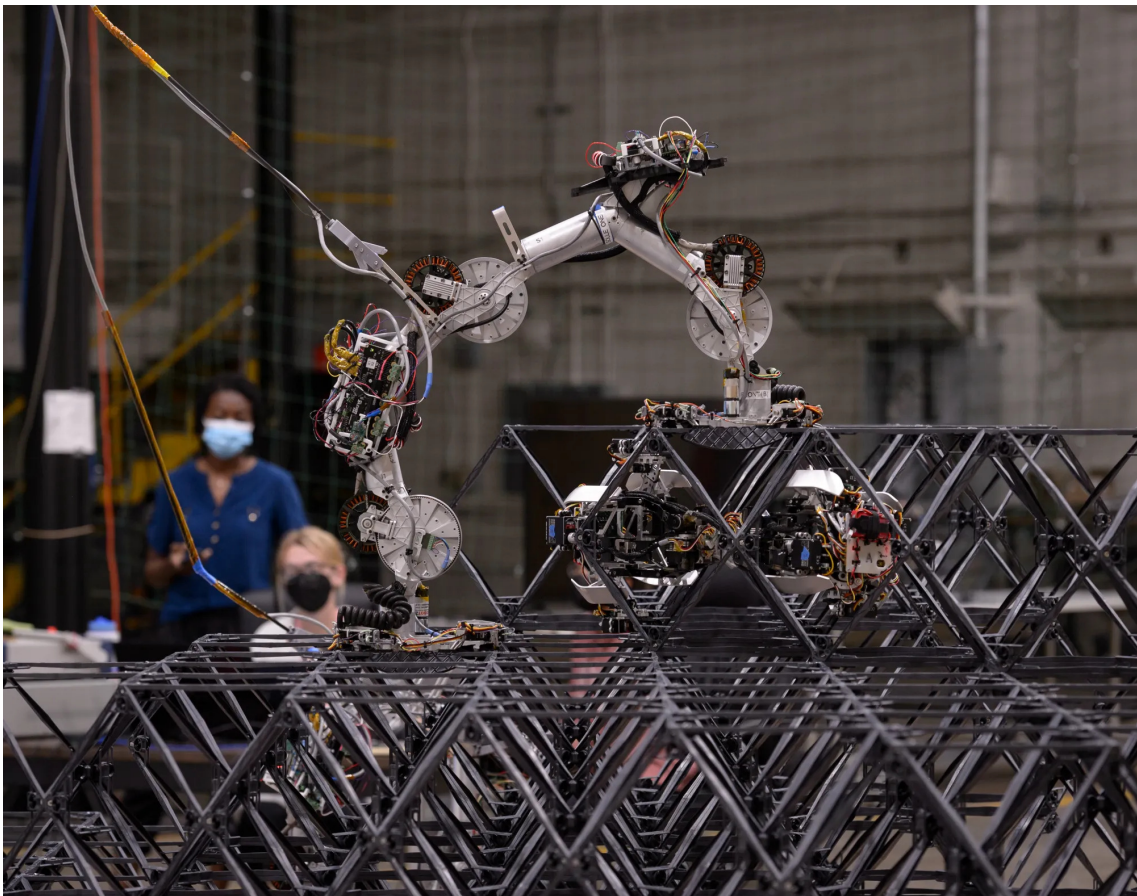


Figure 1.1: Robots SOLL-E (left) and MMIC-I (right) in action, autonomously assembling a voxel structure. The structure takes them over 100 hours to complete (Figliozzi, 2024).

NASA is working closely with the technology readiness level (TRL) scale. SOLL-E and MMIC-I have both reached TRL 4-5, meaning that a component and/or bread-board validation has been made in a relevant environment (Manning, 2023). To climb higher on the TRL scale, to 6+, the concept needs to be tested in a space environment. The space environment sets additional requirements on the robots. For example, they need to be suited for the temperature, vacuum, and radiation. A cheap and easy way to test hardware in a space environment is on a CubeSat

satellite. Hardware will need to be adjusted to fit the standards of the form factor. It has previously been decided in the ARMADAS project that if a CubeSat mission should be an option, the voxel size needs to be shrunk down to 1.5 U (150 mm x 150 mm x 150 mm). See Section 2.2.

The SOLL-E has a descendant (currently under development) called MOLL-E, a sleeker and optimized version of the robot developed by HIBI Robotics. The electronics used are in the process of getting space-qualified. Another benefit of this new iteration is a simplified design that allows for cheaper mass production. This redesign process has not been started for the MMIC-I yet. The main reason for this is its higher complexity of MMIC-I compared to the SOLL-E.

1.2 Aim

The project will iterate on the MMIC-I robot. The goal is for this new version to be able to operate in half-size voxel structures. Hence, the aim is to evaluate if such a downscaling is possible for MMIC-I. This smaller version of MMIC-I will be referred to as mini MMIC-I (mMMIC-I).

To further move towards a space-qualified product, the project aims to reduce the complexity of MMIC-I and make sure it can operate in the CubeSat environment. This includes creating and following a new specification of requirements for mMMIC-I.

The overall purpose of mMMIC-I is to raise the TRL level of the ARMADAS concept to 6+. However, the goal is not to have a flight-ready product by the end of this thesis project. Instead, the aim is to iterate on a solution that lands as close as possible to meeting the set requirements and lays the foundation for future iterations to result in a flight-ready version.

Working on this master's thesis at NASA is a good chance to grow both professionally and personally, especially in the fields of aerospace and robotics. Personal goals for the project include:

- Collaborating with experts at NASA provides the opportunity to learn from some of the best in the industry while also broadening the perspective on how aerospace and robotics are evolving globally.
- The project aims to dive deeper into robotic systems, improving personal skills in design, electronics, and robotics.
- Being part of NASA's mission to explore the Moon and beyond is a huge motivator, and the project aims to play a meaningful role in the progression of the overall ARMADAS project.

1.3 Limitations

Several limitations have impacted the scope and execution of this thesis project. First and foremost, there is a strict deadline for completion at June 15, 2025, which includes the submission of the thesis report. This provides an approximate 4 months of project time. Additionally, at the time of planning the project, there is a freeze on purchases for all United States federal agencies, which restricts prototyping to the components and materials already available at CSL. It confines the project to existing resources and hinders the acquisition of specific parts that may be needed for further development.

In regard to the project scope, the architecture of the MMIC-I will be kept the same. This means that the robot will maintain the same general body parts, while the mechanism allowing each part to fulfill its function may change. mMMIC-I must also be able to navigate the original voxel design. Any changes to the voxel design are considered out of scope. Additionally, the control software will not be included in this thesis project. While minor testing software will be utilized, the design focus is primarily on the physical and mechanical aspects of the robot.

One functionality of MMIC-I that will be disregarded in this project is the bolting aspect. This is due to two reasons. The first one is time constraints. The second reason is that the voxel bolting principle has not yet been decided for the half-sized voxels. Therefore, an exact bolting tool can not be implemented. To make sure the bolting mechanism can be accommodated, bolters will be added with an arbitrary end effector. This will provide enough fidelity to prove that size and alignment can be accommodated, but the bolting itself can not be investigated.

1.4 Specification of the Issue Being Investigated

The specification of the issue being investigated is as follows.

- How can each of MMIC-I's body parts be simplified while maintaining the original functionality and performance?
- What are the important requirements that mMMIC-I needs to meet before being able to operate on a CubeSat?
- Is scaling down MMIC-I possible? What are the limiting factors?

2

Technical Overview

This chapter covers some of the technical information that tangents this project. The following sections will elaborate on how these topics are relevant for the project goals and limitations.

2.1 Technology Readiness Level

TRL (technology readiness level) is a classification that describes the current state of a given technology (Burkhard et al., 2025). The scale goes from 1 to 9, where 9 is the highest. The NASA Systems Engineering Handbook describes in detail how each level should be assessed. Figure 2.1 shows the meaning of the different levels.

The ARMADAS robots are currently at TRL 4-5 since they have no flight heritage. To raise the TRL level to 6+, flight heritage is required. If a different configuration of the system is sent to space, the TRL of the original system remains at 5 until the differences in architecture and environment can be analyzed. A successful analysis could raise the TRL of the entire system to 6 or 7 (Hirshorn, 2007).

Why is the TRL so important? Before sending a technology on sharp missions to the moon, it is important to make sure the chances of the technology working as intended is as high as possible. Preferably, the TRL should be around 8 and flight-qualified before implementing it into existing and mission critical technology systems (Manning, 2023). Hence, scaling down the ARMADAS system to fit in a CubeSat is a logical next step to increase the TRL of the whole system and increase the the probability of this technology being utilized on the Moon.

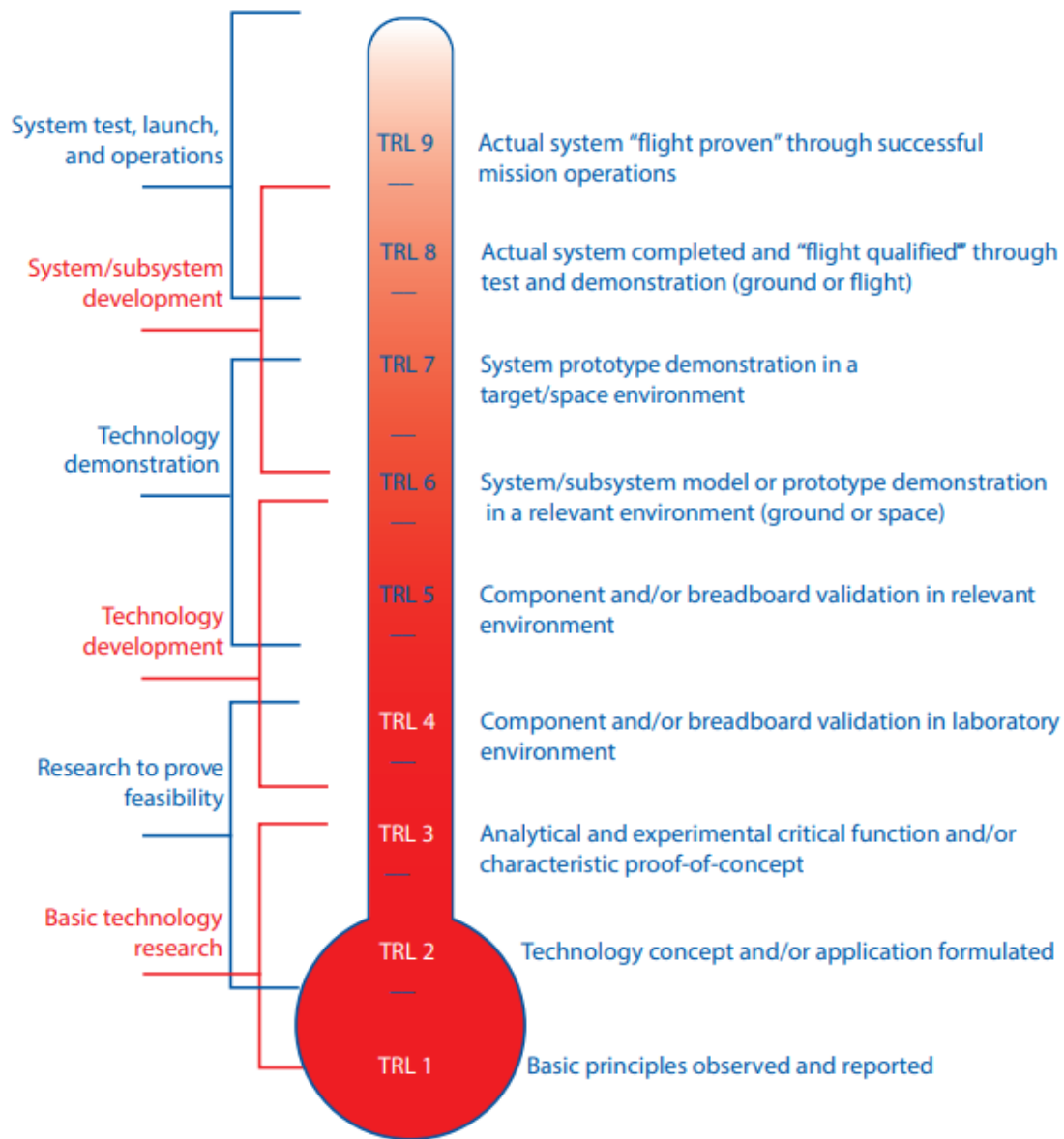


Figure 2.1: Description of each TRL level (Hirshorn, 2007).

2.2 The CubeSat Standard

CubeSat is a small satellite standard that is commonly used (Burkhard et al., 2025). Due to its small and standardized shape, it is a cheaper alternative for low-budget and short-term missions to LEO (low earth orbit). The CubeSat comes in standard sizes with unit U. 1 U (the smallest standard) is a 100 mm x 100 mm x 100 mm cube. These unit cubes can be added together to form larger form factors, where the largest, still commonly used, size is 27 U (300 mm x 300 mm x 300 mm). Figure 2.2 compares some of the more common CubeSat sizes.

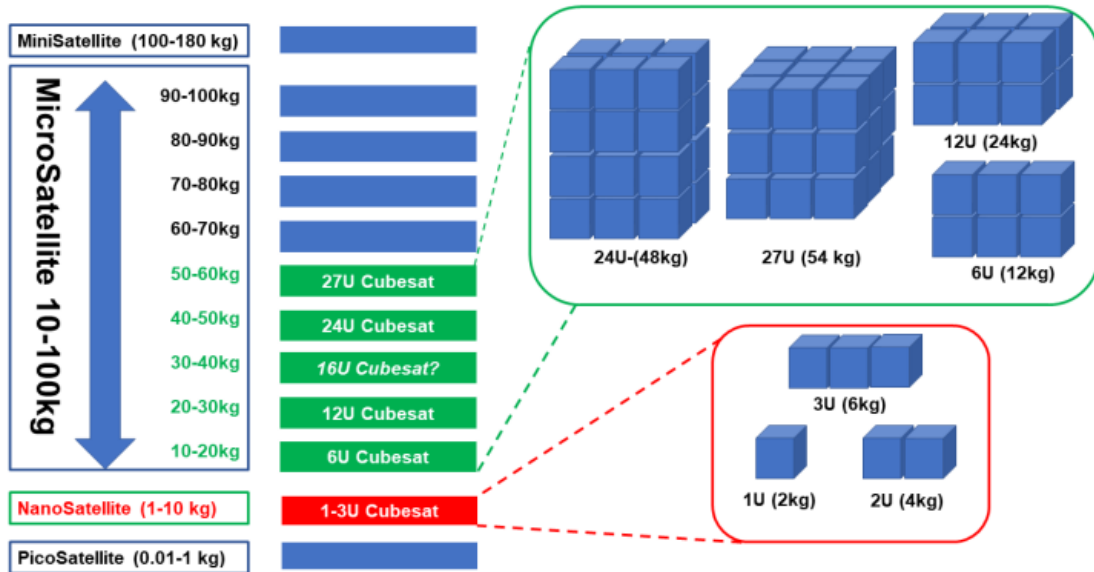


Figure 2.2: Size comparison between CubeSat standards. (Burkhard et al., 2025).

Previous assessments in the ARMADAS project have indicated that to properly test the technology, 8 voxels ($2 \times 2 \times 2$) will be required to produce sufficient results to increase the TRL. Currently, a standard voxel has a side of 12 in (≈ 305 mm). To fit 8 voxels in the 27 U CubeSat, the voxel sides can only be 1.5 U or 150 mm. Hence, the concept of half-sized voxels and mMMIC-I.

When developing hardware for space missions, it is important that the hardware meet NASA’s standard requirements (NASA Technical Standards Program, 2025). There are guidelines for everything from the electronics to the structural materials used. It is a bit different for a CubeSat. Based on internal knowledge, regular 3D-printed plastics and off-the-shelf electronics will suffice for an eventual ARMADAS CubeSat mission. This makes a CubeSat an even better alternative for ARMADAS, since time and money do not need to be spent on space-qualified hardware for a test flight.

2.3 MMIC-I

MMIC-I is a compact, highly specialized robot developed at CSL as part of NASA’s ARMADAS project at ARC. MMIC-I’s mission is to autonomously move through a growing 3D lattice of voxels and bolt new voxels to the existing structure. It works alongside an external delivery robot called SOLL-E, which transports voxels from a depot to the structure. SOLL-E places the voxel roughly in position. MMIC-I then fine-aligns it and performs the fastening operations needed to lock it into place. The bolting system can use many different types of mechanisms located at each corner of the voxel faces. At the moment of writing this report, the goal is to use regular bolts as fasteners. The fasteners are pre-installed and require MMIC-I to apply an axial push and a torque to lock each connection securely (Formoso et al., 2023).

MMIC-I moves inside the lattice using an inchworm locomotion style, ideal for the repeated geometric pattern of the structure. It consists of several key modules (see Figure 2.3):

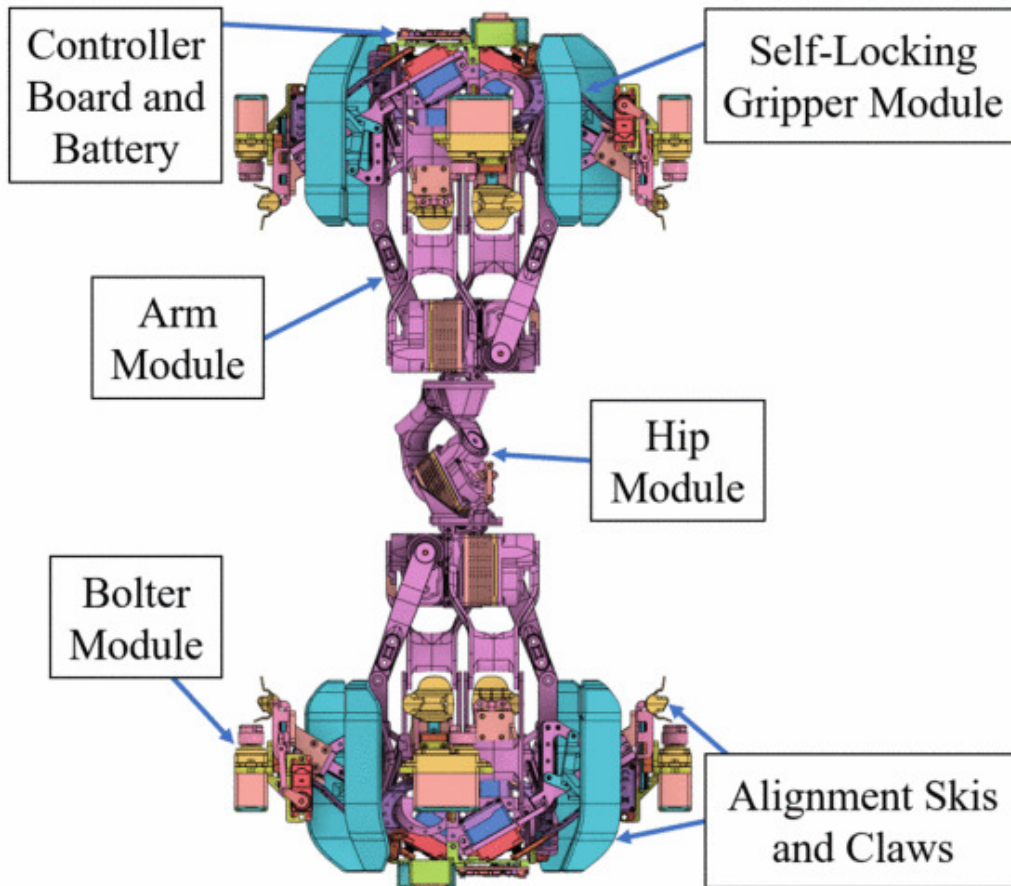


Figure 2.3: The MMIC-I robot in extended configuration. Body parts and features are highlighted. (Formoso et al., 2023).

- **Arm-modules** - Using a Sarrus linkage, the robot extends and contracts through the open faces of voxels.
- **Hip-module** - Located centrally, it allows MMIC-I to turn 90° , enabling movement in any orthogonal direction for full 3D navigation.
- **Gripper-modules** - Grippers extend out onto the voxel faces, locking the robot in place during movements and bolting. They also perform the final alignment for new voxels.
- **Bolter-modules** - Integrated into the gripper arms, these extend out to push and turn the fasteners to secure voxel faces together.

The robot moves by gripping a voxel face, extending its body, gripping the next face, and releasing the original, stepping from one voxel to another. For turns, it partially extends and rotates at the hip before continuing.

MMIC-I is built primarily of aluminum and lightweight plastics, weighing about 3.5 kg. It is densely packed with electronics and can therefore not be scaled down in its current configuration. The Sarrus linkage, for example, uses four actuators to drive it. This was the best solution at the time due to torque density and quicker prototyping during development.

3

Approach

This project follows a product development process influenced by agile methodology. One such influence is the CSL's two-week "sprint" structure, in which members present their progress at the end of each two-week period. These presentations are opportunities to receive feedback on the methodology, prototype, and data. In this project, these feedback sessions ran parallel to the other development phases described in Section 3.1. While the "sprint" summaries do not strictly follow agile procedures as outlined by Cooper and Sommer, 2016, they are based on the same values and still offer valuable peer feedback and allow for project pivots and refinements.

This chapter outlines the methodologies used throughout the project, describing the expected output of each stage and how it should be interpreted. Each phase includes its outputs and the rationale behind its design, drawing from established design methodologies and philosophical reflections on iteration and knowledge creation.

3.1 Schedule

The project is structured according to a Gantt chart (Figure 3.1), which organizes the timeline into phases. This temporal framing is essential due to the limited duration of the project (roughly four months). In that context, each stage must not only produce outputs but also yield insights that guide subsequent decisions.

The last steps are a series of prototyping and evaluation, iterating on the design to identify and address potential issues. If time allows, these steps can be further looped to find and address even more issues. This iterative model aligns with agile thinking, where each sprint yields a prototype that can be evaluated (Cooper & Sommer, 2016). The aim is to reduce lead time and increase design responsiveness. Given the complexity of the product and short project duration, only two full sprint cycles are planned. This should, however, be enough to establish a solid base and allow for a good knowledge foundation for supporting future iterations.

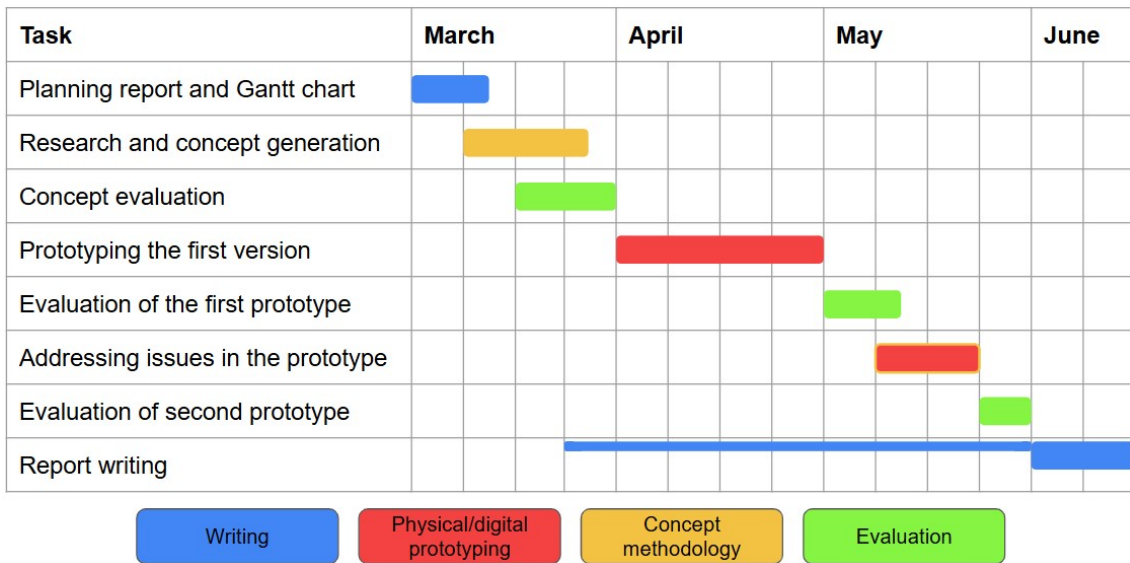


Figure 3.1: Gantt chart of the project process. The time before March was spent on background research and project defining.

This approach embraces the idea that knowledge often develops progressively. Following the notion of reflection-in-action (Schön, 1983), each phase is both an action and a learning moment as well as an opportunity to rethink assumptions and refine the methodology.

3.2 Background Research and Plan

Before defining what the new version of MMIC-I should become, it is crucial to understand what it already is. Comprehending the full architecture, purpose, and context of MMIC-I requires more than reading documentation. It involves studying related papers, reviewing internal documents, and having reflective conversations with those who previously developed it. The target is to define the specific goal with this project and identify why the requested improvements need to be done.

A new specification of requirements will be formulated based on this understanding. While an initial requirement specification already exists from the original MMIC-I project, this version must address a refined set of design goals. These include, among others, size constraints and LEO operation compatibility. The new specification must be both aspirational and realistic, acknowledging what can be accomplished within the time frame while also aiming toward a longer-term vision. More about the expected project result will be covered in Section 3.6.

The outcome of this phase will be an increased amount of knowledge about the project and the end-goal product. This, together with the specification of requirements, will later be used to drive the concept development and prototype evaluation.

3.3 Idea Generation and Idea Evaluation

MMIC-I is symmetrical and has 4 main unique body parts (Formoso et al., 2023). Based on suggestions from the previous developer, dividing the concept development into four parts (one for each module) would be effective. The development order should be arm, hip, gripper, and finally bolter (Figure 3.2). Beginning with the arm and concluding with the bolter allows for downstream dependencies to be more effectively managed. One design can not be decided before it is clear that other parts can accommodate it. This may slow down the concept validation, but it will reduce time spent on prototyping unusable designs.

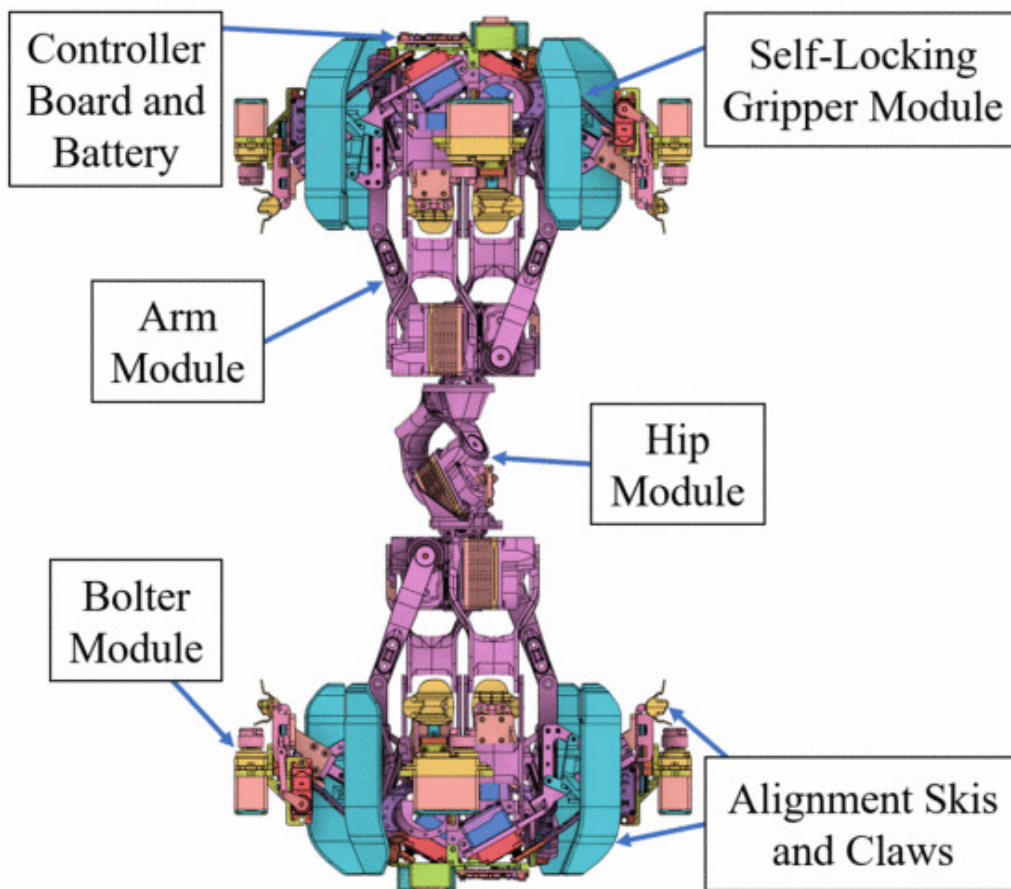


Figure 3.2: Current MMIC-I robot. Body parts and features are highlighted. (Formoso et al., 2023).

The concept generation will start with individual brainstorming. This is to reduce the influence of previously discussed solutions in the team. The realizability will then be evaluated together with other lab members who can provide more insights and lessons learned from the previous iterations. This process takes inspiration from the 5-step method described by Ulrich et al., 2020, but is adjusted to fit an individually conducted project by removing internal group sessions. After this, data-driven concept evaluation methods such as a Pugh or elimination matrix will be used to distinguish the better-performing concepts. It is important to note that

the evaluation does not only reduce the number of concepts; it can also spawn new ideas through realizations in the evaluations, merging, or splitting existing concepts. Even though the brainstorming session has finished, it is important to stay open to new perspectives. The idea funnel should look jagged as described in Wheelwright and Clark, 1993 instead of smooth. The final design decisions of what concept to continue exploring will be taken together with other researchers in the lab.

Several methods can be used to visualize concepts. Digital 3D visualization will be the main method used (due to personal preferences). While this method is a more time-consuming process compared to, for example, Post-it note drawings, it provides a higher fidelity of the concept. Since the robot contains many complex mechanical mechanisms, the higher fidelity can assist in communicating ideas with other researchers. Using digital 3D models also makes it convenient to 3D-print smaller mechanisms to validate general functionality.

The outcome from this phase is a lot of ideas and also an evaluation of these ideas. There will be one most promising concept that will be moving on to the prototyping phase. It is important to not completely trash the unused ideas, as they can be relevant again to pivot towards or to fix issues in the next prototyping iteration.

3.4 Prototyping

The first step of the prototyping methodology is a digital model. This serves two purposes. The first one is to check for basic functionality. For example, make sure links are not intersecting and the robot is within the size limit. The second purpose is that it sets up the concept to be manufactured physically. Ulrich et al., 2020 differentiates between analytical and physical prototypes. This digital version is classified as an analytical. The analysis of the digital model can be very extensive. For this project, it was opted to not dig too deep into it and instead move on to physical prototyping sooner. This is possible in this case because the robot is relatively cheap to prototype, and new meaningful tacit perspectives will be gained from the physical prototyping that are easy to miss if only focusing on the digital. Additionally, this avoids potential "analysis paralysis", as other lab members call it. It implies that too much analysis can lead to overthinking the problem and that things tend to work out when building the physical prototype.

CSL has a prototyping workshop with various tools such as 3D printers, laser and milling CNC, and a lathe. This provides good prerequisites to manufacture almost any mechanical component. Electronics-wise, there is a limitation to existing motors, encoders, and microcontrollers due to the purchasing freeze. Coding and detailed electronics, such as PCB design, are considered out of scope (see 1.3). However, some electronics and programming are required to evaluate the robot. Basic functionality, such as moving individual actuators, needs to be implemented. This is crucial to be able to evaluate mechanism functionality. The robot will be programmed in C/C++ using the Arduino IDE.

It is important to reflect on the fidelity of the physical prototypes. Should they be as close to a finished product as possible, or is it okay to take some "shortcuts" to be able to evaluate faster? Keeping the fidelity lower can result in faster prototyping and sooner proceeding to evaluation. What should be considered is how the lower fidelity will affect the test results. For this project, the initial prototype does not need to have the fidelity of a final product. The main purpose is to make sure that all mechanisms work both individually and together. Potential "shortcuts" that can be taken could, for example, be to use an external power supply instead of batteries or reduce functionality on one side of the robot (since it is symmetrical). The increased prototyping speed gained with the lower fidelity can help address the unanticipated problems often experienced with physical prototypes, explained by Ulrich et al., 2020. For iterations after the first one, when the functionality of the mechanisms is confirmed, the need to evaluate the prototype in more detail will increase. Hence, the fidelity of those iterations should increase along with it to make it comparable to the existing MMIC-I. The exact fidelity of the next iteration will be dependent on what has been improved and expected test results.

The output from this phase is both an analytical and a physical prototype, varying in fidelity. The prototype should represent the most promising concept that was previously decided. The next step is to evaluate the physical prototype through structured testing.

3.5 Evaluation

The next step is to evaluate the newly created/updated prototype. The tests should evaluate the functionality and effectiveness of the robot in regard to the requirement specification. The evaluation should be structured to make sure no aspects are missed. Therefore, this phase should start with evaluating and deciding what to test and how to test it.

Some of the more obvious functions to test are to see if the robot can locomote forward, turn, bolt, and unbolt. This is, however, not necessarily the best way to start, at least for the first prototype. The purpose of the first prototype is to validate the mechanism and make sure mMMIC-I works as intended. Even though calculations have been made with the analytical prototype during the prototyping, some things just turn out differently in the real world (Ulrich et al., 2020). Therefore, it is important to start even smaller. The evaluation should start by testing each core movement by itself, then add link after link until each module can be tested individually. Then, move on to higher levels, how the modules interact. Finally, evaluate full movements and their reliability. This will make sure each concept is being evaluated to its full potential.

The evaluation of some of the mechanisms will subconsciously be done while assembling the robot, which almost merges the prototyping and the evaluation. This opens the question: Should an issue that is discovered during prototyping be addressed directly, or should it be ignored until the entire system is done and is formally eval-

uated? This is a difficult question to answer, and it depends on the problem. The rule of thumb followed in this project is: If the change means a mechanism change or change in functionality, then the building commences. However, if the change does not fundamentally change the concept, such as stiffening a link that is too weak or splitting a part in two to enable the assembly, then the change is implemented right away. By doing this, problems that are easily fixed will not pause the development timeline. It is important that the concept is represented in the best way possible. The reason for ignoring some problems is that continuing to build the robot while knowing that there is a problem further down allows for assessing other components that may not be directly affected by the fault. Additionally, seeing the entire concept malfunctioning, and not just a specific part, can lead to new solutions being born when it can be viewed in a systems perspective.

Creating a good testing environment is important. It is particularly interesting to reflect on this scenario where the final product will be operated in LEO. Things we take for granted, such as gravity, will now be different. There are many ways around the gravity problem. One solution is to use pulley systems, among others, to mitigate the experienced gravity (Arakelian, 2016). However, the ARMADAS project wants to be able to showcase the technology on Earth and is considering terrestrial applications for it too. Therefore, no gravity mitigation system will be used. In general, if the robotic system can operate here on earth, it will most likely not run into any environmental problems during LEO. Based on internal knowledge, this is also true for other factors, such as vacuum, temperature, and radiation.

To document the experiments, video recordings, photos, and manual notes will be taken. These can then be compared to data from MMIC-I. Statistics such as success rate for bolting and locomotion steps can be compared with the data from the paper by Formoso et al., 2023. The output from this phase is therefore a list of performance data and potential improvement areas. These will be used in a new prototyping session to assess these problems and improve the functionality. Possible solutions for this will follow a similar methodology as the idea generation, but slightly accelerated.

3.6 Project Outcome

This methodology aims to develop a robot, mMMIC-I, that improves on the original MMIC-I design. The big question always remains: Will this really become the best solution? The reality is, there's probably no such thing. Engineering is all about trade-offs, and the most optimal solution is always shifting as new technology emerges and new constraints come into play. By using the methods previously outlined and staying flexible, this project aims to get as close as possible to an improved design within the given time frame. Even if mMMIC-I ends up not working at all, the project result will be the evaluation and lessons learned from the concepts that were tested. Failure is also progress.

There's an obvious difference between this project and the original MMIC-I devel-

opment. It took about three years to land on the current version, while this project only lasts for about four months. With such a short timeline, the focus is not on achieving perfection but on making meaningful improvements within the constraints. Some ideas that might have worked with more time will not make it into this version. But that is the nature of iterative design. It is not about getting everything right at once but about making steady progress towards something better.

The chances of this mMMIC-I version being sent to LEO are small. Additional iterations and evaluations will follow this project. As mentioned earlier in Section 3.2, the requirements specification made in the beginning is for the finished product. If the result of this project does not meet those requirements, it should not be considered a failure. It just requires more iterations to get there. The amount of development time required for that can be difficult to estimate before the project has started. After these four months have passed, more information about the challenges with this downscaling will be known. Hence, the project will result in a more accurate time estimation until a finished product can be made. At that point, it is also possible to make a decision if it is worth investing the required resources to complete the project or if there are other, more effective ways of moving forward. This thesis will not evaluate any decisions but mainly present the findings that a decision can be based on in the future.

3. Approach

4

Implementation

The implementation phase shows how all the information presented in Chapter 3 have been applied during the project. The process began with background research and the construction of a requirement specification. This is followed by idea generation and concept selection based on the previously concluded requirements. This concept was represented through a prototype that later was evaluated. The results from the test served as reference for the changes made to the prototype before evaluating it again.

4.1 Background and Requirement Specification

The implementation phase commenced with an initial familiarization process aimed at understanding the current state of the laboratory environment. This involved a review of relevant literature recommended by the on-site supervisor. These resources provided essential insights into the foundational concepts and highlighted the challenges addressed in previous research. In addition to reviewing the literature, interactions with other researchers in the lab were done to gain a practical understanding of ongoing work, operational procedures, and the overarching objectives of ARMADAS.

The first project-specific task was the development of a requirements specification list (see Appendix A). This list was primarily derived from the original design specifications of MMIC-I. Additionally, shortcomings identified through the literature review and discussions with lab personnel contributed to the formulation of these requirements. This approach ensured that the new iteration was both aware of past challenges and consistent with the overall project objectives. Among the various requirements, some were considered particularly significant due to their importance to this particular development project and relation to the research questions. These include:

- **Operation in half-size voxels** - The current version of the robot is packed really dense with components. Since the target of this iteration is a future CubeSat mission, a smaller version of the robot is desired. Scaling everything down with the current design is not possible. Scaling up the mechanisms, however, is easier. This reasoning implies that the new design concepts can

be relevant for utilization in the larger voxels too if they are shown to work in the half-sized ones. While specifying a size limitation even smaller than the half-size voxel is technically possible, it would not be useful for the current direction of ARMADAS. The half-size voxels are the smallest voxel size currently considered. While smaller ones may be relevant further on, the half-size voxels are the most logical size limitation for this project. Further analysis of the improved designs' down-scaling capabilities and limiting scaling factors could possibly assist in future development of a smaller iteration, if necessary.

- **Space-proof compatible** - To increase the TRL and to test the hardware in space, the components need to be space-qualified. Due to the current purchasing freeze, acquiring any components on the market is not possible. Therefore, the requirement specifies "space-qualified compatible". This means that while using accessible components now, switching to space-qualified components should not require a major redesign. Further information acquired during the development revealed that the components used in the design (including 3D-printed parts, HiTec servos, and ESP32 microcontrollers) can be utilized with minimal modification for CubeSat missions but will most likely need to be replaced for deep space missions, such as to the Moon and Mars.
- **Cheap manufacturing** - The cheap manufacturing is important for several reasons. First of all, the general principle of ARMADAS is to have cheap and simple robots with one task that can be mass-produced, compared to having fewer, more complex robots that can do multiple things. Making the robots cheap aligns with the ARMADAS project goals. While some consideration about this has gone into designing the existing version, it was not a requirement due to the focus being on functionality. Another of ARMADAS' goals is to collaborate with other parties to further improve hardware and software design. Having cheap hardware makes it easier for research groups and universities to contribute. Bringing down manufacturing costs can, for example, be done with the utilization of cheaper components, common part design, mass manufacturing methods, and reducing assembly complexity and time. While this requirement has high importance, like previously in the development, the focus will first be on confirming functionality and then looking deeper into manufacturing.

Each row in the requirement specification list contains a title, lowest acceptable and ideal values, a note that further explains the requirement, and finally an importance index. The values are mainly based on the performance of the existing model since the goal is to not have worse performance than what is currently achieved. Many of the requirements are related to each other. For example, the weight requirement and self-propelling requirement are related since a lighter robot will be easier to propel forward. These are still separated as individual requirements since the weight, for example, affects other aspects such as the balance of the voxel structure.

4.2 Idea and Concept Selection

As with many things, all parts of the robot affect each other. Therefore, the idea generation, based on suggestions from the previous developer, was broken down into 4 parts. This would start with the arm linkage, continue with the hip joint, followed by the gripper module, and finish with the bolter modules. All the parts followed mostly the same procedure for idea and concept generation. Subsection 4.2.1 explains the process for the arm module in more detail. It sets the general baseline for how the process went down for the other modules as well.

The idea and concept selection phase resulted in a rough 3D model containing all necessary mechanisms. That model remained digital since manufacturing of every single part had not been considered. Links were intersecting, and assembly would be impossible in many cases. The purpose of it is to showcase the concept principle and assure a size comparison between the modules. Following this, the concept model will be further refined in the prototyping phase.

The final concept was lacking consideration for wiring, electronics, and battery placements. The reason for not including these parts in the idea and concept selection is that they were classified as tiny details. The placement of small circuit boards and battery cells will not majorly change the function of the robot. Since the purpose of this first prototype is to validate the mechanical functionality, they will be added later. This does not mean that they are excluded from entire the project, just disregarded for the first iteration.

4.2.1 Arm Module

The arm module is the part that extends out from the hip joint, pushing the grippers to another side of the voxel. MMIC-I has two arm modules on each side of the hip. For the arm module to work successfully, it must be able to extend the gripper at least half the length of one voxel. For half-sized voxels, that is 75 mm. Important functionalities and requirements that are relevant to the module were noted down. Some of the most important ones are footprint size, number of required actuators, stability, total reach, and weight.

To gather ideas, brainstorming was employed. This was done at first single-handedly with minimal effect from outside sources. After this, outside sources were allowed to be used. This came mostly from looking at the intent and previous prototypes for existing mechanisms that could be employed in this scenario. All ideas were written down on paper in a combination of sketches and descriptive text. The ideas were later classified and grouped until 9 (in this case) unique ideas had emerged. These were further improved with digital 3D sketches using the CAD software SolidWorks. These 3D sketches assisted in the understanding of the more complex mechanisms.

To filter out the more promising ideas, a weighted Pugh matrix was used. The criteria were the ones originally identified to be the most relevant to this module. With the perspective of the Pugh matrix, a meeting was held with some of the other lab

members. Most of them had worked on the previous MMIC-I robot and therefore had a lot of knowledge about dos, don'ts, and previous concepts. The concepts were communicated mainly through the 3D sketches that helped the overall understanding. The outcome was more information, recommendations, slightly altered concepts, and even some new ones. This was taken back to the Pugh matrix that was altered with this new information.

All this had boiled down to 3 promising concepts. All of them were building on the general concept of using a Sarrus linkage, but all with different ways of driving it and with a different number of arms. The decision was made with the previous MMIC-I developer. Based on her recommendation, the 3-armed Sarrus linkage with a worm gear in the center moved on. One potential problem with this concept was that the grippers needed 90° rotational symmetry, and the arm now had 120° . Three arms was, however, superior in both weight and stability, but if necessary, the number of linkages could be increased to four.

To prove that this concept met all the requirements, a more detailed CAD model was made (see Figure 4.1). This model was still just a digital prototype and possessed several unrealistically thin parts, intersecting geometries, etc. The purpose was to make sure all the different modules could operate together without an unsolvable amount of interference.

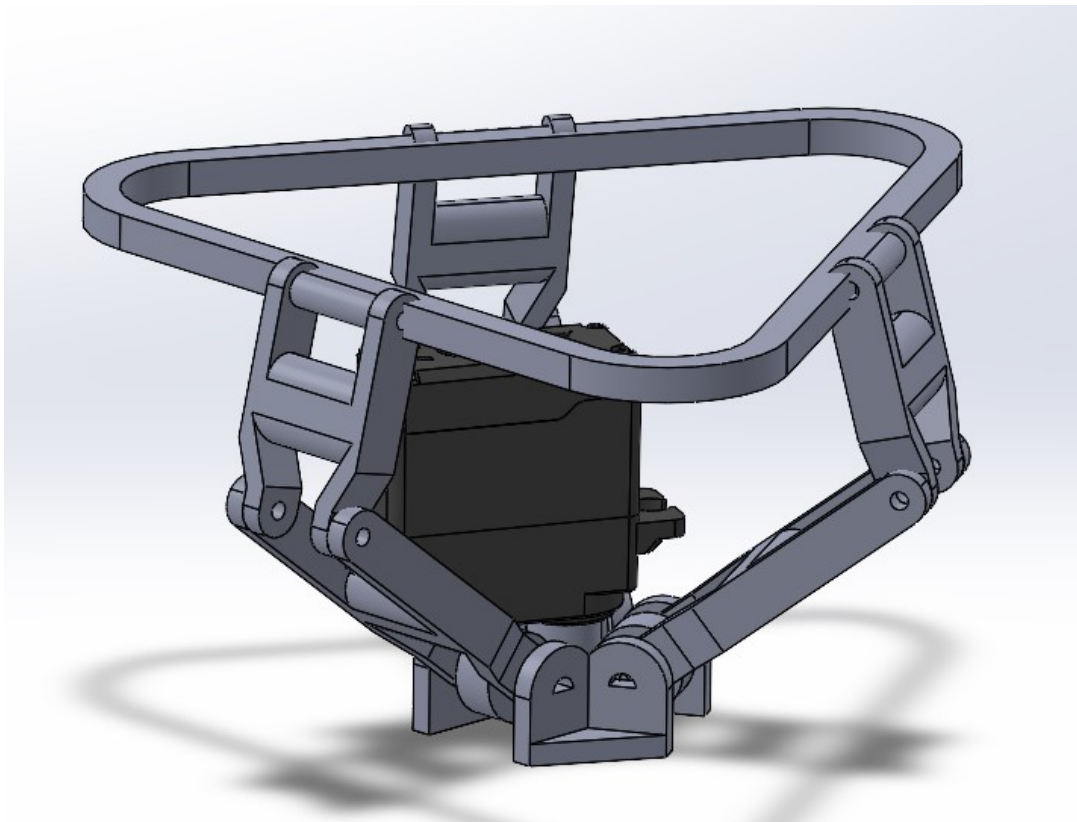


Figure 4.1: More detailed CAD model of the Sarrus linkage. This particular model can not be manufactured but can be compared with the other modules to assess the feasibility of the mechanisms.

4.2.2 Hip Module

The hip module is a much smaller part of the body that links the two sides of MMIC-I together. This is what allows the robot to make turns. Currently, MMIC-I can only turn right or up. Suggestions to make it able to turn to every side of the voxel had been brought up. This requirement was considered while idea generating. Compared to the arm module's process, this started off the same way.

Not long into the process, it became apparent that due to the enormously tight space and motor torque requirements, the feasibility of most ideas was severely limited. Space-wise, smaller is better, but it has to be small enough to allow the Sarrus linkage to go all the way down. This could be tested in SolidWorks. The required torque is more difficult to figure out. It is dependent on both the weight and the length of the robot. Due to the downscaling, the values from the existing model would not be accurate. Scaling those values down linearly is the closest estimate that was possible at that time since the rest of the design was not yet decided. It was suggested that slightly underestimating the required torque was better since it allowed more design freedom. If the motor is too weak, it will be addressed after the first prototyping session by either switching the motor or trying to reduce the overall weight.

After a meeting and a final Pugh matrix, the only reasonable choice is to continue using the same mechanism that MMIC-I utilizes. This works by rotating in the $[1, 1, 1]$ plane $\pm 120^\circ$ to go either left or down. This can be proven by assuming that the origin of the coordinate system is the center of a voxel (see Figure 4.2). Doing a rotation around $[1, 1, 1]$ for either 120° or -120° using Rodrigues' formula shows that a vector pointing straight forward ends up exactly on the right and top face (Equation 4.1).

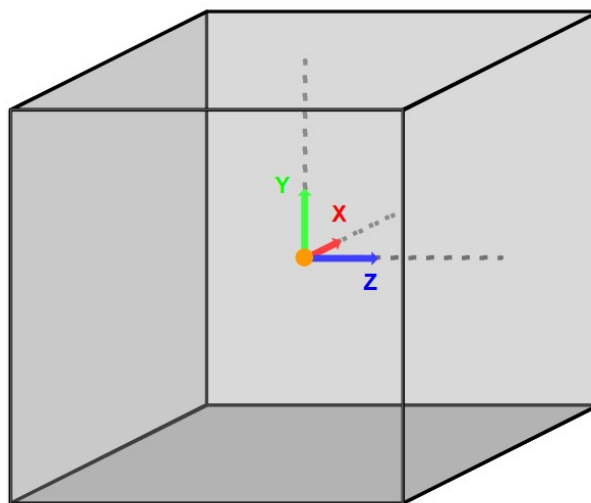


Figure 4.2: Coordinate system for Equation 4.1. The origin is in the center of the voxel. The unit length of the voxel is 2.

$$\bar{v} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad P = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbb{R}(\theta) = P \cos(\theta) + (\bar{v} \times P) \sin(\theta) + \bar{v} (\bar{v} \cdot P)(1 - \cos(\theta)) = \frac{1}{3} \begin{bmatrix} 2 \cos(\theta) + 1 \\ \sqrt{3} \sin(\theta) - \cos(\theta) + 1 \\ 1 - \sqrt{3} \sin(\theta) - \cos(\theta) \end{bmatrix}$$

$$\mathbb{R}(120^\circ) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \mathbb{R}(-120^\circ) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.1)$$

4.2.3 Gripper and Bolter Module

The gripper and bolter modules sit on top of the arm module and are what hold the robot in place and interact with the screws in the voxels. When reaching this part, it immediately became more complex. The interaction between the bolter and gripper is very intricate. After concept generation on the gripper module, it became clear that it was very difficult to evaluate the concepts without knowing how the bolter looked. Therefore, the gripper concept was put on hold, and idea generation for the bolter started.

One particularly interesting aspect that was discovered was about reducing the number of actuators. On the current MMIC-I version, for each gripper, there is one servo for the gripper, a second for aligning the bolter, and a third for engaging the bolting mechanism. Reducing the number of required actuators was a necessity to save space. For the bolter alone, there are at least three movement patterns that need to happen to have full functionality. This is pushing forward and rotating clockwise, moving back without rotating, and moving back while rotating counterclockwise. A motor that can only rotate in two directions can only produce a maximum of two movement patterns. This was confirmed with other members in the lab. To combat this without adding more actuators, there were two solutions. One is to use some kind of clutch. The clutch can be compliant, and a 3D-printed prototype was made (see Figure 4.3). This was, however, deemed to take up too much space. The second solution is to combine one movement with another function. An older concept showed a way to align the bolter every time the grippers are engaged. This concept was expanded upon, effectively merging the gripper and bolter module.

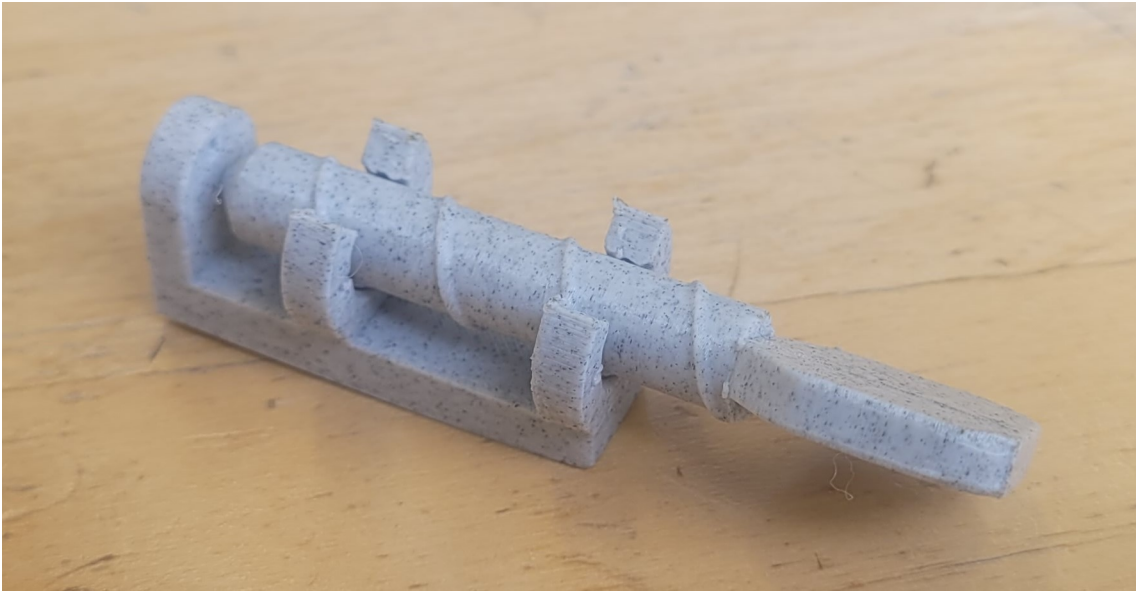


Figure 4.3: Prototype mechanism allowing rotation with helix in and out. When reaching the stop, the nut will flex, allowing the threads to slip. This acts as a compliant clutch, allowing three different movements with one motor.

There was no data for the required torque when bolting the voxels together. There are for the full-scale voxels, but no tests have been made for the half-size ones. The full-scale voxels require 2.52 Nm per screw. Assuming that the half-sized ones have the same physical properties compared to the larger ones and that we need to maintain the same clamping pressure, we can make a rough estimate about the half-size required torque (Eq 4.2). This is by using equations from Bickford and Nassar, 1998. Assuming that the screws for the smaller voxels will not be larger than the ones used for the full-size voxels, the required torque will be <25% of the original. The result is high but possible for the available motors in a feasible size. The HiTec D89MW fits the requirement.

$$M_{Half} \propto \frac{M_{Full} D_{Half} A_{Half}}{D_{Full} A_{Full}} \simeq \frac{M_{Full} D_{Half}}{4 D_{Full}} \quad (4.2)$$

The final concept consisted of a fusion of gripper and bolter, where the bit is spring-loaded to always fit in the voxel screw. All the grippers are engaged with a single motor. Each bolter has its own motor due to the high torque requirements for screwing the voxels together.

4.3 First Prototype

The first prototype was made based on the concept 3D model. Every part was looked over to make it suitable for 3D printing or laser cutting. This took a while since the first prototype consisted of 85 3D-printed parts. The reason for using these

manufacturing methods is speed and accessibility. Other manufacturing techniques could be considered in the future. At this stage, the focus was almost exclusively on functionality. For electronics, HiTec D85MG servo motors were used. For encoders, Poulou #3081. Finally, the microcontroller was an ESP32. The reason for this choice of components was accessibility. More powerful motors existed in the same form factor, which would make it easy to scale up if necessary. However, in case something breaks, it was more economic to use the cheaper version for this iteration.

4.3.1 Mechanism Calculations

Some of the mechanisms did not require any computation, only using links in one or two planes. Some, on the other hand, required calculations to function properly. These were the worm gear design and the angle of the hip servo.

The design for the gear system was based on RoyMech, 2020. To guarantee the worm gear to have 120° rotational symmetry, allowing each connecting gear to be identical, 3 thread starting points were used. Connecting to the worm gear are three regular spur gears. The gear module for all the gears was synced through SolidWorks equations. To make the spur gears fit the helix of the worm gear, they had a twist. This twist is calculated with equation 4.3. 3D-printing gears based on these guidelines is a workable but not an optimal solution. These guidelines only provided a basic mathematical basis for the gears. Additionally, the 3D-printing aspect does not have the same manufacturing tolerances as other methods, affecting the final performance of the gears. Ideally, off-the-shelf gears should be purchased and used for this project, but due to the purchasing freeze, this is not an option. 3D-printed gears also allow for easy trial and error. Adjusting anything is as easy as printing a new gear.

$$\beta = \arctan\left(\frac{m N}{D}\right)$$

$$\Theta = \frac{360 H \tan(\beta)}{m n \pi} = \frac{360 H N}{n \pi} \quad (4.3)$$

As explained in Section 4.2.2, the hip joint needs to rotate in the $[1, 1, 1]$ plane. This plane is 45° from all axes. This will, however, leave a much larger footprint than what is desired. Instead, the servo can be rotated just around the global x- and z-axes. As shown in Equation 4.4, rotating 54.74° around the z-axis and 45° around the x-axis makes the servo axis rotate in the $[1, 1, 1]$ plane.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_x(\theta_x) \times R_z(\theta_z) \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_z) \\ \cos(\theta_x) \sin(\theta_z) \\ \sin(\theta_x) \sin(\theta_z) \end{bmatrix} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \Rightarrow \cos(\theta_x) = \sin(\theta_x) = \frac{1}{\sqrt{2}} &\Rightarrow \theta_x = 45^\circ \\ \Rightarrow \cos(\theta_z) = \frac{1}{\sqrt{2}} \sin(\theta_z) &\Rightarrow \theta_z = \arctan(\sqrt{2}) \simeq 54.74^\circ \end{aligned} \quad (4.4)$$

4.3.2 Assembly

To make sure the robot works as intended, after a refined 3D model (analytical prototype), the parts were printed and assembled one at a time (see Figure 4.4). In some cases, incorrect printing tolerances or overlooked impossible assembly procedures meant that a smaller redesign of that specific part had to be done. To assemble the parts, common hardware such as nuts, bolts, and inserts were used.

The hip module, both arm modules, and the grippers were assembled completely on both sides. Only one side had the bolters to begin with. Since the initial testing will start with confirming the functionality of each module, it was not necessary to finish the other bolter until it could be confirmed that they worked. This did not only save time but also the cost of the components, most notably the four servos.

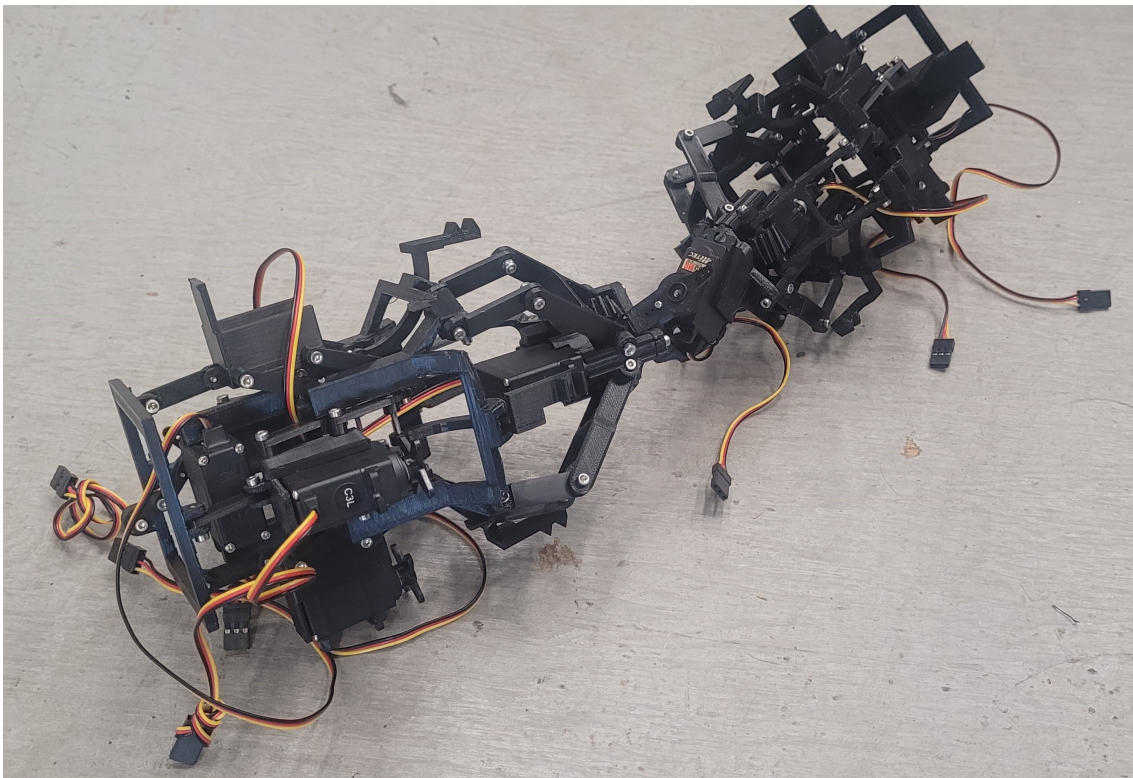


Figure 4.4: Finished assembly of the first iteration of mMMIC-I. In this picture, the bolter motors are missing from their slots on each side.

4.3.3 Control and Software

The prototype located all the electronics, besides the servos and encoders, externally. Those components were the only ones that were needed to ensure functionality of

the mechanisms. The robot was powered by an external power supply. An ESP32 was used to take readings from the encoders. While the ESP32 also could have been controlling the motors, the servos were instead controlled by an HFP-30 Field Servo Programmer. This gave off-the-shelf fine control over each servo. Since no special movement patterns needed to be included for the initial evaluation, this was a better solution to start with.

4.4 Evaluation, Improvements and Adjustments

With the first prototype done, the first evaluation could commence. While building the prototype, it was subconsciously evaluated, which steered the official evaluation planning. Some flaws in the mechanisms became apparent. Therefore, each module was tested separately at first. Based on the testing data, the 3D model was updated with the intention to solve the discovered problems. This was followed with the same updates to the physical prototype.

4.4.1 Testing

Each aspect that needed to be tested was noted down in a to-do list. Using the servo programmer, each motor was operated to test the mechanisms. The tests were video recorded, and all flaws (see Appendix B), but also the performance, were documented. Below follows a short summary with examples of how the tests were executed.

- **Hip module** - This module worked as intended. The tests included making sure that the rotations resulted in appropriate alignment of the robot and that the servo was capable of lifting its own weight (see Figure 4.5). Problems that were found included minor interactions with the arm linkages at certain positions.
- **Arm module** - This module also worked well. The tests included making sure that the mechanism worked as intended, measuring flex while extended, and locomotion through a voxel. It was noted that the arms interacted with the voxel in some positions when moving. There was also a problem with hole tolerances for the Serras linkage.
- **Gripper and bolter module** - This was the most problematic part of the robot. While the grippers managed to successfully hold the robot in place, there were alignment issues, clearance issues, and stability issues with the mechanism. In essence, due to the flex and friction in the joints, the actuator was not able to move all the sides of the mechanism equally.



Figure 4.5: mMMIC-I is attached to a voxel phase and shows that it is capable of lifting itself to a horizontal position. This position was used to measure the hip functionality, as well as flex in the arm and gripper modules.

In general, it could be confirmed that the hip and arm module worked as intended, only requiring minor changes. The gripper and bolter module needs to be reassessed. All encoders could be read successfully, even though the value was not utilized at the moment. The total weight of the robot was 583.6 g. This includes the microcontroller, batteries, and wires that were added externally for this version.

4.4.2 Making Changes

Regarding the hip and arm module, the problems noted were assessed one after the other. As it turned out, most of the interference problems could be solved by employing crescent-shaped links to avoid other components. Everything was adjusted digitally first, making sure the changes did not negatively affect other parts of the robot. Digital interference analysis made sure all links could move freely.

The mechanism for the gripper and bolter module needed to be reassessed. Similarly to how the idea and concept generation was made, solutions for the mechanisms were generated. One interesting aspect that kept coming up was to reduce both the number of grippers and bolters to two. The main concern with this was stability. Other concepts for MMIC-I with only two grippers experienced a lot of sag when being held horizontally. The current grippers on mMMIC-I could not support the robot with only two of them. A test with two significantly larger and altered grippers was printed. This was able to hold up the robot successfully. Flex in the gripper links resulted in a lot of sag, but that was expected for this quick prototype. A fully implemented version would require much stronger links. Due to the success of this quick test, the robot was updated with a two-gripper design. By reducing the

number of grippers and bolters, there is now significantly more space. This could be utilized for the microcontroller and batteries, allowing them to be implemented more easily.

Another update that also saved much space was to add an additional motor to actuate the grippers. Now, each gripper had its own motor to drive it. While adding more servos is not preferred, the intent is to reduce the instability in the mechanism experienced during the testing, increasing the reliability of the grippers and bolter alignment. Overall, the number of actuators from the first prototype has still been reduced since four bolters were removed. Since the motors only need to actuate one gripper each, it is possible to reduce the motor size. The HiTec HS-65MG+ was chosen for this task. They are both smaller and cheaper than the HiTech D85MG.

One important aspect that was noted during this prototyping was that it messed with the symmetry of the robot. More reflections on the symmetry are brought up in Section 6.3.

Implementing all changes was also done like the previous prototyping session. Parts were adjusted on the fly in case of minor problems, such as assembly or clearance problems.

4.5 Final Evaluation

The second prototype felt much better than the first one. There were no obvious flaws noticed during the prototyping that could guide the evaluation planning. The evaluation started from the beginning, like the first one, to make sure all the fundamental movements worked as they should. This was done with an HFP-30 Field Servo Programmer. It could confirm that all the servos were working as intended individually. With all the actuators working as intended, the next logical step is to move to higher-level functionality. To, for example, make the robot move forward, several of the servos need to be active at the same time. To avoid using 11 programmers (one for each servo), an ESP32 was implemented. This allowed the testing of how the different modules worked together with echoer. All movements that the robot could be imagined doing were tested one by one. Videos and notes were taken, and the results are displayed in Chapter 5. To move to an even higher level, the mechanisms needed to work reliably enough to add movement patterns that could be executed on a higher-level command.

4.5.1 Adding ESP32

The prototype has available space on either side of the grippers, on the part side body A and side body B, for microcontrollers and batteries. This version had not considered any exact mounting for either of them. To proceed, options such as just taping the components to the robot were considered. Another option was to keep the ESP32 separate and have a wire tether to provide the servos with signal and

power. This second alternative allowed for commands to be sent from a computer to the control board via USB, and an external power supply could be used instead of the batteries. This streamlined the testing since the batteries would not run out and network communication did not have to be implemented. The second option was implemented (see Figure 4.6). Plugging in all the servos activates them and makes them hold their position. The ESP32 was programmed to adjust the position of each servo, one at a time.

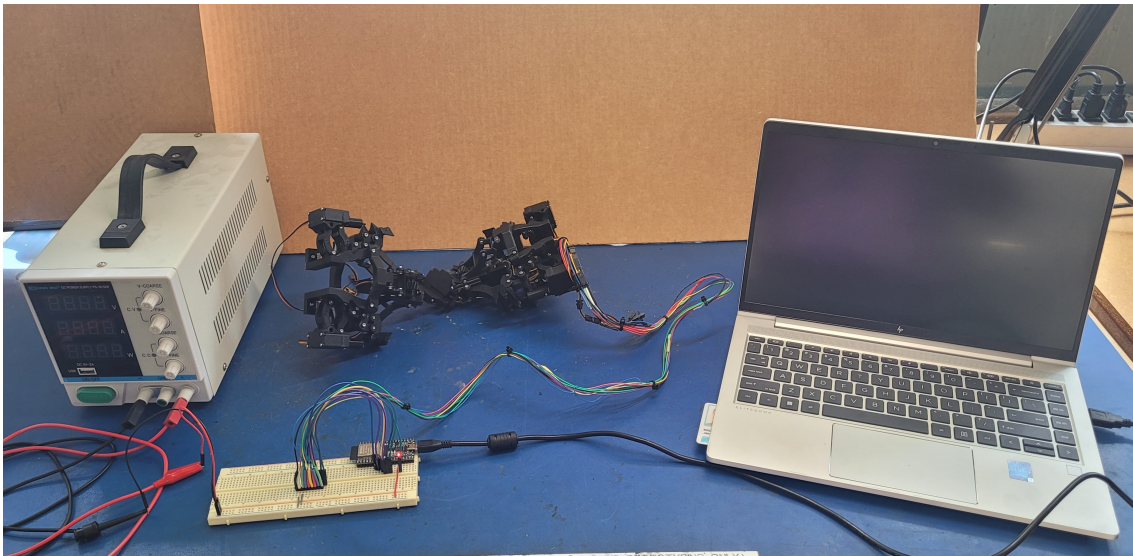


Figure 4.6: Testing setup for the second iteration of mMMIC-I. All cables from the motors are bunched up together and lead to an ESP32 to control them. The motor receives power from the power supply, and the ESP32 receives power and commands from the laptop.

In the previous prototype and during the first phases of testing, the wires from the servos and encoders have been hanging freely from wherever they happen to be. This time, the wires were pulled to one side. This was not a difficult endeavor even though cable management had not yet been considered in the design. To attach the wires during the temporary testing, a hot glue gun was used. While ground and power could be merged on the robot, it was not at this stage. The reason for this is to keep the option of changing the routing or switching out the components during testing. One problem with this was that instead of the minimum 17 wires, there were now 45 wires. The sheer volume of wires affected the mechanism's performance, and one of the arm linkages was completely inoperable. For that reason, the bolter wires, as well as the encoder wires, were removed. The bolters were not going to be tested either way since they are out of scope, and the encoders had already been tested in the previous version. This reduced the total amount of wires to 21, which did not affect the robot's performance.

5

Results

In total, two iterations of mMMIC-I were developed. This chapter will present the second (latest) iteration of the robot. It contains both a detailed description of the robot and the test results from the latest evaluation. mMMIC-I has both an analytical (see Figure 5.2) and a physical prototype (see Figure 5.1).

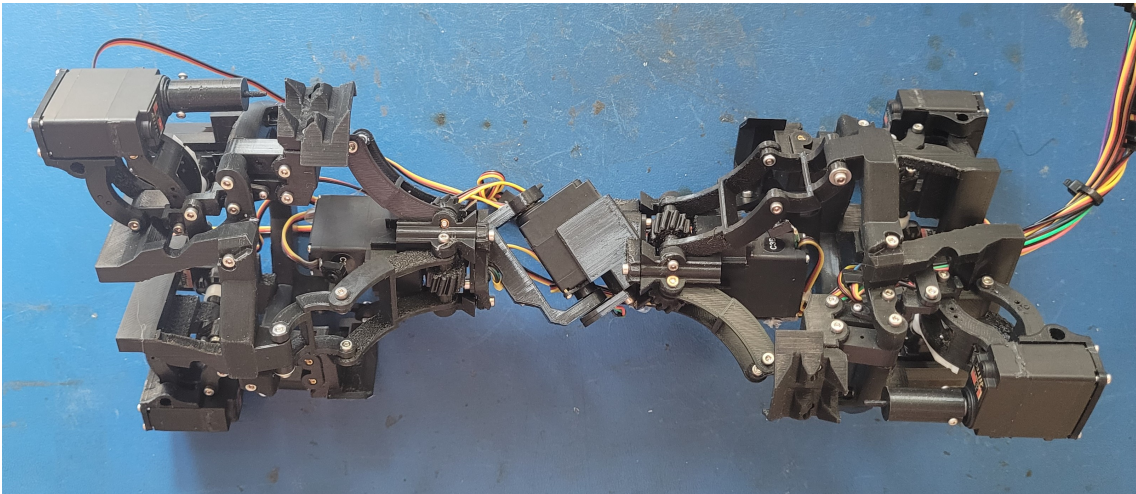


Figure 5.1: Finished assembly of the latest prototype (mMMIC-I second iteration).

5.1 The Robot

mMMIC-I has 4 different modules: Hip, arm, gripper, and bolter. While the high-level functionality of the modules is the same as MMIC-I, the mechanics differ. In Appendix C, each module is color-coded and labeled in the schematic. (See Figure 5.2). In total, mMMIC-I weighs 459.6 g and consists of 325 parts, of which 79 of them are custom 3D-printed parts. For a more detailed part breakdown, see Appendix D.

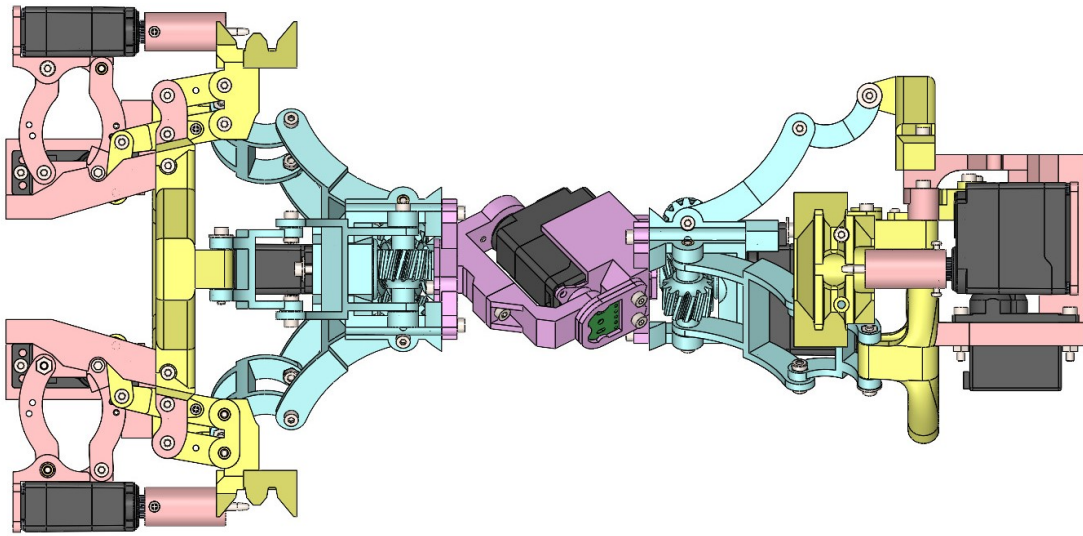


Figure 5.2: An excerpt from Appendix C showing a schematic of the entire mMMIC-I robot. Each module is color-coded.

5.1.1 Hip Module

The hip module is the core of mMMIC-I. It is what allows it to turn across planes. The mechanism that allows this is the exact same as on MMIC-I (see figure 5.3). For more details on how this mechanism works, see Section 4.2.2. The servo in the hip module is a HiTech D85MG that has a stall torque of 0.42 Nm. During the first prototype evaluation, this was proven sufficient. However, there exist servos like the HiTech D98MG that have the exact same form factor but are stronger and more expensive.

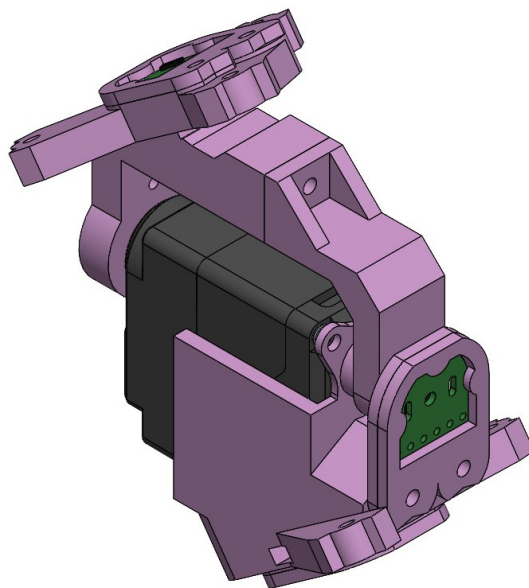


Figure 5.3: 3D model of the latest version of mMMIC-I's hip module.

Originally, the HiTech servos only had a 180° rotation span. The hip module requires 240° rotation. Therefore, the servos are modified to be able to rotate continuously, at the expense of absolute position control. To weigh up from this flaw, the hip module has a separate external encoder. The two other encoders present in Figure 5.3 are covered in Subsection 5.1.2.

The hip module itself is asymmetrical, but it has the same interfaces on both ends. This allows the same arm module to attach to both sides without modifications.

5.1.2 Arm Module

The arm module, similar to MMIC-I, consists of a Sarrus linkage. There are some fundamental differences. mMMIC-I's linkage consists of 3 arms that are all driven by one servo (see figure 5.4). The arms are curved to avoid hitting the hip module when closed and the voxel faces when extending in the structure. The specific width of the linkages is important to make sure the arms do not collide with other components as well.

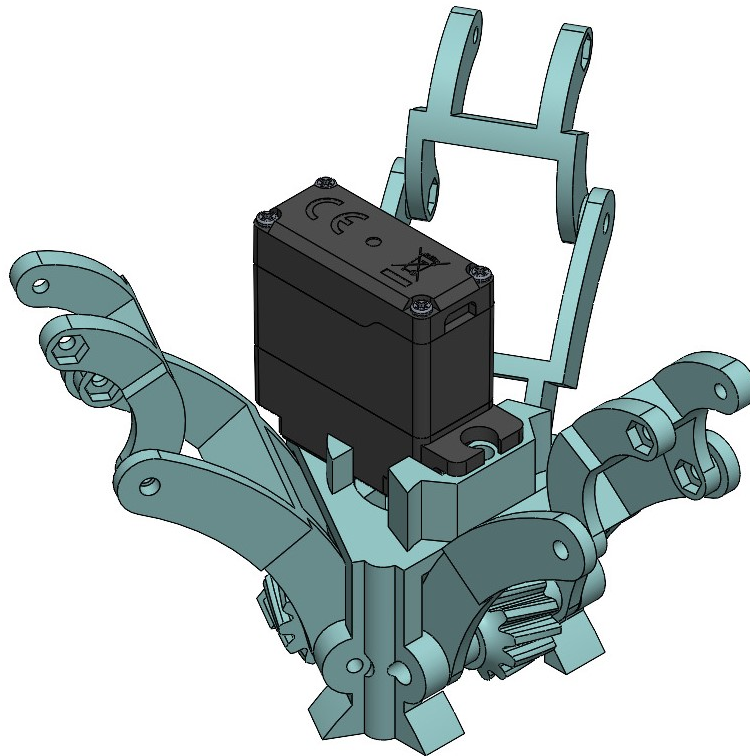


Figure 5.4: 3D model of one of the latest version of mMMIC-I's arm modules.

To drive all four arms at the same time, each arm module has a 4:1 worm gear to transfer the torque to the arms. The servo is a HiTech D85MG. Due to the gearing, the servo required the same modification as explained in subsection 5.1.1, and an encoder was installed underneath the gear (see Figure 5.3).

mMMIC-I has two arm modules. While the hip module makes sure they are rotated the proper way, the arm modules are identical, having no differences between the two sides. The modules are mostly 120° rotationally symmetric except for the motor housing.

5.1.3 Bolter Module

The bolter and gripper module is where mMMIC-I differentiates itself from MMIC-I the most (see Figure 5.5). The bolters are currently driven by HiTec D85MG. While these are too weak to sufficiently bolt the voxels together, the HiTec D98MW, which has the same form factor, is. Besides alignment and space accommodation, the bolting itself is out of scope for this thesis project. Therefore, the current servos are sufficient for this purpose.

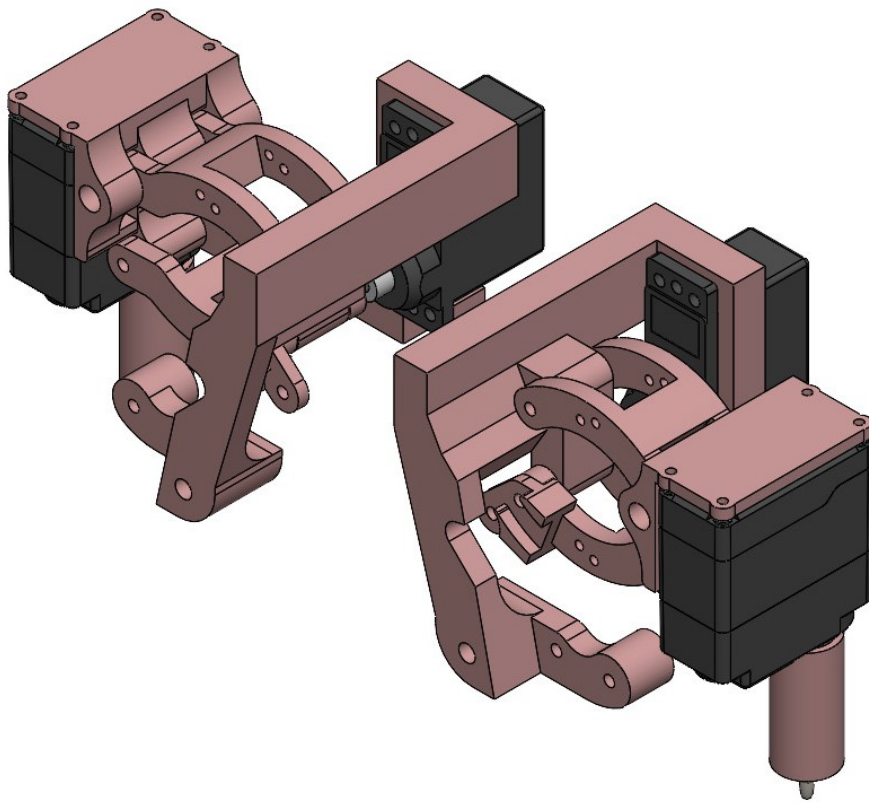


Figure 5.5: 3D-model of one side of the latest version of mMMIC-I's bolter modules.

On the bolter servos, there is a spring-loaded 4 mm precision bit. When the bolter moves into place, the spring makes sure the bit is adequately inside the bolt at all times. The servo is attached with a four-bar link that always keeps the motor parallel with the bolts. When moving towards the bolt, the positioning of the links allows the final movement to be as in-line as possible with the bolt to make sure

the bit hits where it should. Similar to the Sarrus linkage, these links are also cresset-shaped to allow the bolter to move further back.

The mechanism is directly driven by a HiTec 65MG+ servo, attached to one of the links. This link is also attached to the gripper module, allowing the bolter to align every time the gripper activates. Each bolter-gripper combo has one servo driving it, totaling 4 of these servos on the entire robot.

On each side, the bolter module is mirrored. All the mechanism parts are symmetrically mirrored to reduce the number of unique parts. The only side-specific part is the frame holding the mechanism together, named side body C1 and side body C2 in Appendix C and D.

5.1.4 Gripper Module

The gripper mechanism is attached to the bolter mechanism, making them actuate by the same HiTec 65MG+ servo. The gripper employs a similar four-bar link to the bolter module (see Figure 5.6). These links are significantly stronger to be able to hold up the weight of the robot without flexing. Also, similar to the bolter module, the sides of the gripper are mirrored, but the parts are mirror symmetric. This means that while the assembly is mirrored, it does not require side-specific parts.

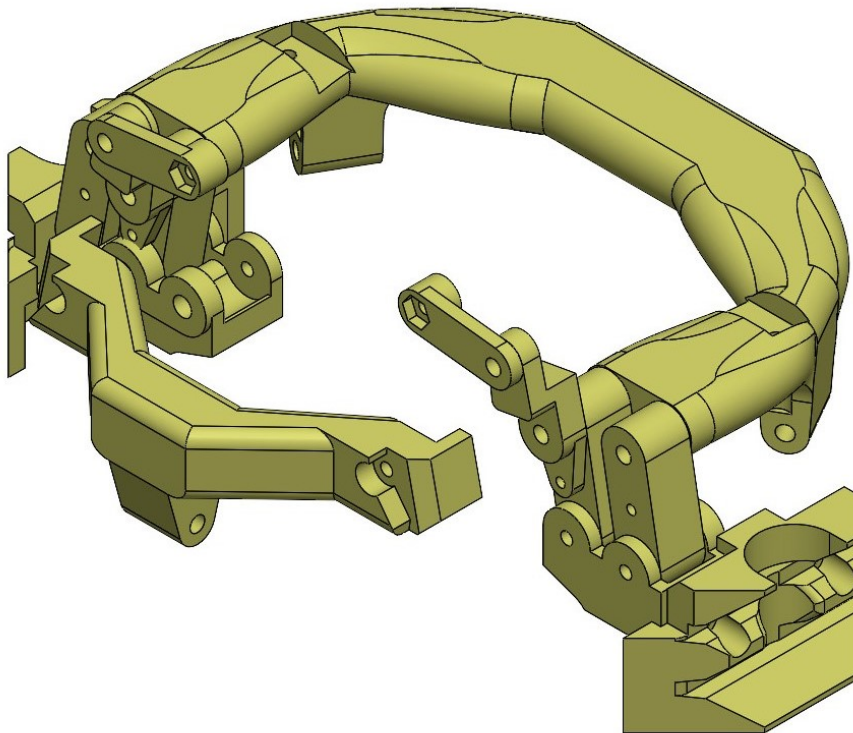


Figure 5.6: 3D-model of one side of the latest version of mMMIC-I's gripper modules.

Each side of mMMIC-I has two grippers instead of four like MMIC-I. This is to save space and actuators. The gripper interfaces are designed to hold on to the voxel structure in multiple dimensions to make sure the two gripping points are sufficient. On a face with two voxel interactions, the grippers hold on to both the faces for extra stability. Also, by holding on to both the faces, mMMIC-I automatically does the fine-tuning alignment for the voxels before joining them.

The gripper and bolter module are very intertwined. The gaps that can be seen in Figure 5.6 are where the bolter modules are supposed to go. The gripper module also attaches at the end of the Sarrus linkage from the arm module. The free space on each side of the gripper module where the mechanisms are not, microcontrollers and/or batteries could be placed. This was, however, not specifically designed for this iteration.

5.2 Latest Evaluation

During testing, notes were taken by both video recordings and manual notes. A compilation of the documented faults that need to be adjusted is displayed in Appendix E. The robot was able to move all the modules driven with the actuators. The encoders can also return the expected data. Three major test sequences were then performed on the robot through the ESP32. This showed the following:

- **Extending and contracting** - The robot was able to extend and contract itself to reach the other side of the voxels. While extending, the robot swayed a little, sometimes making it collide with the voxel phase, not being able to extend further. Manually aligning it allowed the test to proceed. Contracting also experienced an issue with one of the arm linkages. While contracting, the robot needed a light tap to fully go together. It seemed like some mechanism was colliding with something, but it could not be confirmed. Both the extending and the contracting problems were easily fixable with a light tap to realign the robot. The reason was not determined to be due to a design flaw, but the hole tolerances inhered from the 3D-printing manufacturing method.
- **Gripping and ungripping** - The robot was able to grip and ungrasp the voxels. It was also made sure that the mMMIC-I could support its own weight horizontally while being gripped. This was possible for both the vertical and horizontal gripper configurations. This test was also evaluated together with the extending and contracting, making sure the grippers could engage properly after an extension or contraction. A minor problem was experienced when the bolters were attached too. The spring-loaded bit was slightly too long. The grippers were able to close, but the spring became so contracted that it pushed back too much on the entire mechanism. If there was a bolt in the bolting hole on the voxel face, the gripper would not be able to operate as expected.
- **Turing** - The hip is able to rotate as intended. It is also able to move while the robot is attached with its grippers on one side. The evaluation did first take

place outside the voxel structure and showed that all the involved servos are strong enough. Then, mMMIC-I tried to turn inside the voxel structure. Even though, during the prototyping, the analytical prototype showed that a turn in the voxel was plausible by carefully adjusting the arm and gripper positions along with the hip, some overlooked interference between components made it unable to complete the turn. Either the bolter interfered with the voxel face, or the arm module interfered with the arm module on the other side (see Figure 5.7).

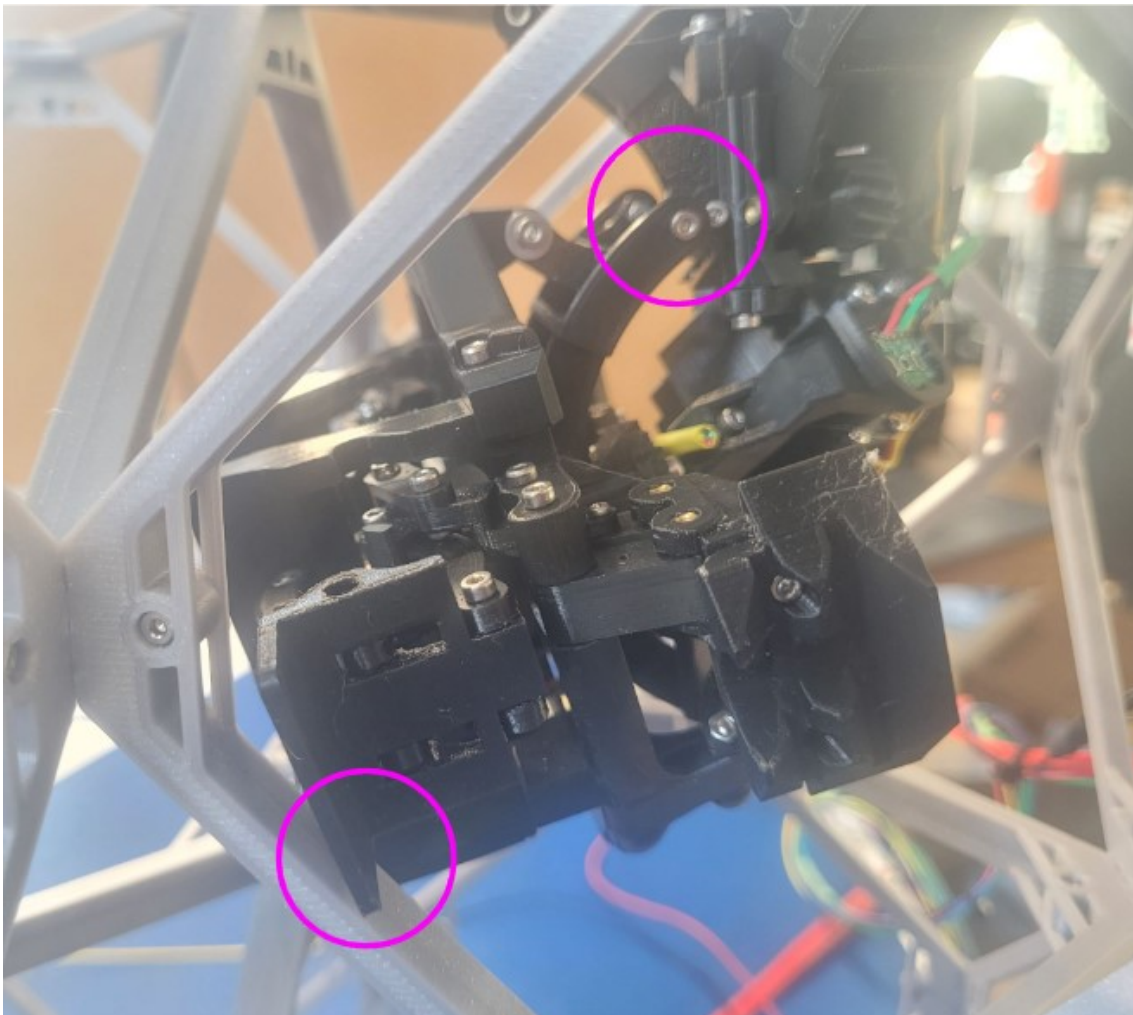


Figure 5.7: mMMIC-I clashing with the voxel face and the arm module when trying to turn inside the voxel structure. The pink circles marks the points of intersection.

Additionally, the hip servo will need to be continuous. This was not the case in this prototype since the servo range was long enough to evaluate the robot on one side. To determine the hip position, a Pololu #3081 encoder is used. This encoder unexpectedly only has a resolution of 90° . This is not enough for the functionality of the hip module and will need to be replaced.

Apart from the problems mentioned above, the evaluation also indicated three other

general improvement areas. First of all, are the tolerances inherited from the 3D printing? This has resulted in the mechanisms having a large wobble, and there is a dead zone that the servos can not control. This is also true for the worm gear. The gear solution experiences a lot of backlash that affects the mechanism's performance. Secondly, the wire routing needed to be improved. Currently, the temporary hot glue that holds the wires to the robot often comes off, and careful consideration has to be made so the wires do not interfere with the mechanisms during testing. Finally, alignment features. To ensure that the arm module can extend as intended, even when extending horizontally, extra passive alignment features need to be implemented. MMIC-I has alignment skis, but currently, mMMIC-I has none.

6

Discussion and Conclusion

The development of mMMIC-I went well. Even though the robot does not operate on the same level as MMIC-I, it has potential. The fidelity differences in the result were expected, given that MMIC-I took 3 years and a larger team to complete. With the same time and effort, it is plausible that mMMIC-I could perform on a similar level. This chapter reflects on the design and process of mMMIC-I and aims to answer the specification of the issue being investigated.

6.1 mMMIC-I Design

One of the outputs from this thesis project is the second iteration of mMMIC-I. This section reflects on the design, results, and future plans for mMMIC-I and its purpose.

6.1.1 Module Design

mMMIC-I is derived from MMIC-I and hence, has many similarities. The hip module, for example, is, at least by function and principle, the exact same. Many other concepts were investigated to replace the module, aiming to add the functionality of having an additional DOF there. The largest hurdle is space. Even using the smaller Pololu #3046 DC motor was considered both too weak and too large. Adding an additional DOF to this module does therefore not seem to be possible at this scale. The encoder for the hip module is the Pololu #3081 and does not provide enough resolution to support the function. The reason this particular encoder was chosen for the prototype was accessibility and size. It was by far the smallest encoder currently accessible in the lab. MMIC-I is using an off-the-shelf board with an AS5048B encoder that provides significantly higher resolution. This board was deemed too large for mMMIC-I, but the chip itself is very tiny and could be used on a smaller custom PCB.

The arm modules are also similar in many ways to MMIC-I's. The Sarrus linkage was kept but reduced to only three arms. By reducing the number of arms, the robot utilized fewer parts and became lighter. The drawback from this was the symmetry mismatch. More about that in section 6.3. The decision to use three arms was a free choice, but in hindsight, it would have been difficult to fit four of them. Using

three was a good decision. The arms are all driven by a worm gear. The entire gear system experienced a bit of backlash due to the 3D-printing nature. Even though RoyMech, 2020 was used to calculate all the dimensions, there are more aspects to gear design than what they have in their database. Optimally, off-the-shelf and tested gears should be used for this instead of custom 3D-printed ones.

The gripper and bolter modules are linked together. This is a new feature of the mMMIC-I. Many different concepts were explored for this mechanism. A four-bar link allowed both the grippers and the bolters to engage relatively straight and keep their orientation. This is especially important for the spring-loaded bolter bit that needs to engage as straight down from the top as possible. Linking the two mechanisms was found much easier if they acted parallel to each other. Another option would have been to make the gripper engage from another angle, utilizing more unoccupied space. No feasible solution was found that would solve this with the minimal complexity that was strived for. Each side only has two gripper-bolter combos, and they are actuated by one servo each. Having an individual servo for each mechanism was the solution for the flex and instability in the joint that was experienced in the first prototype. Even though one unified mechanism would have been better, the sacrifice in functionality was not worth it. If better tolerances were implemented, higher stability in the mechanism might be achieved, hence allowing only one actuator per side.

6.1.2 Result Interpretation

The testing showed that mMMIC-I was capable of operating in the expected way. The largest problem in the current prototype is the wobble in the mechanisms. They cause many errors with the alignment and locomotion. Better joints are not necessarily difficult to do as much as time-consuming. That is why they were not in focus during the first and second prototypes. Since the general design of the robot is now confirmed to work, spending more time tuning the tolerances to improve the performance can be worth it.

The only feature that did not succeed was the turning inside the voxel structure. Parts that clashed with each other prevented the turn. The margin for the robot not turning was really small. By adjusting the dimensions of some parts, it may be possible to allow the robot to turn without needing a complete redesign.

6.1.3 Next Steps

mMMIC-I is obviously not ready to fly yet. More iterations need to be made to fix the problems discovered in the evaluation. Appendix E serves as a type of backlog for the robot, containing things that should be addressed in the next iteration. Some of the mentioned things are not necessarily a problem with the current design, but rather a logical next step to take mMMIC-I to the next level. These include, for example, adding a space for the ESP32 on the robot and adding wire routing channels. Apart from the backlog, the bolters need to be addressed more thoroughly. That includes everything from making sure the spring-loaded bit can withstand the

torque output to deciding what bolting interface to use. Additionally, focusing on minimizing the wobble and backlash in the joints will go a long way to increase the overall performance.

Before the robot can be sent on a CubeSat mission, there are other components that need to be ready. A smaller version of SOLL-E (MOLL-E) needs to be developed to work together with mMMIC-I. Additionally, the half-sized voxels used now for reference and during testing are the simplest downsizing of a voxel possible. Eventually, a more thought-through design of the struts, fasteners, etc. should be made.

6.2 Simplification

One of the goals of mMMIC-I was to simplify the design of the robot. It is important now that the word simplification is very abstract. A robot could be deemed simple because it has less DOF. It could also be deemed simple based on the complexity of the mechanism, the number of parts or number of unique parts, energy efficiency, etc. There is currently not a standardized way to compare and evaluate the total simplicity of a robot. It becomes even more difficult to compare if, for example, a design change has reduced the number of actuators, but the trade-off is that the mechanism is much more complex. What would make the system more or less simple?

mMMIC-I has 11 actuators, compared to MMIC-I's 33. That is a significant decrease. The reduction of the actuator was important to clear the size requirements. The reduction to only two grippers and bolters on each side plays a big part in this, as well as the arm module only using one motor with a worm gear instead of four individual motors. This is a significant simplification of the design. Fewer actuators are cheaper and have lower control demands, and actuators are generally seen as points of weakness in a mechanical system (Yuan, 2023). Another reason for the actuator reduction is that the bolter and gripper engagement are joint in the same mechanism. Having the bolter align even if the bolt should be ignored is fine, and it saves one additional motor per bolter. The trade-off from this is that mechanical links need to connect the bolter and gripper, increasing the complexity of the mechanisms.

Having a more complicated mechanism results in more points of failure, more wear, and probably an increase in assembly time. In mMMIC-I's case, it also means more unique parts that need to be manufactured. Having fewer unique parts is also an aspect of simplicity. Optimally, the mechanism on mMMIC-I should be designed with as many common parts as possible. Having as few unique parts as possible makes it possible to focus on the quality and performance of those specific parts more. It is also cheaper and easier to keep spare parts. This was kept in the back of the head during the mMMIC-I development but was not something that was focused on. Keeping the same mechanism principles, the simplicity in regard to part uniqueness could be improved in future iterations.

The big question is: Are mMMIC-I simpler than MMIC-I? It is not an easy thing to answer. mMMIC-I is simpler in terms of the number of actuators, control states, and overall amount of custom parts. The mechanisms in mMMIC-I are more complex, utilizing more custom linkages and gears. Additionally, it is not determined if these changes affect the overall performance. mMMIC-I may not be an overall simpler robot. It explores new simplicity aspects that make it simpler in some ways. Based on those, the drawbacks can be evaluated to gain more knowledge about where the simplification efforts are worth implementing to the full-scale MMIC-I as well.

6.3 Symmetry Dilemma

One of the beauties of MMIC-I is the symmetry of the design. Having symmetry in the robot has all sorts of benefits. For example, manufacturing becomes cheaper, and spare parts become easier to keep. The assumption going into the development was to keep the symmetry aspect. While idea generating, TRIZ problem-solving principles came up, and one that came up a lot was asymmetry (Solid Creativity, 2025). This is what prompted me to stop thinking about the natural 90° rotational symmetry inherited from the voxels. The Sarrus linkage with three arms was born from these thoughts.

The grippers need to stay out of the Sarrus linkage arms' way. MMIC-I's four arms are spaced evenly around the four grippers. mMMIC-I, with only three arms on each side, does not keep the rotational symmetry. To allow maximum clearance between the two sides of arm modules while closing the Sarrus linkage in the first prototype, the two sides are mirrored (see Figure 6.1 A). For the grippers to align with the voxels, there are three scenarios indicated by the three pink circles on Figure 6.1 A. These are gripper-in-line-with-arm, gripper-opposite-to-arm, and gripper-next-to-arm. These are the same on both sides. After evaluating the prototype and deciding to use only two grippers, the symmetry between the two sides changed.

The two grippers need to attach to the voxel faces, one vertically and one horizontally. Playing around with the digital prototype, it showed that one side needed two cases of the gripper-next-to-arm scenario, while the other side needed to have the gripper-in-line-with-arm and gripper-opposite-to-arm scenarios. This was asymmetry showing, meaning that the two different sides needed to have different mechanisms to be able to stay clear of the Sarrus linkage. The solution landed on was to offset one Sarrus linkage by 30°, allowing the same combination of gripper-arm combos on both sides. Making both sides of the robot symmetric again (see Figure 6.1 B). However, this came with a drawback. The Sarrus linkage from the opposite sides will retract closer to the echoer. And the gripper will sit closer to an arm than previously. The solution that was supposed to "fix" the symmetry between the two sides also reduced the available space for the grippers and Sarrus linkage. The question to consider is if this adjustment is worth it. Currently, no parts intersect with each other during operations in mMMIC-I. The operational mechanisms work as intended. What could be investigated further is if not doing the symmetry "fix" and having the two sides different could result in a better solution.

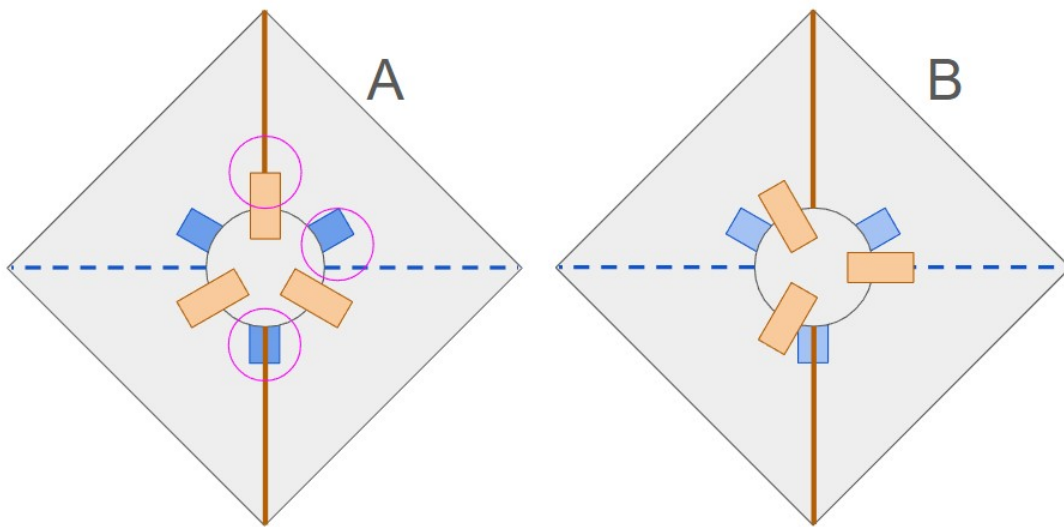


Figure 6.1: Display of how the grippers and Sarrus linkage arms line up in the prototypes. The orange and blue color coding indicates on what side of the hip module the components are. The boxes represent the arms of the Sarrus linkage, and the lines are where the grippers go. A: The first prototype. The pink circles show the different gripper-arm interactions that can happen. B: The latest prototype. One Sarrus linkage is offset by 30° .

One other solution to this symmetry dilemma that could keep more space for the grippers and eliminate the asymmetry of the grippers is by introducing asymmetry in the Sarrus linkage instead. The Sarrus linkage has 120° rotational symmetry in all investigated scenarios. If the spacing of the arms were different (asymmetrical), they may be able to create more space for grippers while still avoiding each other. Investigating this would also include making sure the new arms do not interfere with the other components or the voxel faces, as well as if the new asymmetric Sarrus linkage is stable enough for the robot to meet the requirements.

6.4 Methodology Evaluation

This section evaluates how the methodology has worked in the project. It discusses two main dilemmas that were tackled in regard to the process. In general, the methodology successfully produced expected levels of results. The planned approach was implemented and felt appropriate to be applied in this project context.

6.4.1 Agile Sprint Duration

In regard to the planning, the project followed the time schedule very well. In agile methodology, the sprint is often a pre-determined fixed time length (Rigby et al., 2016). In the beginning of each sprint, based on what is in the backlog, a few select items are chosen to be handled that would fit in that time frame. For

this project, the opposite was done. The length of the sprint (or prototype and evaluation phase) was determined based on what had to be done. This is a very different approach compared to the classical agile methodology. Having a fixed time frame makes it easier to estimate delivery times for milestones. It also pushes the project to keep tight deadlines in an effort to improve productivity. The reason for choosing the different approach was based on the knowledge gap of how long certain implementations would take and what data could be extracted from the evaluation. Setting a fixed sprint time would risk setting unrealistic goals. The initial prototype required significantly more time to get working, since all the modules needed to be addressed completely from scratch before the prototyping could begin. With a baseline of working mechanisms, minor improvements and redesigns of specific parts could be addressed.

Based on the efforts of this project, an altered view on sprint duration is not a bad thing. Here, the project benefited from this approach by being flexible while starting development. This does not mean that fixed agile sprint times are bad? Quite the opposite. After developing the first prototype and making sure all the general principles are working, a lot of uncertainty has been removed. This new knowledge allows for better estimations of the time required for new implementations. It also makes it easier and safer to keep fixed-time sprints, gaining the above-mentioned benefits for further iterations.

6.4.2 Prototyping vs Evaluating

One aspect during the development is the relationship between prototyping and evaluation. During prototyping, the prototypes are subconsciously evaluated. During this thesis project, especially during the first prototype, the feasibility of the overall concept became clearer even before the formal evaluation. In this case, the gripper and bolter module did not work. What could be considered is if the design should be altered the moment the problem was discovered or the prototyping should be followed through.

The benefits of assessing the problems directly are shortening the development time if less time is spent on unfeasible solutions. This is why a thorough concept evaluation should be done and smaller prototypes to demonstrate that the mechanical principle is feasible. This thesis project utilized the analytical prototypes as a first control and occasional 3D-printed versions. The first prototype did not have high fidelity, aiming for a more rapid approach to evaluate the first concepts faster.

The only changes that were implemented directly during prototyping were smaller changes such as tolerance or clearance measurements. This ensured that the concept had the best representation of feasibility. Noticing that a mechanism would not work as intended was ignored at the time. Obviously, it was noted down to be assessed later. The reason for proceeding with the concept that was found unfeasible during prototyping was to further investigate why it was not working as intended and if there are other problems that should be assessed in the next iteration. For the first prototype, it was noticed early on that the gripper and bolter mechanisms were

not working as intended. Having the full concept built allowed for a more detailed analysis of what went wrong. In this case, there was an imbalance in the links due to flex in the material. Seeing the problem in a system view makes it easier to consider if other parts could be altered to solve the problem or if the performance of other parts is affected negatively by the same problem. Even though more time will be spent on completing the faulty prototype, the knowledge gained from the system perspective will help further improve the next iteration. Additionally, while the actuation of the grippers and bolters did not work in the first prototype, other tests could still be conducted. Strength tests for the grippers could be made (if the grippers were moved into place manually). Also, even though the mechanism would change in the future, testing if the hip and arm modules are strong enough to lift the weight of the gripper and bolter modules is a good way to estimate if the servo strengths are adequate for a solution of this approximate size.

6.5 Concluding Thoughts

Scaling down MMIC-I is definitely possible. The main limiting factor for the downsizing is the size of the actuators. The motors for the bolters and the hip need to be strong enough to fulfill their purpose. The current prototype works for the most part, but it is not able to turn inside a voxel. This is a small flaw that can most plausibly be designed away.

Each part of mMMIC-I has been going through the product development methodology and investigating how they can be simplified. Simplification is a very abstract term. A robot can be simple in one way but more complex in another. The current prototype has reduced the number of actuators and total amount of parts while maintaining the same functionality. Other aspects to look at in the future include reducing the number of unique parts and merging mechanisms to reduce the number of control states.

Before going on a CubeSat mission, mMMIC-I needs to be further iterated. It is important that it can work with high reliability. Additionally, the other components of the mission (mini SOLL-E and the voxels) must be assessed too. Component-wise, the same off-the-shelf components can be used for the actual CubeSat mission.

Bibliography

- Arakelian, V. (2016). Gravity compensation in robotics. *Advanced Robotics*, 30(2), 79–96. <https://doi.org/10.1080/01691864.2015.1090334>
- Bickford, J. H., & Nassar, S. (1998). *Handbook of bolts and bolted joints*. Marcel Dekker, Inc.
- Burkhard, C. D., Stupl, J. M., Ticknor, R. L., & Yost, B. D. (2025). *State-of-the-art small spacecraft technology*. NASA.
- Cooper, R. G., & Sommer, A. F. (2016). The agile–stage-gate hybrid model: A promising new approach and a new research opportunity. *Journal of Product Innovation Management*, 33(5), 513–526. <https://doi.org/https://doi.org/10.1111/jpim.12314>
- Figliozzi, G. (2024). Robot team builds high-performance digital structure for nasa. <https://www.nasa.gov/general/robot-team-builds-high-performance-digital-structure-for-nasa/>
- Formoso, O., Trinh, G., Catanoso, D., Park, I.-W., Gregg, C., & Cheung, K. (2023). Mmic-i: A robotic platform for assembly integration and internal locomotion through mechanical meta-material structures. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 7303–7309. <https://doi.org/10.1109/ICRA48891.2023.10161263>
- Gregg, C. E., Catanoso, D., Formoso, O. I. B., Kostitsyna, I., Ochalek, M. E., Olatunde, T. J., Park, I. W., Sebastianelli, F. M., Taylor, E. M., Trinh, G. T., & Cheung, K. C. (2024). Ultralight, strong, and self-reprogrammable mechanical metamaterials. *Science Robotics*, 9(86), eadi2746. <https://doi.org/10.1126/scirobotics.adi2746>
- Hirshorn, S. R. (2007). *Nasa systems engineering handbook*. NASA.
- Manning, C. G. (2023). Technology readiness levels. <https://www.nasa.gov/directorates/somd/space-communications-navigation-program/technology-readiness-levels/>
- NASA. (2025). About nasa [Accessed: 2025-03-05]. <https://www.nasa.gov/about/>
- NASA Technical Standards Program. (2025). Nasa technical standards [Accessed: 2025-04-25]. <https://standards.nasa.gov/NASA-Technical-Standards>
- Park, I.-W., Catanoso, D., Formoso, O., Gregg, C., Ochalek, M., Olatunde, T., Sebastianelli, F., Spino, P., Taylor, E., Trinh, G., & Cheung, K. (2023). Soll-e: A module transport and placement robot for autonomous assembly of discrete lattice structures. *2023 IEEE/RSJ International Conference on Intel-*

- ligent Robots and Systems (IROS)*, 10736–10741. <https://doi.org/10.1109/IROS55552.2023.10341479>
- Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016). Embracing agile: How to master the process that's transforming management [Accessed: 2025-05-22]. *Harvard Business Review*. <https://hbr.org/2016/05/embracing-agile>
- RoyMech. (2020). Roymech [Accessed: 2025-04-11]. <https://roymech.org/>
- Schön, D. A. (1983). *The reflective practitioner book by donald schön*. Basic Books.
- Solid Creativity. (2025). Solving problems with triz matrix [Accessed: 2025-05-22]. https://www.triz40.com/TRIZ_GB.php
- Ulrich, K. T., Eppinger, S. D., & Yang, M. C. (2020). *Product design and development, seventh edition*. McGraw-Hill Education.
- Wheelwright, S. C., & Clark, K. B. (1993). *Revolutionizing product development*. Free Press.
- Yuan, Z. (2023). Current status and prospects of actuator in robotics. *Applied and Computational Engineering*, 11, 181–191. <https://doi.org/10.54254/2755-2721/11/20230232>

A

Specification of Requirements

A. Specification of Requirements

Requirement Specification						
Specification	Unit	Marginal value	Ideal value	How to test	Note	Importance (5 higher)
Locomote in voxels	-	-	-	Does it move in the voxels?	It should be able to operate inside standard voxel structure.	5
Bolt voxels	-	-	-	Can it bolt the voxels?	It should be able to bolt together the voxel using the standard bolting screw.	5
Unbolt voxels	-	-	-	Can it unbolt the voxels?	It should be able to unbolt the voxels using the standard bolting screw.	5
Operate in half-size voxels	millimeter	150 x 150 x 150	150 x 150 x 150	Test operation in 3D-printed half-size voxels.	It should be able to operate in the 1.5U CubeSat standard voxels.	5
Self powered	-	-	-	Is it self powered?	It should be powered by an onboard battery. Not a wire.	5
Self propelling	-	-	-	Is it self propelling?	It should be able to lift and propel itself in all directions.	5
Locomote multiple planes	-	-	-	How many planes can it move in?	It should be able to change direction and locomote in all 3 dimensions.	5
Confined in structure	-	-	-	Is it poking out?	It should only operate inside the voxel structure and not poke out.	4
Reliable locomotion	success percentage	93.5	100	Documenting multiple test of prototype.	Locomotion should be reliable.	4
Reliable bolting	success percentage	94.5	100	Documenting multiple test of prototype.	Bolting and unbolting should be reliable.	4
Cheap manufacturing	US dollar	(to be assessed)	(to be assessed)	BOM + assembly time. Price estimation.	It should be cheap to produce and assemble.	3
Simple mechanism	-	-	-	Investigate points of failure.	The mechanism should be as simple as possible	3
Operate in gravity	gravity (9.82 m/s ²)	0	up to 1	Test in Earth gravity.	It should be able to operate in many different gravitational fields.	2
Low weight	kilogram	1	0.5	Put on scale.	It should have as low mass as possible.	2
Few actuators	pices	33	5	Count the actuators.	It should use as few actuators as possible.	2
Space-qualified compatible	-	-	-	Can space-proof components be used?	Space-proof components should be able to be used with minimal to no adjustments.	1
Energy efficient	watt (during standard operation)	18	6	Voltage meter on the battery. The battery specifications must be known.	It should be as energy efficient and consume as little energy as possible while still completing the tasks.	1
Timely performance	-	-	-	Does the execution time match the prediction?	It should finish the tasks in the expected time frame. It can still be slow as long as the slowness is expected and accounted for.	1

B

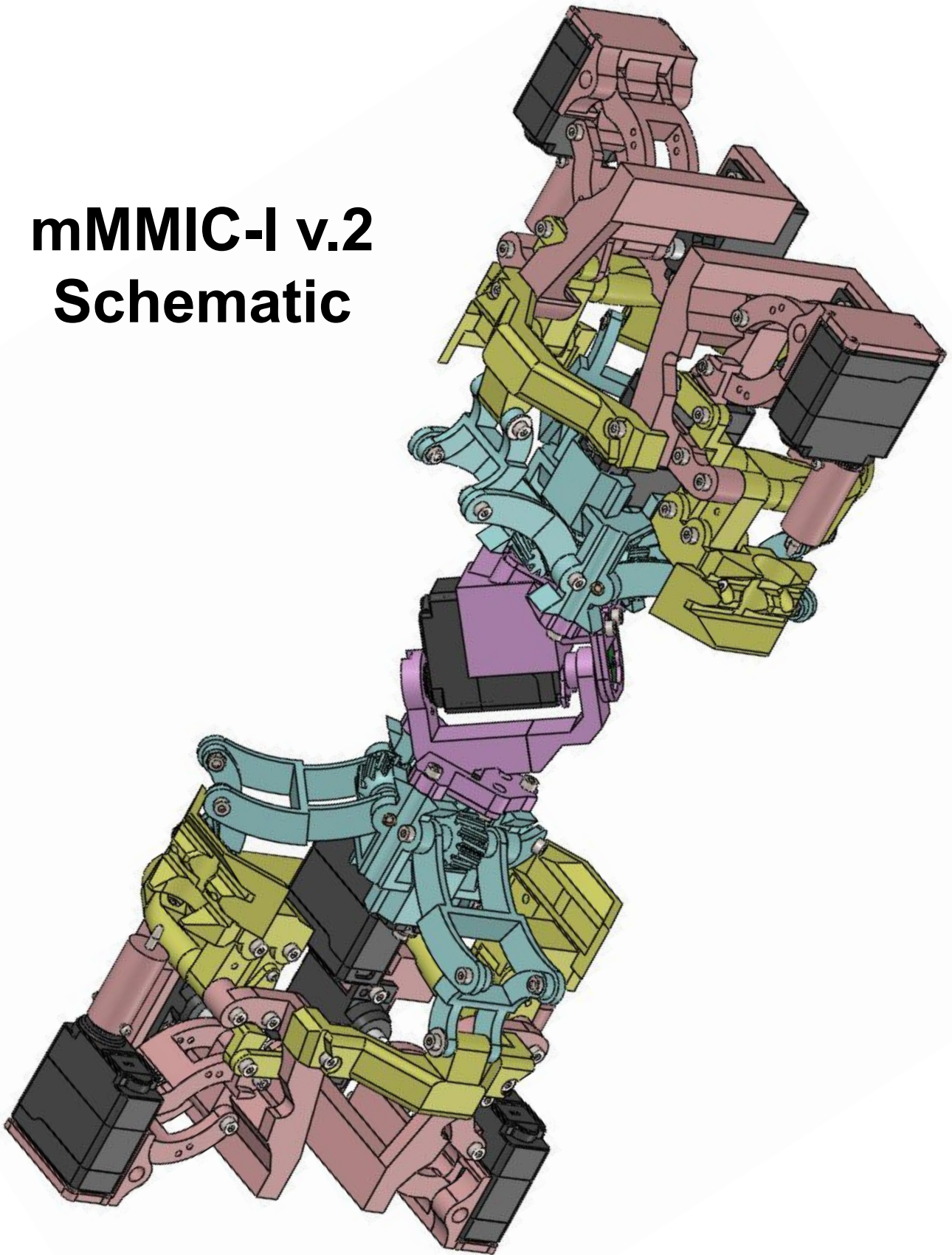
Evaluation Log 1

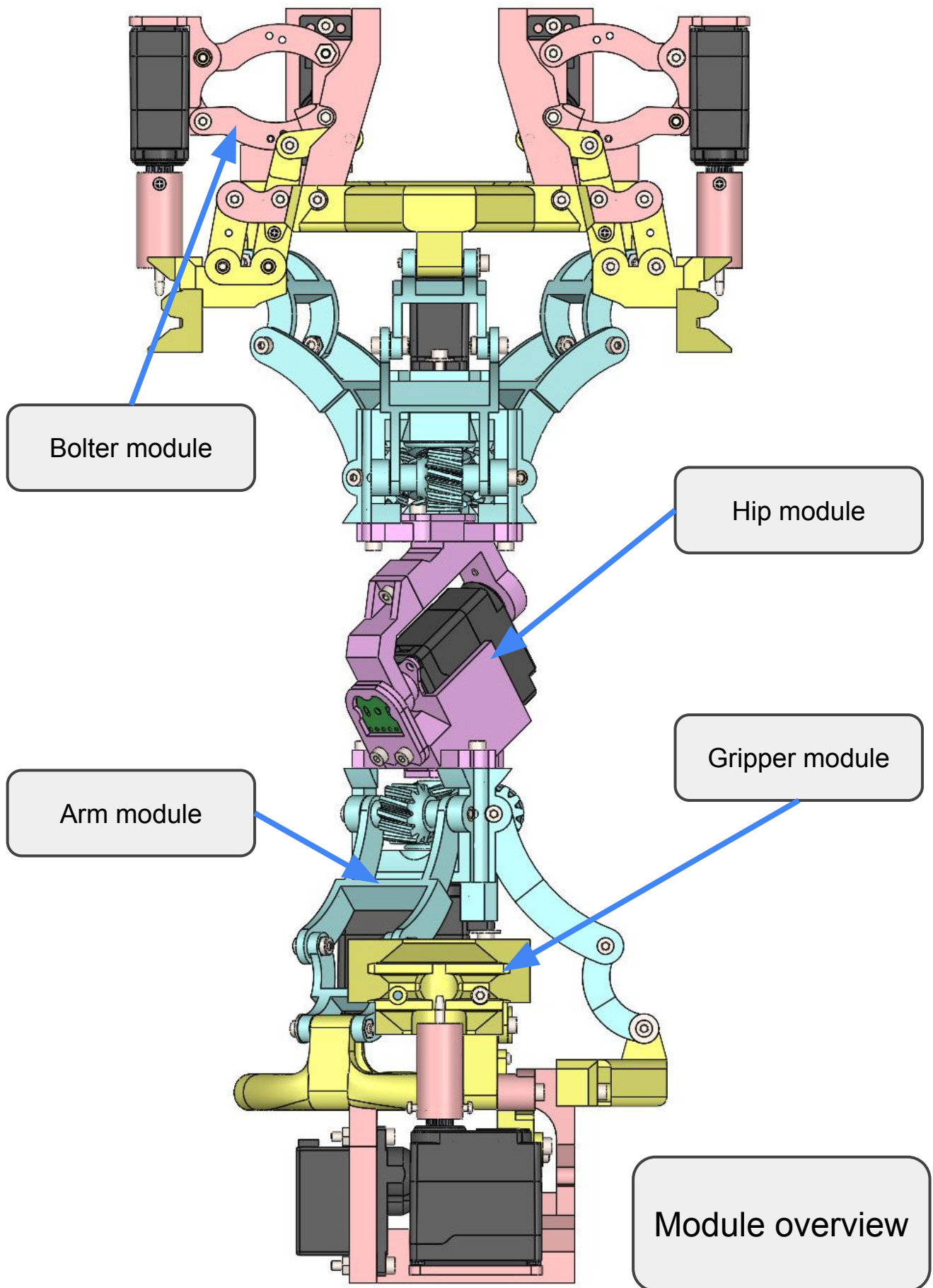
Evaluation Log 1		
Part/Module	Description	Value
Hip-module		
Motor holder / Motor axis	Interacts with the arms while contracted.	-
Motor holder	Mounting screw holes needed to be drilled out.	4 mm
Motor holder	Missing holes for wires from encoder.	-
Arm-module		
Gears	Some backlash in the gears.	-
Short arm	One of the arms interact with one of the grippers.	-
Long arm	Missing hard stop for the arm movement. Can lead to instability.	32° - 158°
Long arm	Two of them interacts with the voxel face on the side that is gripping.	-
Long arm / Short arm	Bolts fall out and locknuts are to big.	-
Gripper-module		
Gripper interface	Should hold both voxel faces to adjust final alignment before bolting.	-
Gripper-module	Mechanism is to large to turn in inside the voxel. It needs to be much shorter.	68 mm
Gripper link	Too much friction in the links.	-
Gripper link	These could be merged to reduce the total number of parts and hardware required.	-
Bolter-module		
Motor holder	Interacts with the voxel face when moving through it. It sticks out too much.	2 mm
Bolter-module	Two of the bolters are not technically necessary. Can be reduced.	-
Motor holder	Do not align well with the screw. It should be higher up and further back.	5 mm / 3 mm

C

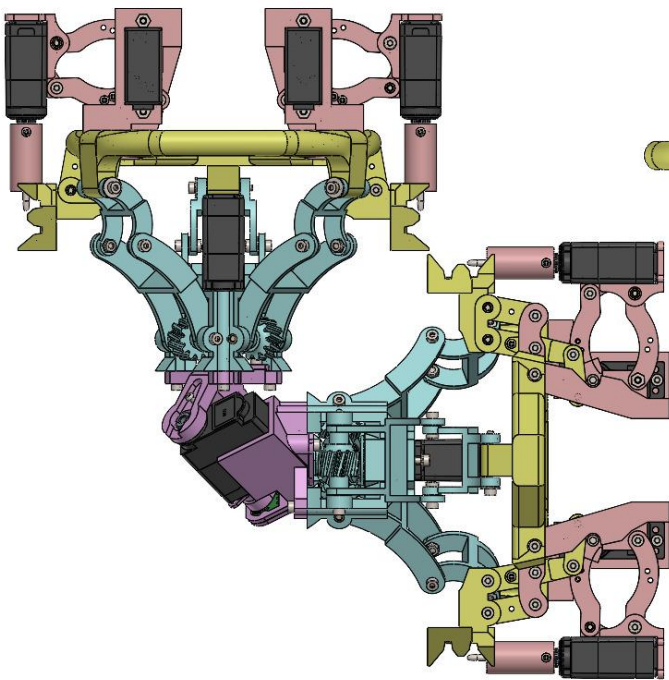
mMMIC-I Schematic

mMMIC-I v.2 Schematic

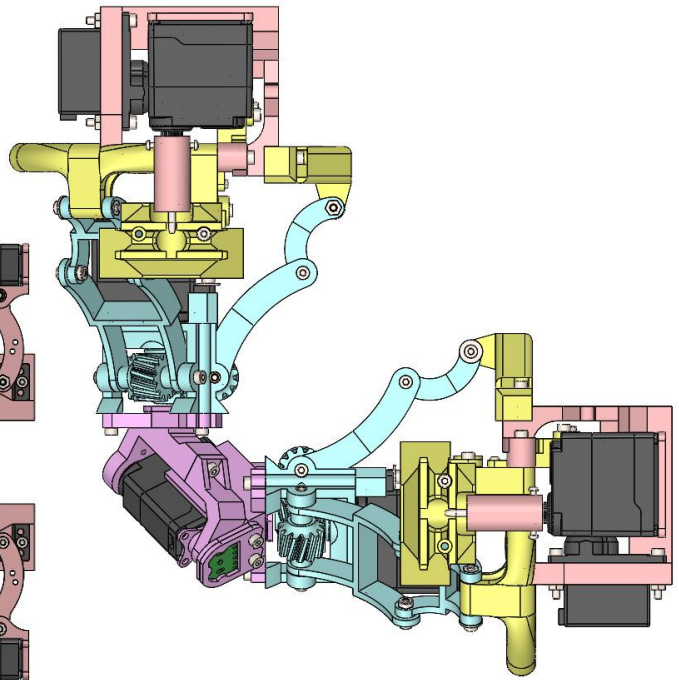




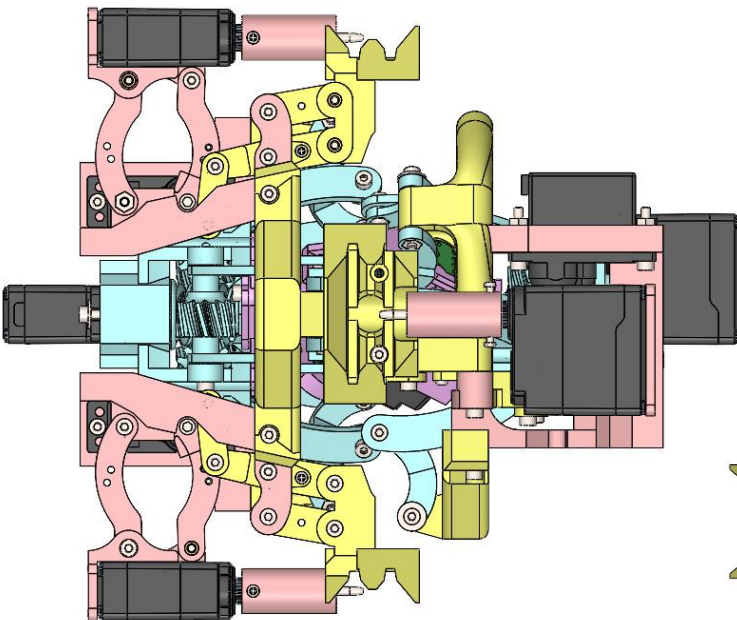
C. mMMIC-I Schematic



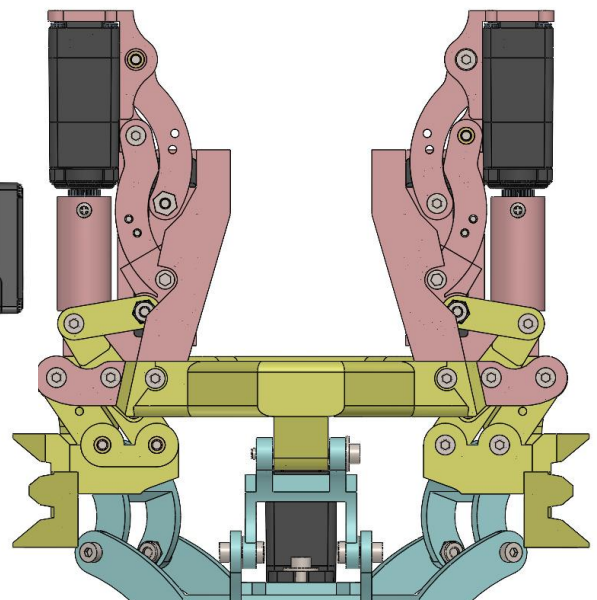
Bent 120°



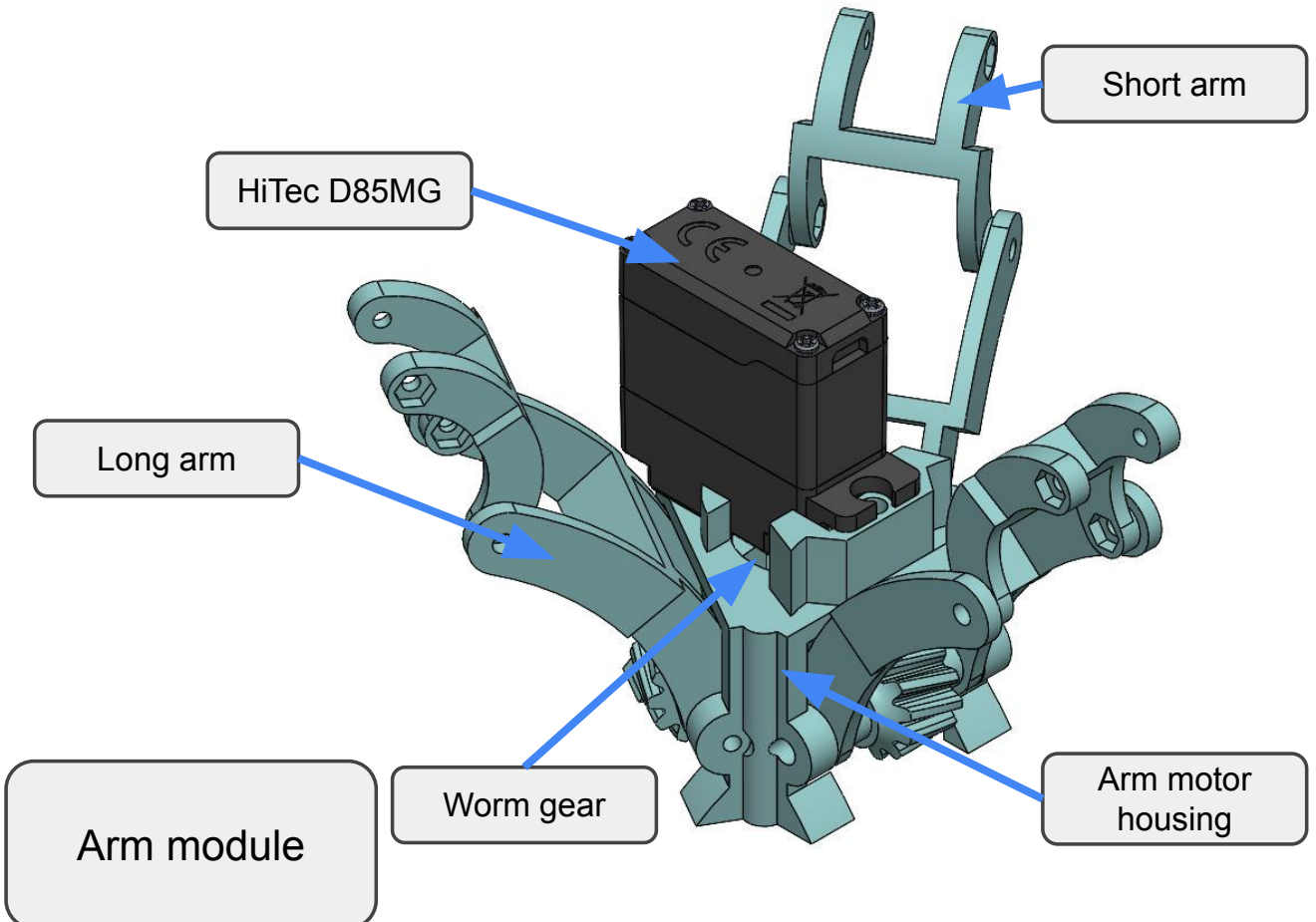
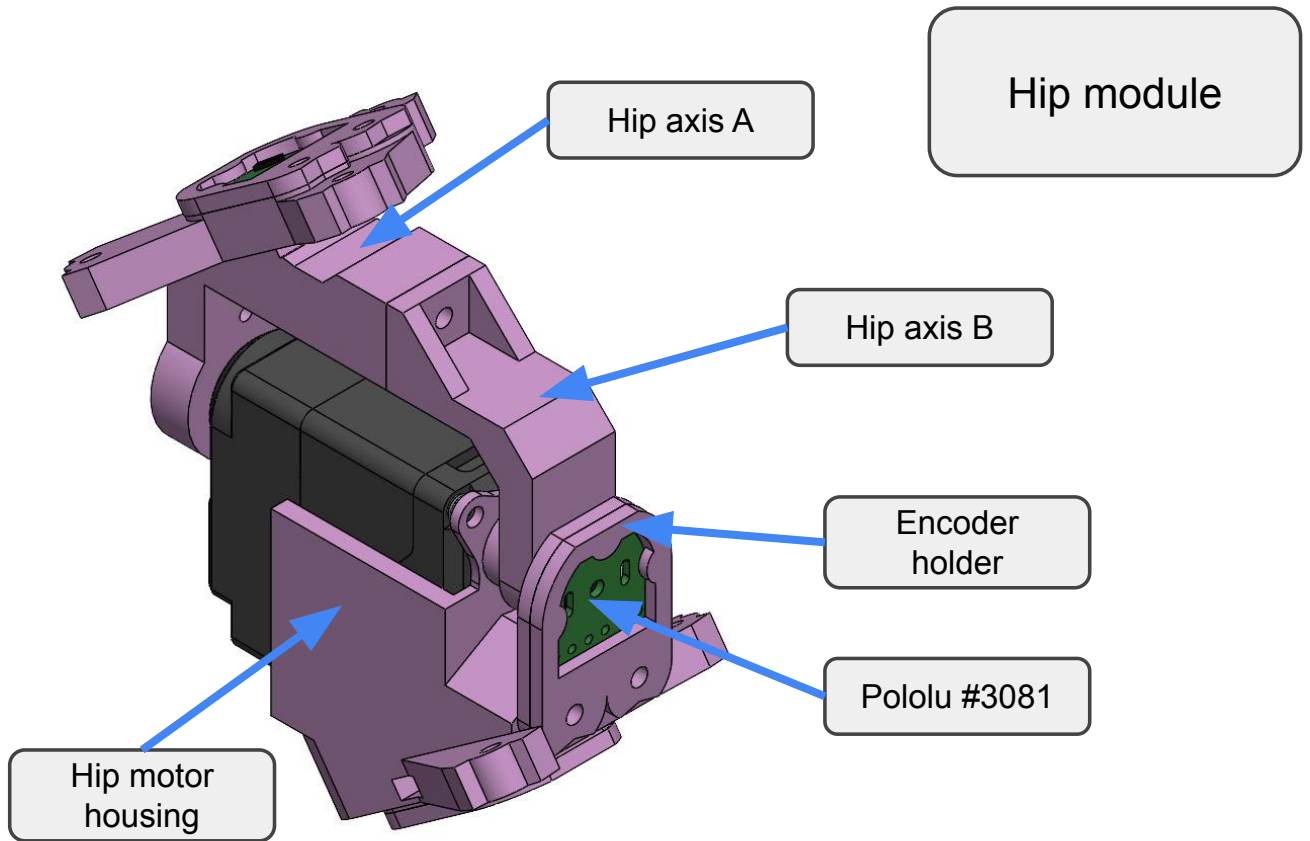
Bent -120°

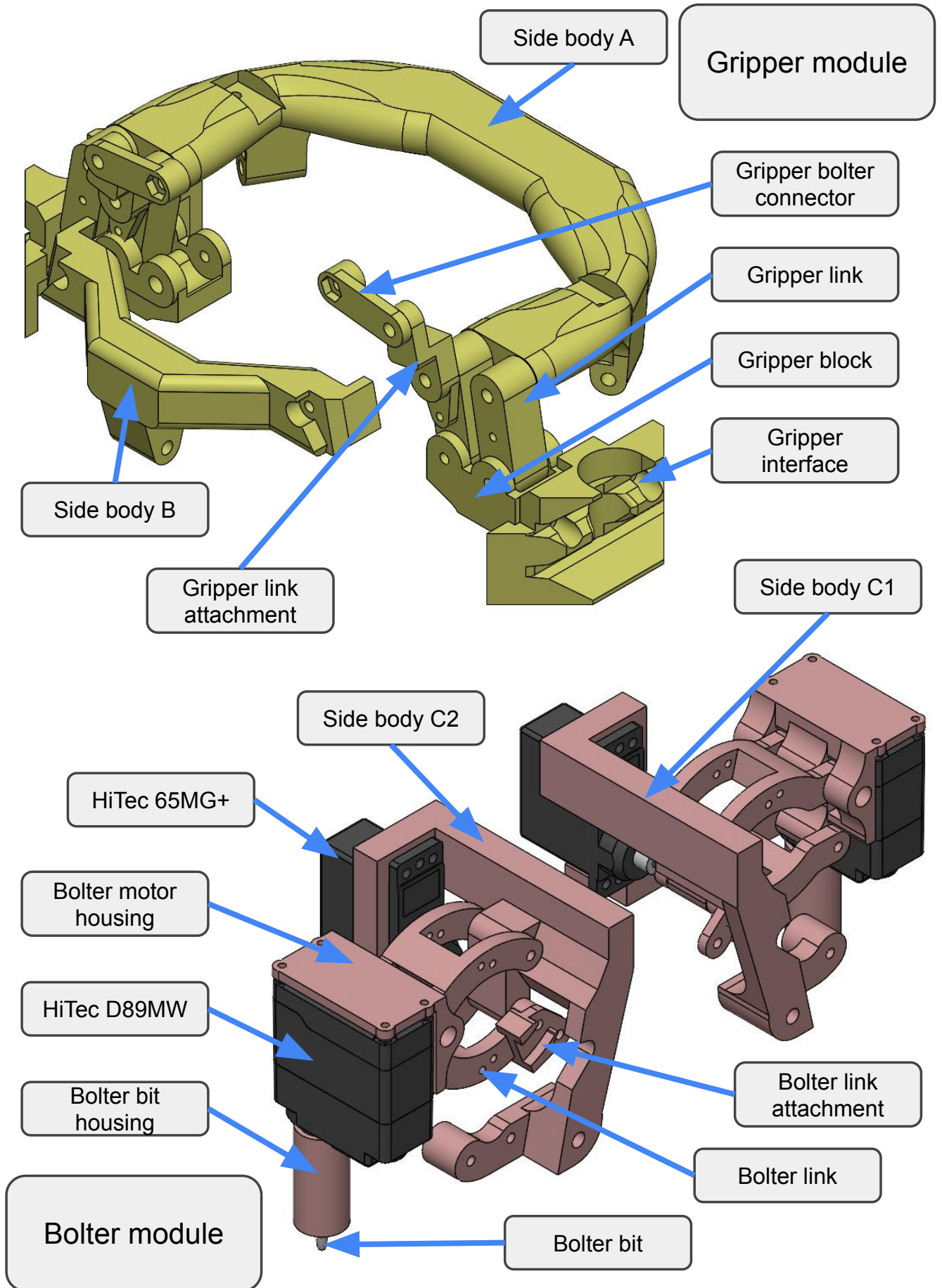


Contracted



Grippers and
Bolters Disengaged





D

mMMIC-I Bill of Materials

BOM v.2			
Part nr	Item	Material	Amount
Hip-module			
1	Hip motor holder	3D-print PLA	1
2	Hip axis A	3D-print PLA	1
3	Hip axis B	3D-print PLA	1
4	Encoder holder	3D-print PLA	4
23	HiTec D85MG	Servo	1
26	Pololu #3081 Encoder	Encoder	3
27	Pololu #3081 Magnet	Magnet	1
29	Threaded M2 insert	Brass	7
30	5 mm M2 bolt	Stainless steel	5
31	6 mm M2 bolt	Stainless steel	2
32	8 mm M2 bolt	Stainless steel	6
41	3/16 in 0-80 bolt	Stainless steel	1
Total			33
Arm-module			
5	Arm motor housing	3D-print PLA	2
6	Long arm	3D-print PLA	6
7	Short arm	3D-print PLA	6
8	Worm gear	3D-print PLA	2
23	HiTec D85MG	Servo	2
27	Pololu #3081 Magnet	Magnet	2
29	Threaded M2 insert	Brass	14
31	6 mm M2 bolt	Stainless steel	14
35	18 mm M2 bolt	Stainless steel	6
37	30 mm M2 bolt	Stainless steel	6
38	Nut M2	Stainless steel	6
39	Lock nut M2	Stainless steel	12
42	1/4 in 2-56 bolt	Stainless steel	2
44	Washer 2-56	Stainless steel	8
Total			88
Gripper-module			
9	Gripper interface	3D-print PLA	4
10	Gripper block	3D-print PLA	4
11	Gripper link	3D-print PLA	8
12	Gripper link attachment	3D-print PLA	4
13	Gripper bolter connector	3D-print PLA	4
14	Side body A	3D-print PLA	2
15	Side body B	3D-print PLA	2
29	Threaded M2 insert	Brass	20

30	5 mm M2 bolt	Stainless steel	4
31	6 mm M2 bolt	Stainless steel	8
32	8 mm M2 bolt	Stainless steel	4
33	12 mm M2 bolt	Stainless steel	8
36	20 mm M2 bolt	Stainless steel	8
39	Lock nut M2	Stainless steel	4
41	3/16 in 0-80 bolt	Stainless steel	4
Total			88
Bolter-module			
16	Side body C1	3D-print PLA	2
17	Side body C2	3D-print PLA	2
18	Bolter link	3D-print PLA	8
19	Bolter motor housing	3D-print PLA	4
20	Bolter link attachment	3D-print PLA	4
21	Bolter bit	3D-print PLA	4
22	Bolter bit housing	3D-print PLA	4
24	HiTec D89MW	Servo	4
25	HiTec 65MG+	Servo	4
28	Spring	Spring	4
29	Threaded M2 insert	Brass	16
32	8 mm M2 bolt	Stainless steel	4
33	12 mm M2 bolt	Stainless steel	8
34	16 mm M2 bolt	Stainless steel	12
38	Nut M2	Stainless steel	12
39	Lock nut M2	Stainless steel	4
40	1/8 in 0-80 bolt	Stainless steel	8
41	3/16 in 0-80 bolt	Stainless steel	8
43	7/8 in 2-56 bolt	Stainless steel	4
Total			116
Total			325
3D-print PLA			(79)
Actuators			(11)
Hardware (nuts and bolts)			(225)
Miscellaneous			(10)

E

Evaluation Log 2

Evaluation Log 2		
Part/Module	Description	Value
All parts		
Axis holes	Tolerances for holes are too bad. Result in too much dead wiggle in the mechanisms.	-
Alignment skis	Add alignment skis to the robot to make sure it can pass through each voxel face.	-
Wire routing	Improve wire routing possibility. Merge ground and power as soon as possible to minimize the wire volume.	-
All parts	Reduce the number of unique custom parts.	-
Hip-module		
Hip encoder	Current encoders can only position with 90° accuracy. That is not enough.	New
Arm-module		
Long arm	Interfere with long arm on other side while turning.	-
Gears	Gears are still slipping. New design or manufacturing method.	New
Gripper-module		
Side body A / Side body B	Make space and attachment for battery and ESP32.	-
Bolter-module		
Bolter motor holder	Interfere with the voxel while turning.	-
Bolter bit housing	Too long, interfere with voxel too soon.	2 mm

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden

www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY