AFRL-AFOSR-JP-TR-2024-0024

Variable Topology Truss for Robotic Humanitarian Missions

Yim, Mark
TRUSTEES OF THE UNIVERSITY OF PENNSYLVANIA
3451 WALNUT ST
PHILADELPHIA, PA,
US

12/12/2023
Final Technical Report

Air Force Research Laboratory
Air Force Office of Scientific Research
Asian Office of Aerospace Research and Development
Unit 45002, APO AP 96338-5002

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE 20231212	**2. REPORT TYPE** Final	**3. DATES COVERED** **START DATE** 20170421	**END DATE** 20210120

4. TITLE AND SUBTITLE
Variable Topology Truss for Robotic Humanitarian Missions

5a. CONTRACT NUMBER	**5b. GRANT NUMBER** FA2386-17-1-4656	**5c. PROGRAM ELEMENT NUMBER** 61102F
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Mark Yim

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
TRUSTEES OF THE UNIVERSITY OF PENNSYLVANIA
3451 WALNUT ST
PHILADELPHIA, PA
US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR IOA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-JP-TR-2024-0024

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This is the final report of Variable Topology Truss (VTT) for Robotic Humanitarian Missions. The objective of this project was to develop a self-reconfigurable robotic truss system suitable for disaster response. A VTT robot is a truss where each edge is a linear actuator module. The robot can adapt its shape both by changing the length of its edges to change its geometry, and additionally by rearranging their connections to change the truss topology. In the first year of the project, together with our Seoul National University partners, we developed the conceptual framework of the VTT robot system. We developed techniques for analyzing the reconfiguration capabilities of the system and for selecting useful truss topologies. For the VTT hardware, we developed basic prototypes of the linear actuator component and reconfigurable joint component. For software, we developed basic collision checking and path planning implementations, and laid the groundwork for software tools for planning, control, and visualization of the robot.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE
U	U	U

17. LIMITATION OF ABSTRACT
SAR

18. NUMBER OF PAGES
74

19a. NAME OF RESPONSIBLE PERSON
JERMONT CHEN

19b. PHONE NUMBER *(Include area code)*
315-227-7003

Standard Form 298 (Rev.5/2020)
Prescribed by ANSI Std. Z39.18

Variable Topology Truss for Robotic Humanitarian Missions,

Final Report

Grant FA2386-17-1-4656,
January 20, 2021

PI: Mark Yim
Researchers: Jangho Bae, Alexander Spinos, Devin Carroll, Chao Liu
University of Pennsylvania

Executive Summary:

This is the final report of Variable Topology Truss (VTT) for Robotic Humanitarian Missions. The objective of this project was to develop a self-reconfigurable robotic truss system suitable for disaster response. A VTT robot is a truss where each edge is a linear actuator module. The robot can adapt its shape both by changing the length of its edges to change its geometry, and additionally by rearranging their connections to change the truss topology.

In the first year of the project, together with our Seoul National University partners, we developed the conceptual framework of the VTT robot system. We developed techniques for analyzing the reconfiguration capabilities of the system and for selecting useful truss topologies. For the VTT hardware, we developed basic prototypes of the linear actuator component and reconfigurable joint component. For software, we developed basic collision checking and path planning implementations, and laid the groundwork for software tools for planning, control, and visualization of the robot.

In the second year of the project, we developed a more mature prototype of the VTT edge module, with all of the major subcomponents. This includes the spiral zipper actuator, tension cable system, and reconfigurable VTT joints. We developed a ROS package to manage VTT planning and control, as well as a grid-space based planning algorithm to achieve fast path planning for the robot.

Since then, we have made significant progress in demonstrating the feasibility of the system. To parallelize our progress, we decided to first pursue two hardware prototypes: a locomotion prototype with a manually reconfigurable topology, and a reconfiguration prototype that would explore the details of the topological self-reconfiguration process. The locomotion prototype would have stronger, but simpler joints, allowing us to improve the spiral zipper actuator and test our locomotion algorithms without worrying about damaging the more delicate reconfiguration mechanisms. Once both aspects of the design were proven, it would be relatively straightforward to combine them. Unfortunately, with access to our lab limited due to COVID-19, we realized it would be impossible to finish both prototypes before the project deadline. We

decided to cut the reconfiguration prototype and instead focus our efforts entirely on completing the locomotion hardware prototype and developing more sophisticated planning and locomotion algorithms.

In the first section of this report, we review the VTT concept. We give a general overview of the components of the robot and describe how the reconfiguration functionality works.

In the second section, we describe the development of the VTT hardware. The major subcomponents of the hardware are the spiral zipper actuator, tension cable system, and both reconfigurable and non-reconfigurable truss joints. For each subcomponent, we give a brief history of the design and discuss the final prototypes in more detail. This year, we built twelve copies of our first-generation prototype edge module. Most of our experiments were performed with these first-generation edge modules. More recently, we have developed a second-generation prototype edge module that addresses the shortcomings of the first-generation prototype.

In the third section, we review the software architecture of the system. This includes the embedded software running on each module, as well as the ROS control code running on the central computer. We also discuss the new algorithms we developed. These fall into three categories: collision-unaware high-level topology planning, collision-aware reconfiguration planning, and locomotion planning that considers all relevant constraints. The high-level topology planning explores what arrangements of truss members are possible, and is used to select the topologies to achieve certain tasks. For locomotion planning, we developed a polygon-based random tree search algorithm. This algorithm finds desired support polygons to reach the goal with the random tree search algorithm. After that, VTT follows desired support polygons with non impact rolling locomotion, which prevents damages from tipping over.

In the final section, we describe the truss control tests and other experimental results. We successfully demonstrated the hardware in various tetrahedral configurations, showing good trajectory-tracking performance. We also used twelve modules to construct an octahedral truss, which we used to test our locomotion algorithms. However, we experienced repeated hardware failure due to the low bending strength of the modules. To address this issue, we developed a second-generation edge module, rebuilt the truss, and repeated the experiments.

1 Variable Topology Truss Overview

In many applications, trusses are ideal due to their high structural efficiency, as they consist of members which are only in pure tension or compression, resulting in reduced maximum stress. The class of robots commonly known as variable geometry trusses (VGTs) can be obtained by replacing some or all of the members in a truss with linear actuators. There are many applications for this type of robot, including parallel manipulators, long chain actuators, collapsible structures for space applications, and locomotion platforms.

During disaster emergency operations, a mechanism that enters a damaged structure and quickly shores the structure as first responders search for victims could save lives by finding victims in the golden hours, and keeping the rescuers safe from harm. If the system is quickly extractable and reusable as well, multiple structures could be explored. This could be a game changer in the search and rescue field for large scale disasters.

A robot that could conform its shape to be large enough and strong enough to shore the structure, and then extract itself and shore another structure, would enable much safer, faster and cost effective search and rescue.

Trusses are inherently materially efficient, lightweight, and spacious, yet strong; they are the standard framework for bridges, cranes, roof tops, anywhere large strength and low weight are required. A reconfigurable truss that could change its shape and topology, a Variable Topology Truss (VTT) would be an ideal approach to solve the above scenario

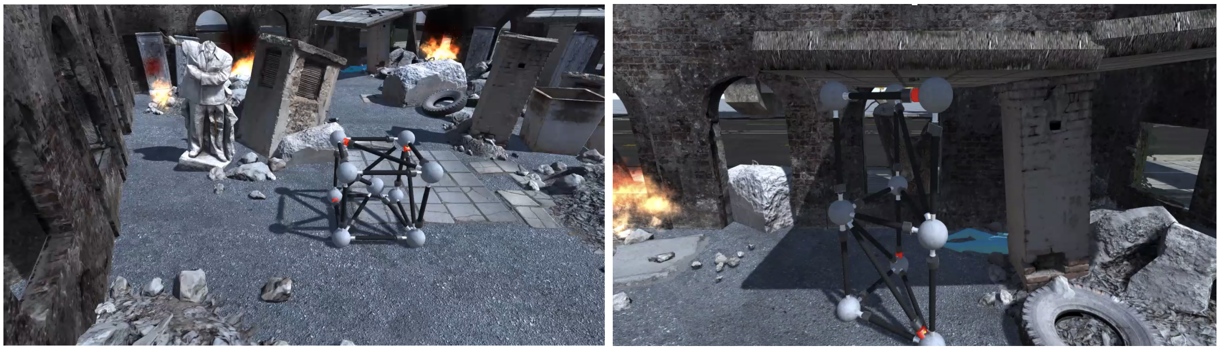


Figure 1.1: A concept simulation (from collaborators at SNU) that shows a VTT system in a disaster scenario entering a hotzone via a tumbling gait (a) and shoring a partially collapsed structure (b).

A variable topology truss (VTT) starts from the same truss framework as a VGT: linear actuators form the members (beam elements in the truss) and passive spherical joints connect the members at the nodes in the truss. However, the truss has the additional capability to self-reconfigure, changing its topology by merging or splitting nodes. That is, two separate nodes in the truss can dock to form one single node which connects all of the involved members. Similarly, a single node with a sufficient amount of members can undock into a pair of nodes. In this way, a VTT can be thought of as a chain type self-reconfigurable robot consisting only of linearly actuated elements and freely movable spherical joints.

While VGTs only have control over the shape or geometry of the truss, a VTT can additionally change the topology of the underlying truss. This brings benefits typically associated with reconfigurable robots; the VTT can select the topology which is most suited to the task at hand. For example, a single VTT may have the capability to self-reconfigure to suit many applications: a collapsible topology for storage, a multi-limbed topology for locomotion, or a long thin topology for an extended antenna application. This task flexibility is not possible with the single fixed topology of a VGT.

The additional reconfigurability of a VTT could have applications in space missions as a lightweight adaptable structure, or as versatile structural reinforcement to assist first responders during disaster scenarios.

2 VTT Hardware

In this section, we describe our progress and results with the development of the VTT hardware. First, we give a high level description of the components of a VTT edge module and clarify the difference between our prototypes that have a reconfigurable topology and our prototypes that have a fixed topology. Then, we discuss each subcomponent in detail.

2.1 Edge Module Overview

A VTT edge module is the fundamental unit of a variable topology truss. It corresponds to one member in the truss, and contains all of the necessary joints to connect to other edge modules in the truss. It is fully self-contained, with battery power, computation, and communications modules all incorporated into the module. Each module consists of a high extension ratio linear actuator with two special spherical joints, one on each end. A complete edge module can be seen in Figure 2.1 below.

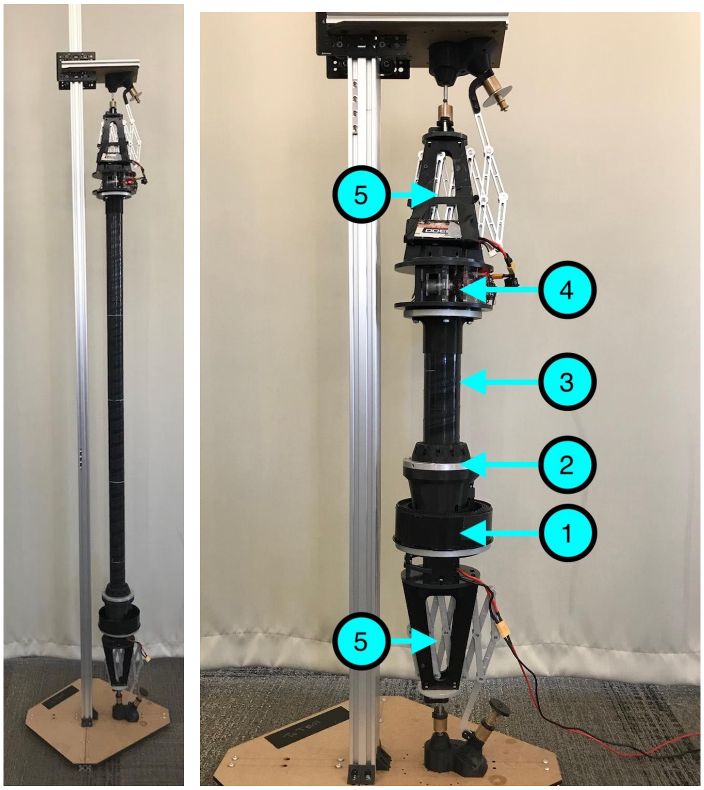


Figure 2.1: [Left], *a fully extended VTT edge module.* [Right], *a shorter configuration with the subcomponents marked: the band management system (1), the slider and support ring (2), the spiral zipper band column (3), the tension mechanism (4), and the VTT joints (5).*

The core of the module is a high-extension ratio linear actuator that can support large compressive and tensile loads. This length-changing element is the fundamental way that the

VTT changes its shape and moves around. The linear actuator can be broken down into a *spiral zipper actuator* and a *tension cable system*. The spiral zipper actuator is the main structural element and is strong in compressive loading conditions. The tension cable system augments the spiral zipper actuator by applying a controllable amount of tension to a cable that runs down the middle of the band. This ensures that the spiral zipper actuator always experiences the favorable condition of compressive loading, even when the module overall is experiencing tensile loading. These two subsystems are placed on either end of the edge module to produce a more even mass distribution. Each subsystem needs its own actuator and control system, but it is difficult to electrically connect these two subsystems across the linear actuator element. For this reason, each edge module has two separate batteries and microcontrollers, one on each end.

The spiral zipper actuator can be further broken down into subcomponents. The material making up the column of the actuator is the *spiral zipper band*, which is stored in a spool in the *band management system.* As the spiral zipper band spins, the *slider* is the guiding component that meshes the band teeth, transforming it from its spooled-up state to a rigid column state and vice versa. A *drive mechanism* applies torque to the band column, causing it to spin. In its current embodiment, the drive mechanism is a friction wheel located inside the band column. Outside the band column, the *support ring* provides a balancing force to the outward pressure from the friction wheel. It also helps support the band column in bending.

Aside from the core of the module, the other important part is the pair of VTT joints on either end. The VTT concept requires reconfigurable spherical joints, so we have developed several prototypes of these. However, for many of our tests, we are focused on developing an effective main actuator and controlling the geometric reconfiguration of a fixed topology. For these tests, we use non-self-reconfigurable joints that are more stiff and reliable than our reconfigurable joint prototypes.

Figure 2.1 shows the state of the edge module at the end of last year's report. In this reporting period, we produced two, more mature generations of the edge module prototype, which are shown in Figure 2.2. Compared to last year's prototype, the core of the first-generation edge module that we produced this year uses the same ABS spiral zipper band, a reduced-weight band management system and slider, and an improved tension cable system. The core of the second-generation edge module uses a wear-resistant nylon spiral zipper band, low-friction slider, and the same tension cable system as the first-generation module.

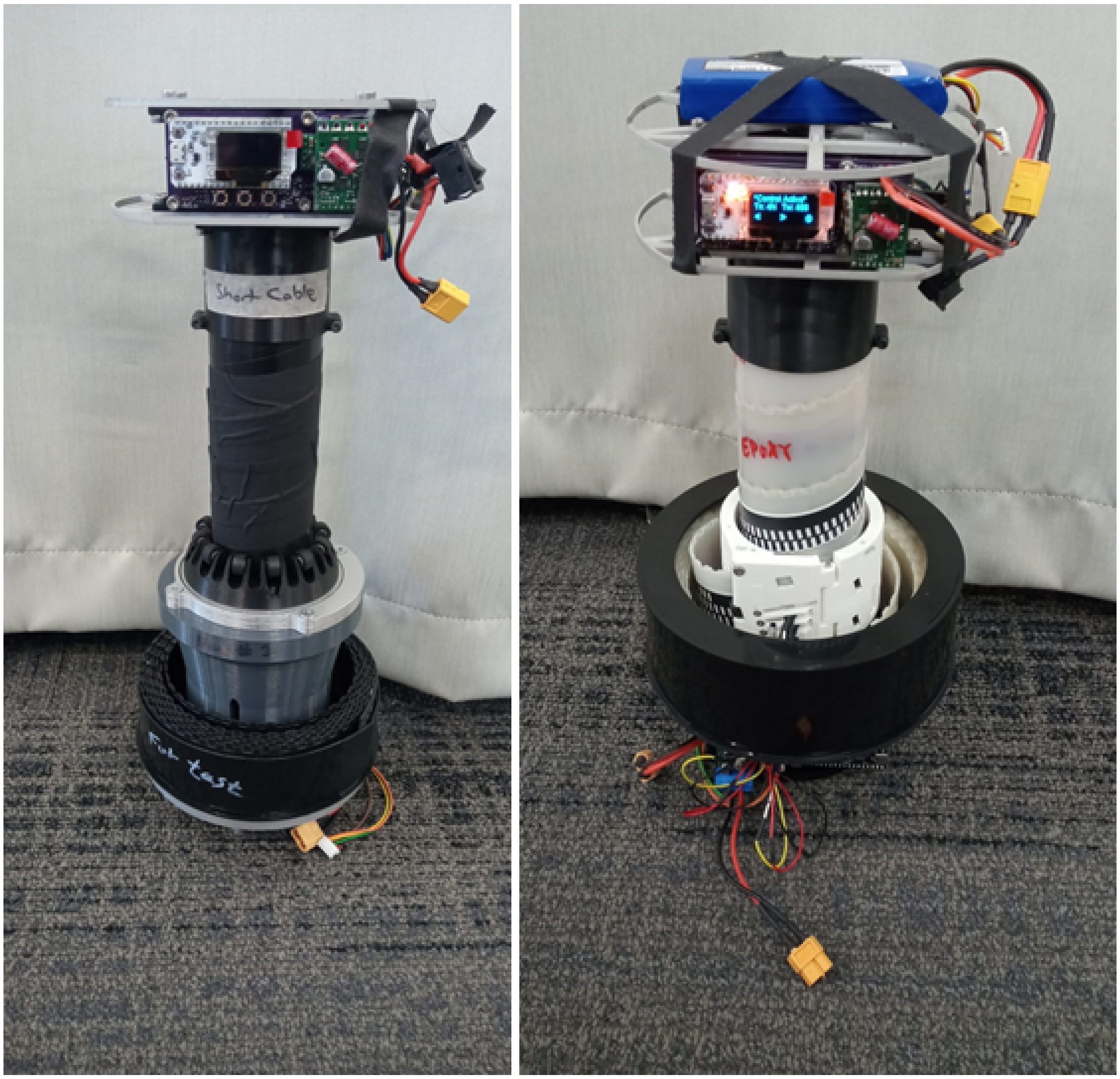


Figure 2.2: The updated edge module core. It uses the same tension cable design, but the band and friction drive have been updated.

2.2 Spiral Zipper Band

The spiral zipper band forms the main structural element of the truss members. Although the initial concept predates the start of this project, our efforts have made it better.

One of the proposed use cases for the VTT is to shore buildings - requiring forces up to 10,000 N. As such, the spiral zipper must have a load capacity at or near this value. In the first year of the project we conduct compressive strength tests on 4.5 inch diameter, 12 to 80 inch spiral zipper columns. We use a pulley based attachment jig connected to the MTS machine to test these longer members (Fig. 2.3).

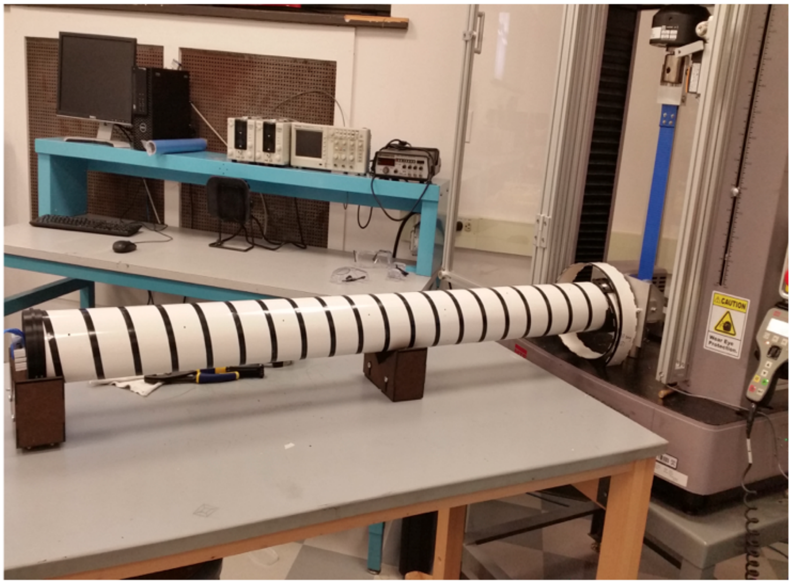


Figure 2.3: The spiral zipper in the pulley based attachment jig connected to a material stress-strain testing (MTS) machine.

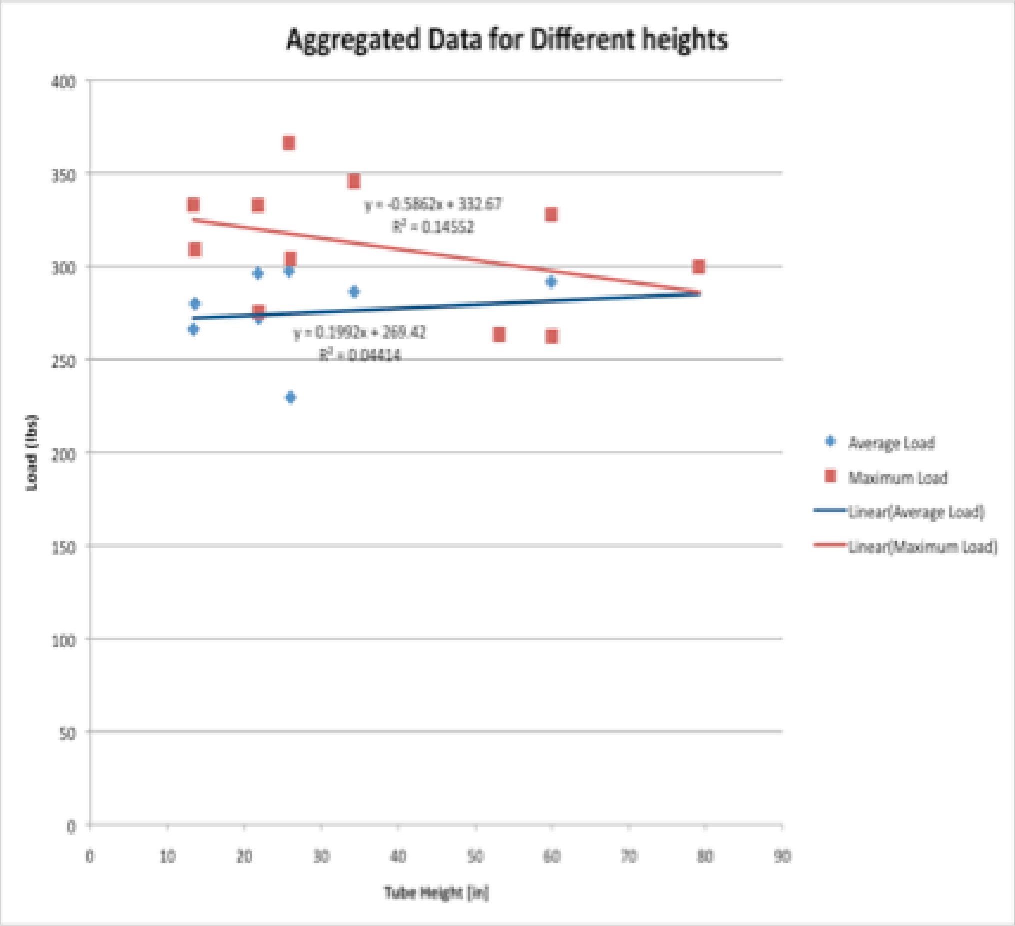


Figure 2.4**:** The test results of the MTS testing.

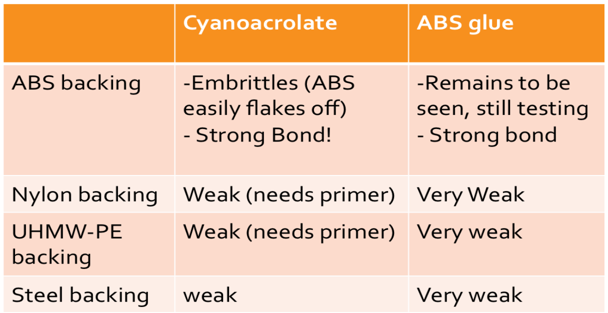
Fig. 2.4 presents the results from the tests. The data shows the column supports ~300 lbs regardless of the length of the column. In order to reach the load capacities required for structural support we investigate 3 approaches to increase the column strength without significantly increasing the cost of the member:

1. Change the backing material (*material*)
2. Improve the bond between the band and backing (*adhesives*)
3. Change the band material (*material*)

Changing the backing material (1) will result in a different bond strength (2) for the same adhesive. Critical considerations for material change include:

1. Resistance to stress cracking
2. Fatigue resistance
3. Adhesive bond strength
4. Low friction
5. Low cost

Table 2.1 Adhesive Backing Summary



	Cyanoacrolate	**ABS glue**
ABS backing	-Embrittles (ABS easily flakes off) - Strong Bond!	-Remains to be seen, still testing - Strong bond
Nylon backing	Weak (needs primer)	Very Weak
UHMW-PE backing	Weak (needs primer)	Very weak
Steel backing	weak	Very weak

Table 2.1 presents qualitative results of the strength of different adhesive bonds betweens the band and backing. An ABS backing on a steel band also failed at approximately 300 lbs (not shown). In all failures, the backing ripped or buckled (Fig. 2.5). Obviously then we must find solutions which create a high strength bond between the backing and the band. One method investigated in 2017 and 2018 was thermo-forming the backing and the band from one sheet (Fig. 2.6).

Figure 2.5: Compression test results showing backing failures with plastic and metal bands and backing.

Figure 2.6: Thermoforming the band using heated dies (left) and the resultant structure (right).

A PCB oven heats the metal dies which are clamped onto the band in order to form the teeth. The process is time intensive and because we are forming with flat, heated plates, the formed ledge vanishes when the band is wound to the desired diameter. We briefly experimented with cylindrical dies to solve the problems with the flat dies and reduce the time investment to make a band but were unable to uniformly heat them when forming the band.

This year we experimented with using dies to form metal bands with the eventual goal of replacing the plastic bands with stronger and stiffer metal bands. The band material can be thinner than ABS (0.005” thickness) for the same strength giving the zippers a greater strength to weight ratio. We test with 3D printed ABS dies (Fig. 2.7) as well as machined dies to form the band. Interestingly, the plastic dies can form the bands reasonably well though they do not lock together as nicely as those formed with the machined die (Fig. 2.8).

Figure 2.7: An example of one of the ABS cold forming dies. The plastic dies would begin to wear after two or three uses on 0.005” thick bronze.

Figure 2.8: A section of 0.005” bronze formed with the ABS dies

One of the original plans for the edge module was to place the slider and band management system in the center of the edge module to minimize the size of the node-to-node connections. We test the deflection of the 2.25” diameter arm in this configuration for feasibility (Fig. 2.9).

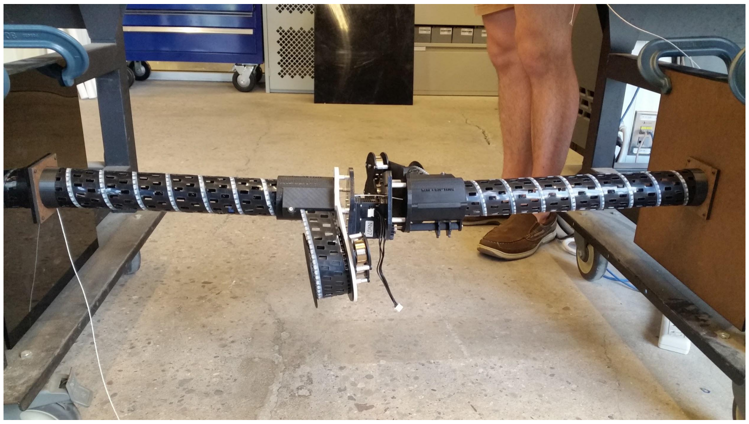


Figure 2.9: Testing a simply supported edge module made from two spiral zippers with the sliders and band management systems located in the center of the module.

Band Reliability and Robustness

The demo for IROS (Fig. 2.10) revealed a number of faults with the previous band design. A 1” diameter column puts too much stress on both the band and backing material causing them to crack (Fig. 2.11). To reduce the stress on the material to a value well below the material yield stress we chose a 2.25” diameter column (Fig. 2.12). This is later increased to a 2.5” diameter column when we switch the band material to nylon.

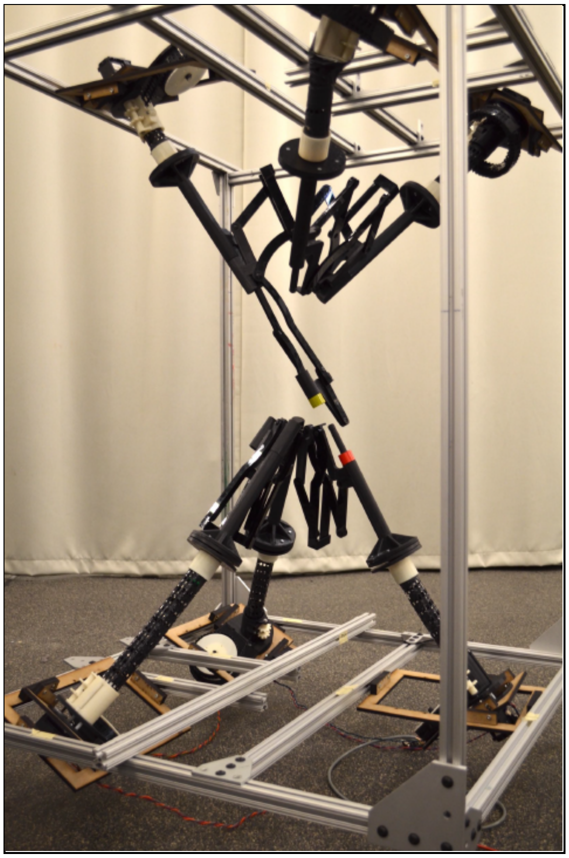


Figure 2.10: The VTT structure in a fixed frame to demonstrate the reconfiguration capabilities for IROS 2017

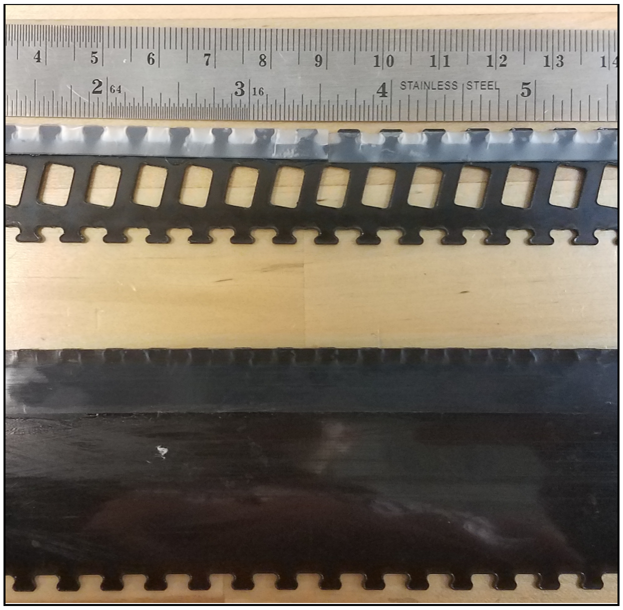


Figure 2.11: The 1” wide band (top) for the 1” diameter zipper versus the 1.881” wide band (bottom) for the 2.25” diameter zipper. Note the presence of cracking on the 1” wide band near the backing.

The backing tended to crack under cyclic loading more than the band diameter. This decreased the maximum compressive strength of the column especially at lengths exceeding 2’ and noticeably reduced the column’s stiffness. To increase the reliability of the backing the thickness is increased to 0.010” from 0.005”. We changed the manufacturing techniques to get a tighter meshing between the teeth of the zipper to increase the stiffness of the column (Fig. 12).

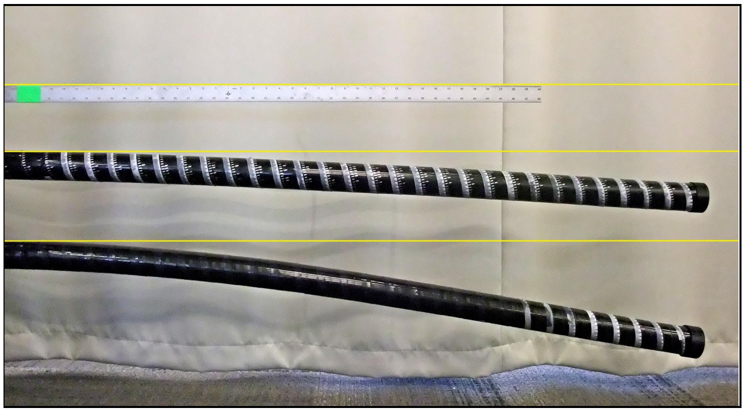


Figure 2.12: Comparing the old manufacturing method and tolerances to the new. The bottom band was created with the original manufacturing techniques while the upper was created with the newer techniques and has a tighter meshing between the teeth of the zipper.

Band Interface with Drive Mechanism:

In the original zipper, a gear interfaced with slots on the spiral zipper to drive it up and down. From a reliability standpoint this was the largest failure point in the system. Regardless of the tooth shape or extension speed, most loading conditions on the band caused the gear teeth to cut into the band causing permanent deformation (Fig. 2.13). Once the gear deformed the band, it was no longer usable and had to be replaced.

As an alternative to the tooth driven band, we designed and produced an internal friction drive mechanism to extend the zipper (Section 2.3). Not only did this remove the central failure point of the system, we were able to remove the slots in the band leading to a stronger column (Fig. 2.13). When the drive did fail it was due to slippage and did not permanently affect the usability of the band.

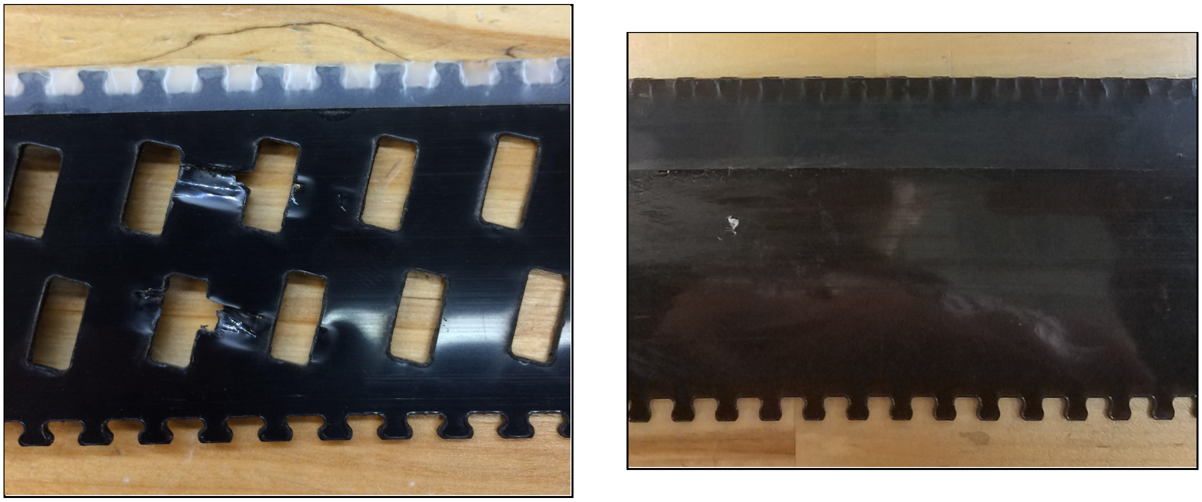


Figure 2.13: The gear driven band with deformations from the driving gear (left) and the friction drive band (right).

Previously a LIDAR sensor placed at the center of the tube tracked the height of the tube as it was extended or retracted. The internal friction drive fills the interior of the slider so this is no longer possible. Instead we use a handmade absolute encoder stuck to the outside of the band (Fig. 2.14) to track the height of the column. The lower level of marks tracks the band height along a single wrap. The upper level of marks is a binary code that tells the microcontroller the number of wraps in the column. A light sensor at the base of the slider reads this data and passes it to the microcontroller. Theoretically the sensor is accurate to within 0.6195 mm. In practice the sensor is accurate to within 1 mm.

Figure 2.14: The spiral zipper with light sensor and encoder markings. Originally marked on the zipper with a permanent marker, we now use printed stickers.

The new design allows the band to undergo 100-200 cycles (the band begins to crack) before failure even though the backing begins to chip and fall away between 50-100 cycles. In order to repair the band the broken section is removed from the full length, the remaining halves are melted together with acetone, and the joint supported with reinforced tape. While the solution is of sufficient strength, the repair operation is time consuming. Furthermore we have not explored the reliability of the joint although qualitatively we can say that it fails well under 100 cycles over the joint.

Backing Design:

We discuss how different widths and different backing material affect the performance of the band. The styles analyzed are no backing, narrow ABS backing (0.5″ wide), and wide backing (1.5″ wide) in ABS and 1095 spring steel. The end goal of this design choice was to find a backing that would increase the bending strength of the column while not significantly decreasing the performance of the band in the system. As previously discussed, we analyze the performance of each band by recording the number of times it either slows, stops, or demeshes as it extends and retracts to its limits.

The results of our testing show that a band with no backing has the best extension and retraction performance when compared to the other types of bands. The narrow, ABS backed band ranked second, while the wide, ABS backed band and the wide, spring steel band which had relatively similar performances ranked third. Intuitively this makes sense because of how the backing is attached to the freshly cut bands. To attach backing to a band, the backing and the band are joined via a liberal application of a quick dry cyanoacrylate glue on a flat surface. This decreases the natural curvature of the band. The stiffer the backing the greater the decrease in the natural curvature. As the natural radius of the band increases it becomes harder to wind the band into a column which means we see reduced extension and retraction performance.

Via qualitative observation we have found that leaving a band wound into a column for the first 24 hours post attachment of the backing to the band reduces the amount by which the natural radius of the band increases. This was most obvious in the difference between testing a band's performance immediately following the attachment of the backing and after leaving the band wound into a column for 24 hours. For all tests, bands undergoing testing were left fully wound for 24 hours after the backing was glued to the bands.

Band Strength Analysis:

In the previous report, we presented some results about the compressive strength of the spiral zipper column. This is important for the truss loading conditions, since theoretically each member will be subject to purely tensile or compressive forces. However, in actuality, the truss members are subjected to bending loads due to the weight of the module itself. These loads can be as high as about 7 N·m in our 2.25″ diameter band setup. In order to evaluate different band designs in bending, we tested several zipper columns in a four-point bending rig, shown in Figure 2.15. We chose the four-point bending test because it subjects a large section of the specimen to a constant bending moment, rather than a three-point test, which has a peak bending moment at a single point. The four-point test more closely mirrors our expected load, and avoids inconsistencies that may arise due to the periodic nature of the band teeth.

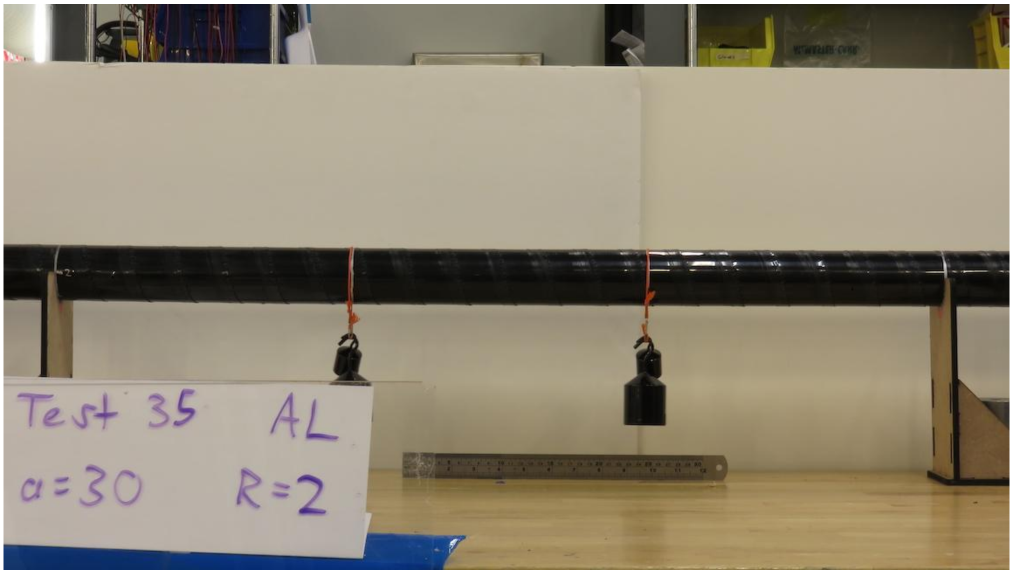


Figure 2.15: A spiral zipper column subjected to four-point bending in the test rig.

Each bending test was performed by measuring the deflection caused by at least five increments in the load, and in four different orientations of the column to account for possible asymmetry. Since some of the band specimens used combinations of different materials and different backing widths, the quantity we chose to compare was the flexural rigidity (E·I), which includes the effect of both the material and geometric properties. The results can be found in Table 2.2.

Table 2.2: Flexural rigidity of various band columns, measured in four-point bending.

Band Specimen	Flexural Rigidity ($N \cdot m^2$)
Band with no backing	34
Band with narrow ABS backing, very worn	36
Band with narrow ABS backing, slightly worn	59
Band with narrow ABS backing, fresh (1)	75
Band with narrow ABS backing, fresh (2)	112
Band with wide ABS backing, worn	79
Band with wide ABS backing, fresh	99
Band with wide metal backing	40
Pure ABS Tube (theoretical)	115

We tested three different backing designs at different levels of wear. The theoretical flexural rigidity of an ABS tube of the same diameter and thickness of the spiral zipper is included for comparison. We found that the rigidity of the column deteriorates rapidly as the band continues to wear down and the zipper teeth get looser. Also, there is a large variation in stiffness even among otherwise identical bands. The bands marked (1) and (2) in Table 2.2 were cut from the same batch, glued together using the same process, and tested before being used in the friction drive. However, band (2) is almost 50% stiffer than band (1).

The conclusion we drew from this test is that the other designs do not offer significant stiffness improvements over the narrow ABS backing, which is the easiest to manufacture, so we will continue using this design. However, we will need to measure the rigidity of each new band to ensure it meets the quality requirements. We will also perform more rigidity tests on our worn bands to better quantify how often the bands will need to be replaced.

ABS Band Further Developments

We also explore changing the tooth design to further increase the strength of the band when cantilevered and prevent demeshing. We investigate two variables: tooth radius and tooth width (Fig. 2.16). Nine configurations of these parameters were tested with a smaller length band

(~12-18”) on a cantilever test rig. We found Increasing the tooth width by 33% and decreasing the tooth radius by 7% resulted in the strongest band.

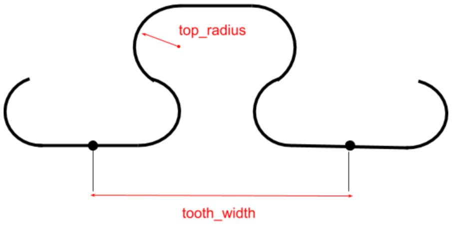


Figure 2.16: A diagram of the band tooth shape and the two variables we are investigating.

Additional tests were performed to determine the maximum number of missing teeth that will still enable the band to function normally.

Additionally we found that demeshing typically occurs at points where the band is missing teeth when this point passes through the mesh point of the slider. If the band had six consecutive missing teeth demeshing would occur under low tension. Increasing the tension to higher values allowed more missing consecutive teeth but at the cost of efficiency and the maximum actuation force of the column. A support roller placed after the mesh point supports the weakened point as it passes through the mesh point. With this design change, up to ten consecutive teeth could be missing before demeshing occurred (Fig. 2.17).

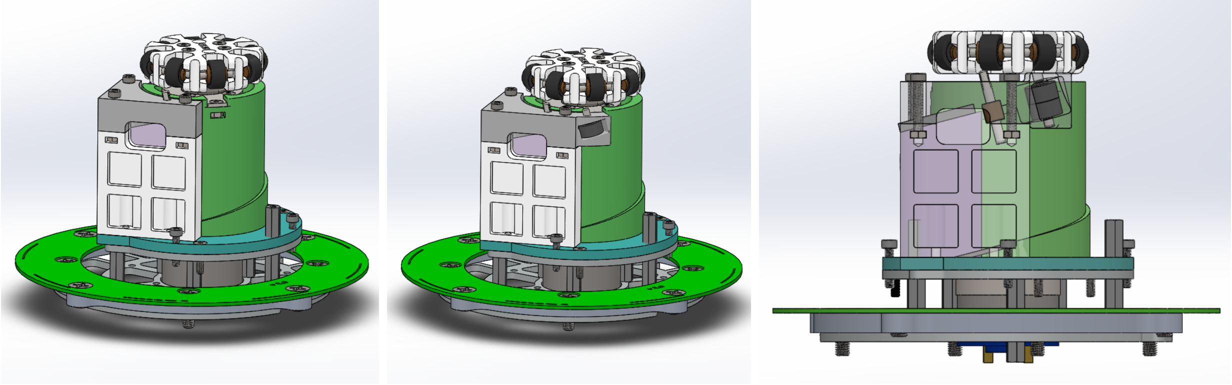


Figure 2.17: The original slider design (left), with an additional roller after the mesh point (center), with two additional rollers after the mesh point (right)

Nylon Band

The ABS spiral zipper band worked well for our initial experiments, and it was cheap and easy to manufacture. However, when we started building larger structures such as the octahedron that had more members and larger bending loads, the ABS band was not reliable enough.

The ABS band design had two main issues: stiffness degradation and broken teeth. While freshly made bands work well, once the band gets worn it begins to lose stiffness. If the drive system fails to mesh the band or a demeshing failure occurs for some other reason, this tends to break the teeth on the band. Once the band has some broken teeth, demeshing tends to occur more often, compounding the problem. As we discussed in the previous section, we made several optimizations to the band geometry to mitigate these issues, but it eventually became clear that we needed to move to a stronger, more wear resistant material.

This year, we investigated many semicrystalline thermoplastics, which are resistant to stress cracking and have better wear properties than ABS, which is an amorphous thermoplastic. The requirements for our new material were as follows:

- Easily laser-cuttable to have precise features.
- Bonds easily using adhesives
- Good wear properties.
- Inexpensive and readily available.
- Moderate flexural stiffness: needs to form a stiff column, but also needs to be flexible enough to easily transition from the spool diameter to the wound-up diameter.
- High tensile/impact strength

Most semicrystalline thermoplastics are either very difficult to glue or impossible to laser cut, so this narrowed down our choices significantly. After testing many combinations of materials and adhesives, we arrived at the combination of nylon 66 and a fast-setting, flexible-after-curing epoxy. This combination satisfied all of our requirements, except that the stiffness was slightly lower than the ABS band. To compensate for this, we increased the thickness of the band by 60% and the diameter of the column by 12% (Fig. 2.18).

Figure 2.18: Left: the previous ABS band. Right: The new nylon band.

Since the diameter and thickness of the band were changed, the new band was no longer compatible with the previous friction drive and slider. While we were redesigning those components, we made some other improvements which will be discussed in their respective sections.

Nylon Band Manufacturing Process

We use 0.05” nylon 66 for the main band material, 0.02” nylon 66 for the backing material, and 3M DP100+, a two-part epoxy adhesive.

Before gluing, we roughen both the band and backing with coarse grit sandpaper and clean the surfaces with 99% isopropyl alcohol. Since the epoxy takes a few minutes to set, we can’t glue the band in small sections on a single desk any more. Instead, we lay out the entire 20 ft band along a series of tables. First, we attach the first few inches of the backing using a fast-setting CA. This will prevent the backing from sliding around too much while the epoxy is setting. Then, one person applies the epoxy using a mixer nozzle in a thin www shaped line, the same way we used to apply the CA. The second person follows behind, making sure the backing is aligned properly with the top of the band as the glue sets and cleaning off any excess glue (Fig. 2.19).

Once the gluing is finished, the band is left alone, still flat, to set for 20 minutes or so until the backing no longer slips around and the epoxy is gummy but not hardened. At this time, we check over the teeth and make sure none of them have too much epoxy trapped between them. If any do, we clean them out with a pick or precision knife. Once clean, we let it set for another 30 minutes or so while still flat to reach a decent strength, then wind it up and leave it to fully cure for 48 hours.

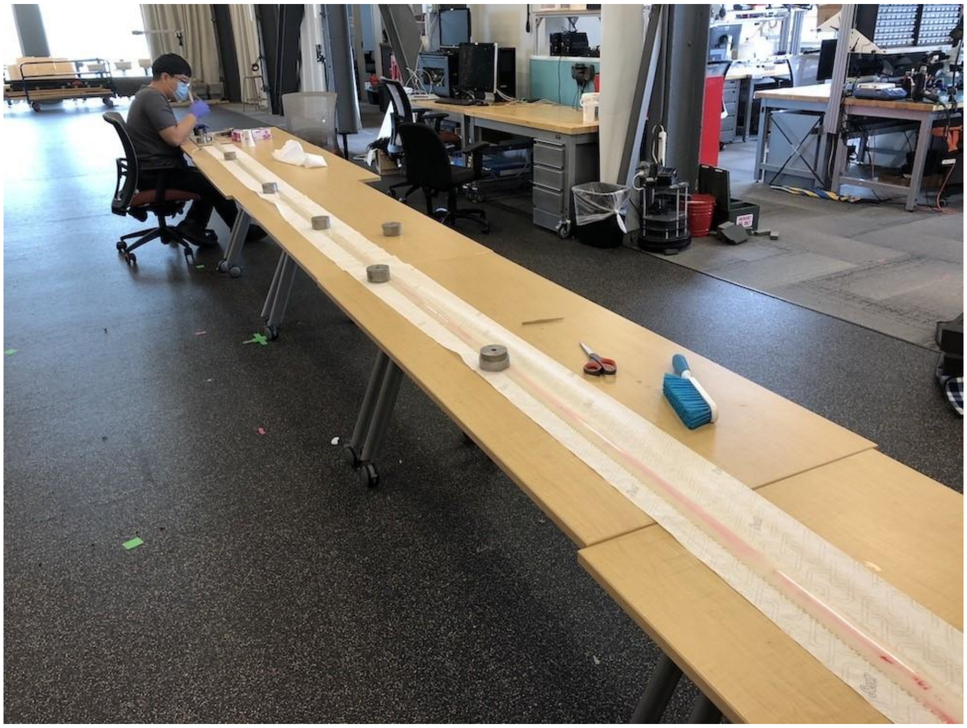


Figure 2.19. Band manufacturing process.

This procedure seems to be the most successful so far in yielding a band with no gaps in the glue application or meshing issues.

2.3 Drive Mechanism

The drive mechanism is the way the motor transmits torque to the band. Early prototypes used a geared mechanism to drive the band, but this compromised the strength of the band. We soon moved to a friction-drive design, which we continued to improve throughout the project.

Friction Drive Design:

The friction drive design, shown in Figure 2.20, is a motor with a friction drive wheel attached. The friction drive has an interference fit with the band column and uses friction to extend or retract the band column. The motor is chosen for its large torque output and operates close to the maximum output power condition of the motor.

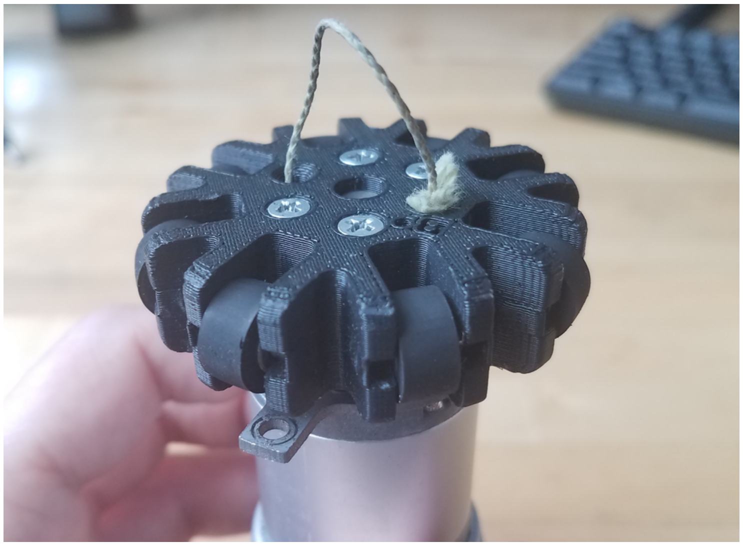


Figure 2.20: Image of friction wheel with rollers attached to the drive motor to create the friction drive assembly.

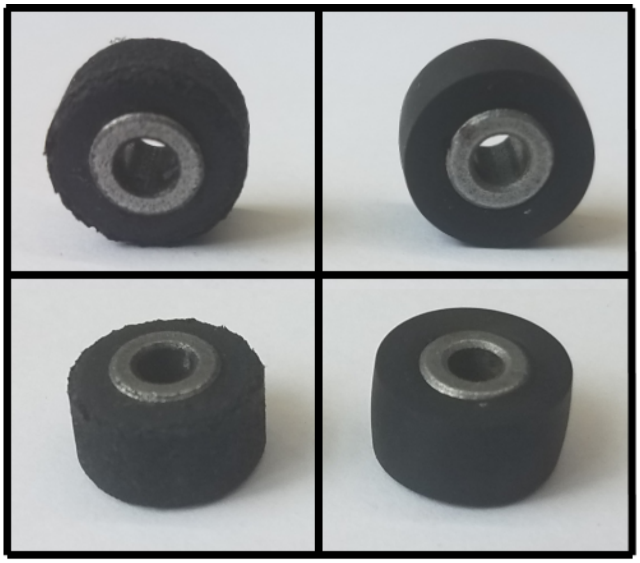


Figure 2.21: *Left:* An old roller after use in the friction wheel; *Right:* A new roller prior to use in the friction wheel.

The second problem in the design was an early fatigue failure of the friction drive wheel. The friction drive wheel consists of a wheel with 8 slots to hold the custom made rubber rollers in Figure 2.21. These rollers are manufactured by press fitting a bronze bushing into the rubber roller from a cassette wheel. Throughout testing we have found that these rollers require replacement after a low number of tests (fewer than 50 full extensions/retractions of a 5′ tall column). If they are not replaced the wheel begins to spin freely within the zipper and is not able to adequately extend the zipper under a no-load condition let alone under the 40N compressive loading condition we will need to overcome in order to use the zipper in a practical setting. With fresh rollers we have shown that we can actuate up to 50N in compression.

The proposed solutions were (1) increase the amount of interference between the rollers and the slider and (2) change the rubber material on the rollers to a higher durometer that would be less likely to wear under the loads these wheels were experiencing. In the case of (1) we expected to see the rollers slip less as the wheel was spun, reducing the amount of wear the rollers saw. In (2) we expected that the higher durometer would be better able to withstand the wear seen as the rollers slipped within the column.

Testing of both solutions was performed. To quantify the performance of each friction wheel, a band was timed to full extension and to full retraction at the top speed of the motor. Notably, the band tested appears to retract at the same rate, independent of the amount of interference or the type of rubber used in the roller. The difference can be seen in the time to full extension. Interestingly we observed that the higher durometer rollers tended to slip more than those made from the cassette wheels, even as the nominal interference increased. From the data we see that the best solution appears to be using the cassette wheels with either a 0.020″ or 0.030″ interference fit to the band.

As a mechanical solution to (1), we designed a friction wheel with a variable diameter (Fig. 2.22) (presented at UR 2020). As the screw is tightened the arms holding the rollers extend outward increasing the diameter of the wheel. In this way we are able to account for the wear in the rollers during normal operation of the spiral zipper and can increase the overall life of the system before we must replace the rollers.

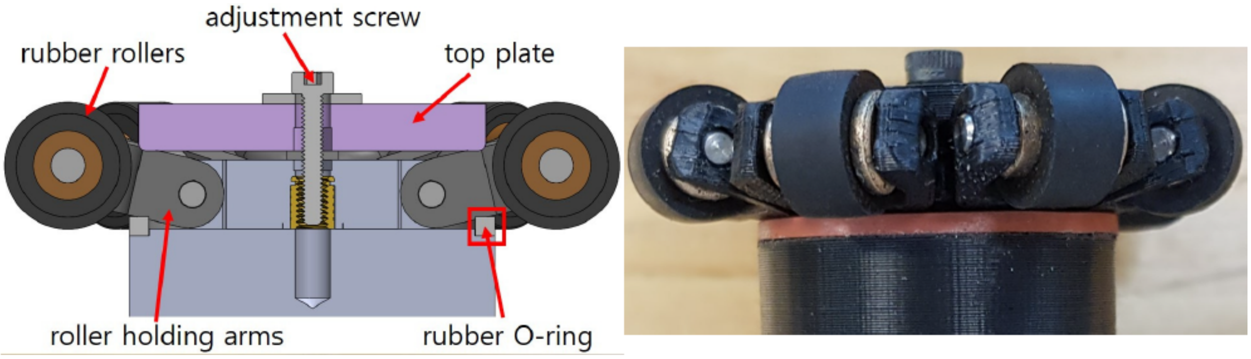


Figure 2.22: CAD design (left) and prototype (right) of the variable diameter friction drive

Support Ring Design:

The addition of a support ring, Figure 2.23, is incorporated into the drive system in order to enable operation when the column is subjected to a bending load. When the edge module is in an off-vertical position, the weight of the module induces a bending load on the module. The truss structure that will be created from the edge modules will require the zipper to be able to operate at arbitrary spatial orientations, so the drive system must be able to support this bending load. The design consists of 2 rings of rollers (identical to those used in the friction wheel) attached to a lazy susan. The lazy susan rests above the slider on a cone. The bottom row of rollers in this design are placed at the same height as the rollers in the friction wheel. This design choice allows us to increase the diameter of the friction wheel more than we would otherwise be able to without causing a demeshing failure. This, in turn, allows us to create a stronger interface between the drive mechanism and the column resulting in better actuation performance from the band and a lower fatigue life in the friction wheel rollers. The cone assists the slider design by holding the door rigidly in place. Prior to this addition, the door had the ability to deflect slightly as the band passed by it. This had a tendency to cause demeshing in the system as the meshing roller could only exert so much force before the door would deflect.

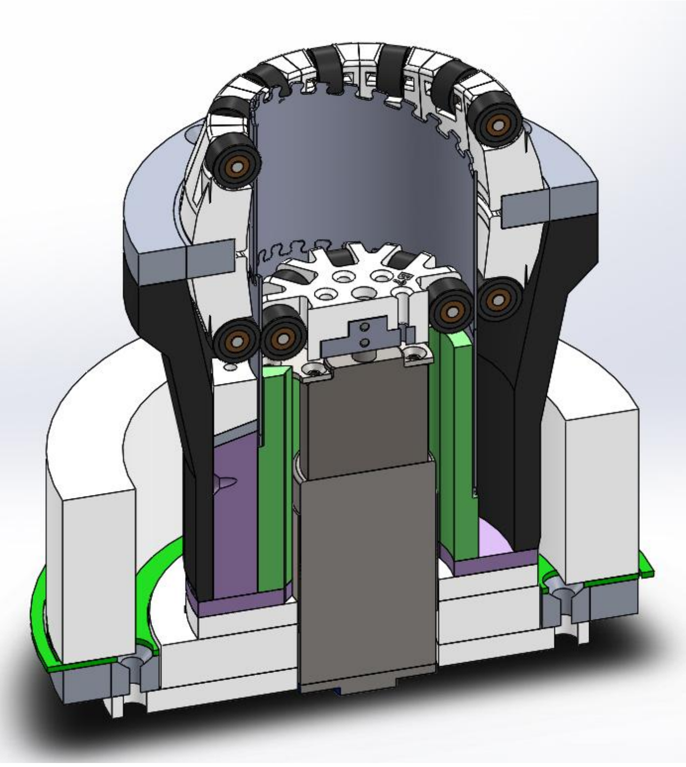


Figure 2.23: A split view of the CAD showing the support ring design surrounding the spiral zipper (grey), the slider (green and purple), and the friction wheel (white).

The design of this system reduces the deflection of the band relative to the friction drive by 0.6° when the module is subject to a 2 N·m bending load in the simply supported case. Another source of deflection can be attributed to the lazy susan which deflects 0.5° when in cantilever. Finding a stiffer lazy susan would help further reduce the maximum angle of deflection. However, regardless of this deflection, the support ring performs as expected and enables the friction drive to extend and retract the column in a horizontal configuration.

Modifications for Nylon Band

For the nylon band system, the diameter of the band was increased. The design of the friction drive wheel was kept the same; we simply increased the diameter to match the new band.

Although we did not change the friction drive wheel, we took the opportunity to experiment with a new support ring design. The major shortcoming of the support ring design was its weight. Even after the weight reduction efforts described in the previous section, the aluminum lazy susan was still a significant source of weight. Additionally, the support ring does not help isolate the meshing mechanism against tension and compression forces.

To address these issues, we designed a new support ring (Fig. 2.24) with angled rollers which are aligned with the pitch angle of the band helix. Since the rollers are precisely aligned with the band motion, we only need one degree of freedom and can eliminate the lazy Susan entirely.

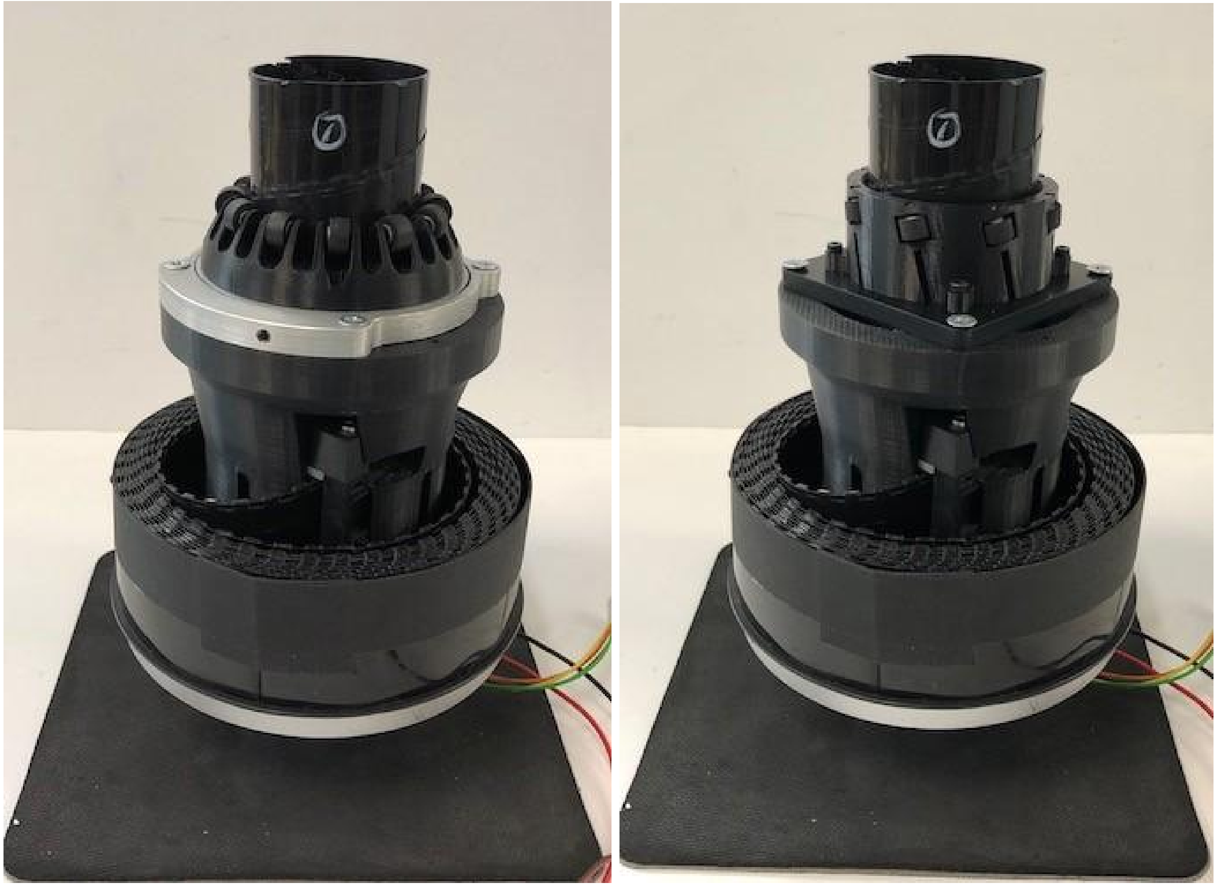


Figure 2.24: *Left:* standard support ring. *Right:* angled roller prototype.

After the slider redesign that will be discussed in the next section, we integrated this new support ring design into the new low-friction slider for the nylon band.

2.4 Band Slider

The slider is the guide that aligns the band to mesh and de-mesh correctly.

Modular Slider:

The purpose of the slider is to provide a site for the spooled band to mesh with and become part of the extended spiral zipper column. The previous version of slider was 3D printed from ABS as two parts: the main body and a door with a meshing roller to force the flat parts of the band to mesh with the column. The new version has been broken up further into more specialized parts.

Fatigue:

When using 3D printed ABS as the slider material, wear quickly becomes a problem as the spiral zipper band, which is a 1/32″ thick ABS strip, moves repeatedly over the slider surface at approximately 1′ per second. The entrance to the ramp of the slider in this design tends to wear away and the slider performance degrades to the point where the band has difficulty meshing with the existing column. A robust, albeit infeasible solution to this problem is to use a metal slider. to provide a hard, wear resistant surface on which the band could run. The timeline for this project and the quantity of sliders required made this solution impossible.

An initial design solution was to break the slider into four separate sections (Fig. 2.25). The door of the slider is broken into two parts for ease of assembly. The modular design enables the user to replace fatigued members of the assembly as needed. Furthermore, the parts can be interchanged in a cost effective manner as we iterate through design ideas. Figure 2.26 demonstrates one such possible design change where the band enters the slider on a slight curve rather than entering the slider completely straight.

Figure 2.25: *Top image:* Fully disassembled (clockwise from left: exterior column, center column, door, base); *Bottom image:* Center column attached to base; *Right image:* Completely assembled.

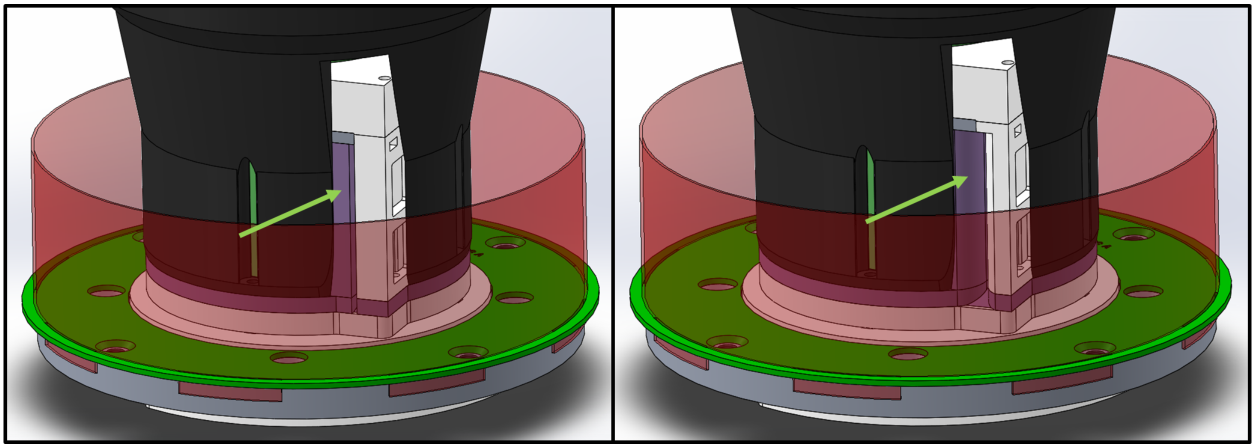


Figure 2.26: *Left:* The original interface design between the slider with the band management system. As the band enters the slider it must turn sharply otherwise it will not be pulled into the slider. *Right:* The new interface design between the slider and the band management system. As the band enters the slider it does not have to turn as sharply, reducing the amount of energy that needs to be put into the system to drive the column upwards.

In addition to the modular design, we introduce a wear plate to the slider, shown in Figure 2.27, at the location seeing the most wear over the course of general operation to improve the fatigue life of the slider. Four different materials were tested: ABS, ABS covered in PTFE, aluminum, and HDPE. These different wear plates are shown in Figure 2.28. Of these four wear plates, the ABS plate had the lowest extension/retraction performance of the spiral zipper. HDPE had slightly better performance compared to the ABS covered in PTFE and the aluminum plate.

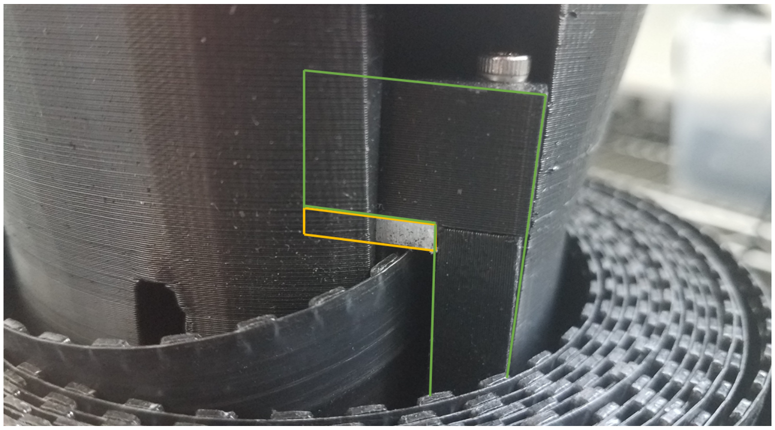


Figure 2.27: The wear insert, outlined in yellow. The door is outlined in green.

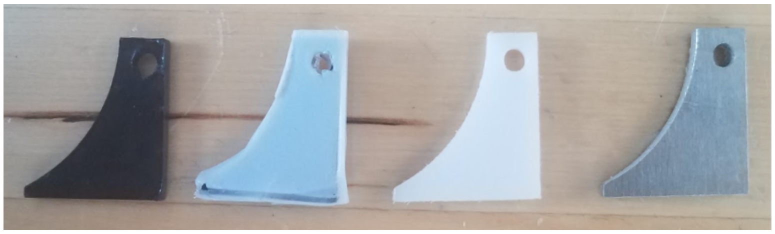


Figure 2.28: Examples of wear inserts (left to right: ABS, ABS + PTFE, HDPE, aluminum).

Meshing Roller:

The engineering purpose of the slider is to provide a location where the band can repeatedly and reliably mesh and de-mesh to and from the extended column. The addition of the meshing roller in Figure 2.29 is a design change to increase the reliability of the zipper. The roller puts pressure on the band and forces it to mesh with the column at a precise and consistent location.

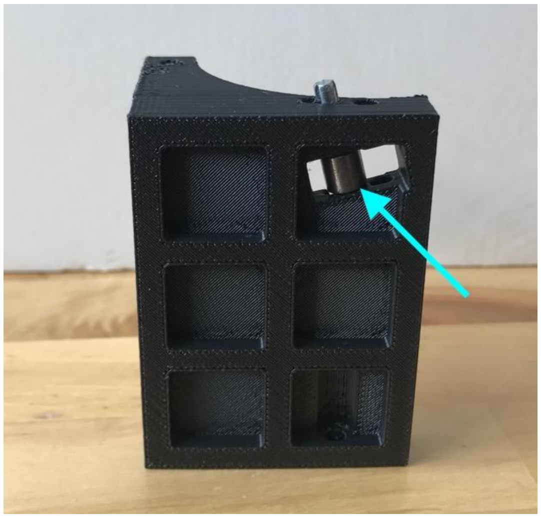


Figure 2.29: The slider door with the meshing roller indicated by the blue arrow. In this image the meshing roller is placed at site 2 as indicated in the text.

Based on the geometric interactions between the flat band and the column, we identified 3 locations where meshing was important: (1) the inner edge of the flat band first sees the outer edge of the column, (2) the point at which the band is halfway meshed with the column, and (3) the point where the band is fully meshed with the column. Qualitatively each of these locations seem to increase the system's reliability more than systems run without a meshing roller.

Two variables determine the effectiveness of the meshing roller: the location at which it forces the band to mesh with the column and the amount of pressure it puts on the band. Too much pressure increases the amount of friction in the system and decreases its actuation capabilities. Lastly, by switching to a meshing roller, we are able to move from a closed slider (one in which the band is completely surrounded by walls as the band is spun by the friction wheel) to an open version of the slider shown most clearly in the right image of Figure 2.25. This results in a further reduction of the friction on the band and increased performance in terms of actuation capabilities.

Low-Friction Roller Cage Slider

The slider design presented above worked well for our first generation edge module. However, as we discussed above, wear was a constant issue due the material properties of the 3D printed part. When we switched to a nylon band, we also increased the diameter of the band column, so we had to make new sliders to match. We took this opportunity to address the wear and friction issues in a more fundamental way.

In the new slider prototype, the band no longer makes contact with any 3D printed surface. The entire helical ramp has been replaced with a low-friction, wear-resistant UHMW PE insert. In addition, the walls that guide the band into a circular path have been studded with rollers to further reduce friction. This design can be seen in Figure 2.30.

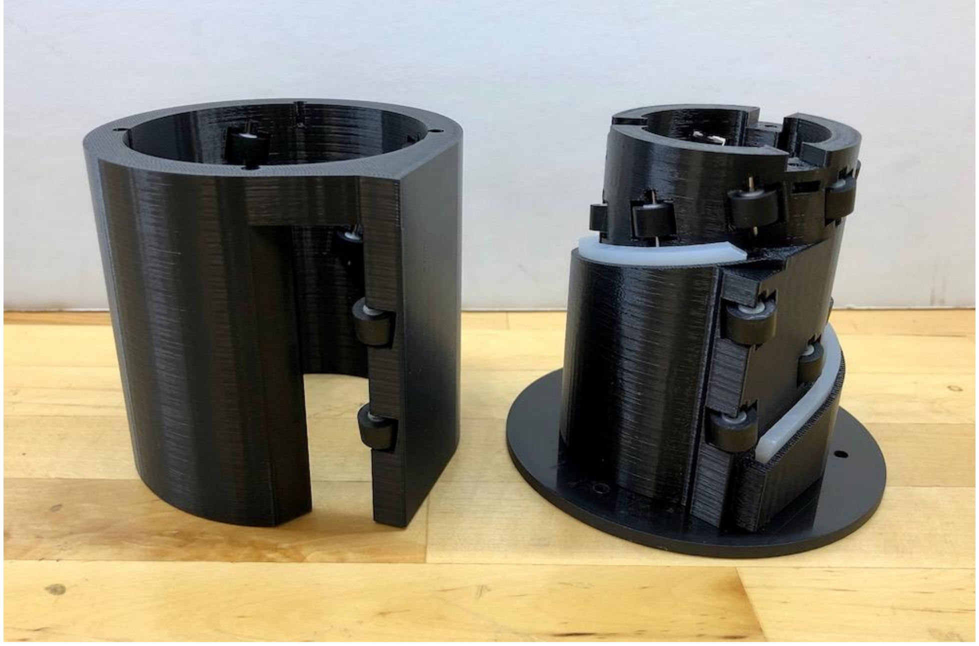


Figure 2.30: The improved slider, with wear-resistant bearing surfaces.

By stiffening and extending the outer part of the slider, we were able to attach the new support ring directly to the top of the slider, eliminating the large 3D printed cone that held the support ring in the previous design.

The new design has less friction, so it takes less energy to operate the band. At 12V, the original friction drive drew about 5.1A while lifting a 60N load. The new friction drive draws about 3.7A while lifting a 68N load. The weight of this part was also greatly reduced by switching to sparse 3D material. It is no longer necessary to print solid material for wear reasons.

2.5 Band Management System

The purpose of the band management system is to hold excess band when it is not being used in the main column. The excess band rests on a large lazy susan that encompasses the spiral zipper column and slider. When the column is retracting, the band pushes out against the walls of the mechanism and spins itself around the slider. When the column is extending, the excess band spins around the slider in the opposite direction.

The previously mentioned active node design is designed to allow for a minimum angle of 24 degrees between two edge modules. With the old band management system, a passive, spring loaded system that sits to one side of the slider, this is not always possible. This led to the development of a passive, concentric band management system (Fig. 2.31).

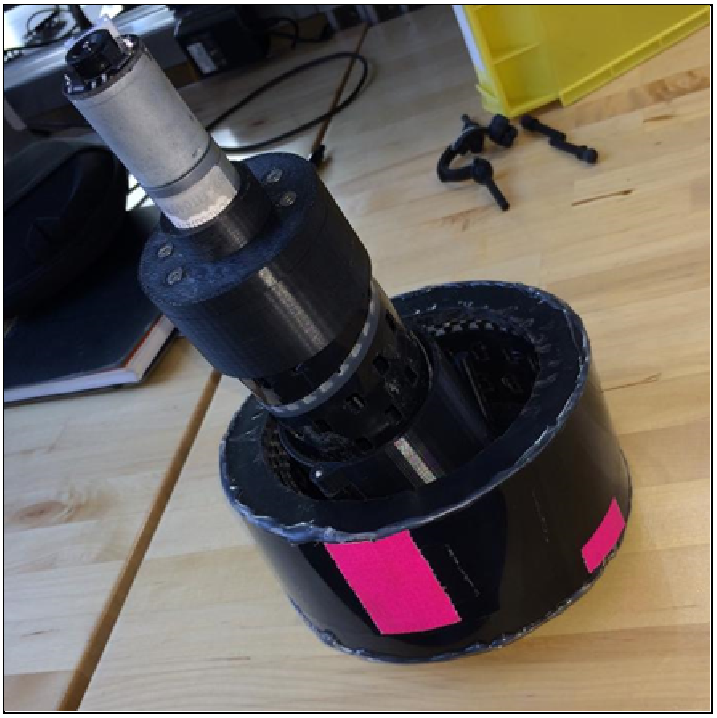


Figure 2.31: A prototype of the angled band management system.

Band Management System Design:

Last year we tackled the band management system problem using an angled band spool. While this approach worked, the solution was not easy to manufacture - the base was 3D printed. While normally acceptable, the size of this part (6″ OD, 4″ tall) and required print time made it infeasible to print the quantity of these parts required for the final VTT robot.

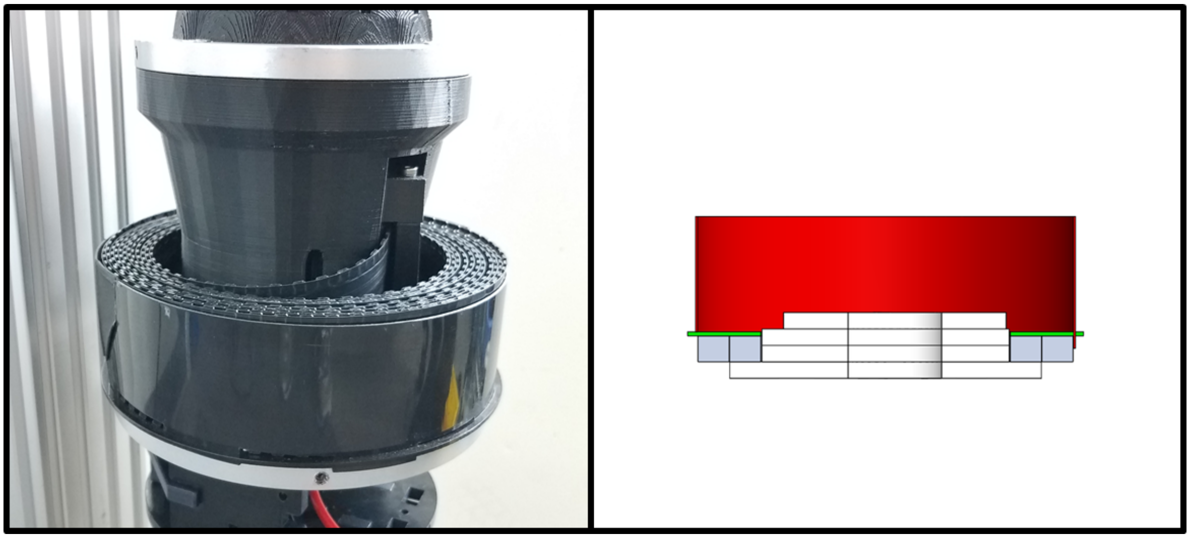


Figure 2.32: *Left:* Assembled band management system in system. The band can be seen contained by a thin strip of ABS material and sitting on an aluminum lazy susan. *Right:* The

CAD model of the band management system with the laser cut base (white) and ABS containment cylinder (red). A slider would sit on top of the laser cut base and the hole through the center of the base allows access to the motor leads.

Our solution was to laser cut the band management system instead of printing it, redesigning the containment system from a 3-dimensional system to a 2.5-dimensional system. For simplicity we redesigned the base of the band management system to be made from sets of plates stacked on top of each other. In addition to the straightforward manufacturability, this new design provided a larger band containment space. The effectiveness of the new design at multiple radii was compared with the original angled design and its containment diameter of 150mm. To compare the effectiveness of each band management system, a band was repeatedly extended and retracted in a vertical test stand while contained by each of the different systems. Regardless of the angle or containment radius, there was no significant difference in the extension or retraction performance of the band. Therefore we chose to use the more easily manufacturable, 2.5-dimension design that can be laser cut from sheets of ABS, as shown in Figure 2.32.

More recently, we have achieved significant weight savings by replacing the laser-cut ABS plates with a lightweight waterjet-cut aluminum frame, and by modifying the Lazy Susan bearing to remove excess material.

2.6 Series Elastic Tension Cable System

Although the spiral zipper is strong in compression, it can fail under large tensile forces. Additionally, the slider fails to mesh much more often when the band is in tension. To apply consistent tension to the spiral zipper band, we use a force-controlled internal tension cable system.

Initial Design

During the second year of the project, we designed several prototype tension cable systems. The design we selected uses a series elastic spool to maintain constant cable tension. It is shown in Figure 2.32 below. To improve the symmetry of the edge module, the tension cable mechanism is located at the opposite end of the edge module from the spiral zipper friction drive. Since the friction drive uses an internal drive wheel, one end of the tension cable must be attached to the friction drive wheel itself. A fishing swivel prevents the cable from becoming twisted.

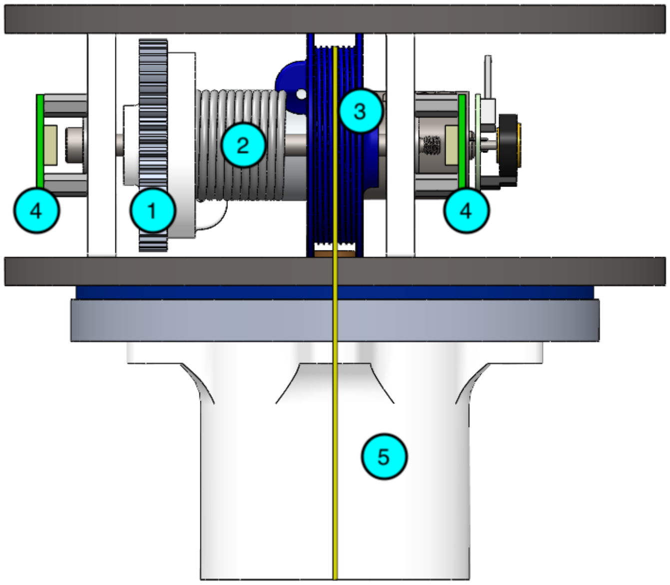


Figure 2.32: The series elastic tensioner. The drive gear (1) is connected to a torsion spring (2), which is connected to the spool (3). The yellow line shows the path of the cable, which continues down through the band column and attaches to the friction wheel. The drive gear and spool positions are measured by encoders (4). The cap (5) is the attachment point to the zipper.

The cable is stored in a spool inside the tension cable module. The spool is connected in series to the drive motor by a torsional spring. The twist in the torsional spring is therefore proportional to the tension in the cable. A pair of encoders measures this twist by measuring the rotational offset in the spool and drive motor, providing the input for feedback control. Controlling this twist allows the cable to extend and contract along with the zipper while maintaining a fixed tension. This fixed tension is configurable, although this prototype was only capable of supplying 30-40N.

Improved Design

This year, we made a few more iterations of improved prototypes to arrive at our current design, shown in Figure 2.33. The basic method of operation of the device is the same, but all aspects of the design have been improved. We produced twelve tensioner modules, which we used for our tetrahedral and octahedral truss experiments. We continue to use the same tensioner modules in our second generation edge module as well.

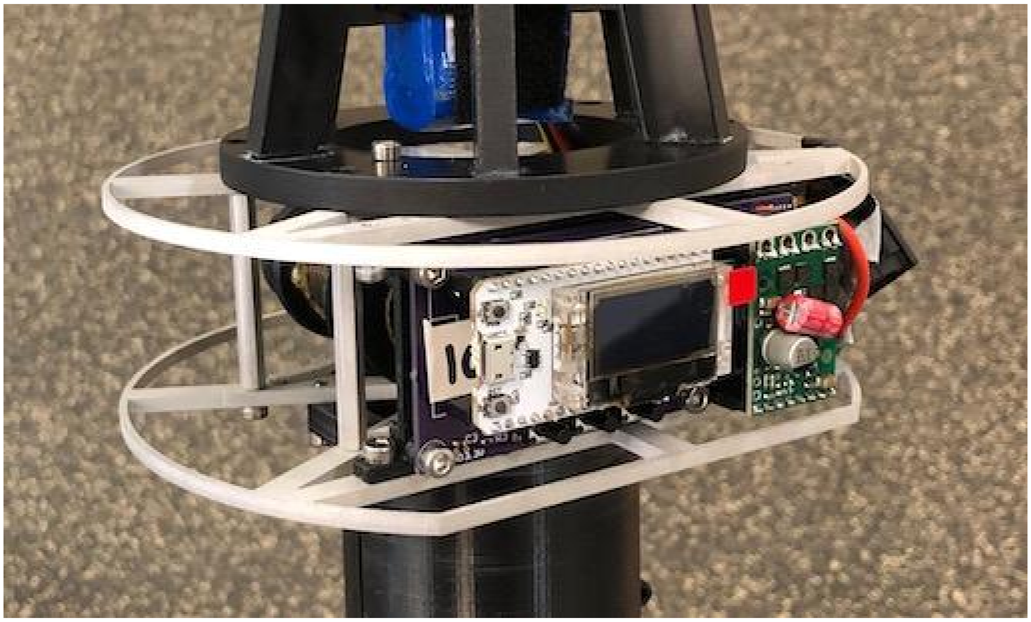


Figure 2.33: The improved tension cable system.

In the current prototype, the ABS housing has been replaced by a stiff and lightweight aluminum frame. The weight was reduced from 590g to 380g (including electronics, not including battery), and the size of the module has also been reduced. The motor, spring, and spool have been tuned to increase the maximum tension force to 80-90N, with a cable speed of 9cm/s. The module is easier to assemble, disassemble, and repair.

The perfboard circuit from the previous prototype has been replaced with a custom PCB. Additionally, the arrangement of the magnetic encoder sensors has been changed to allow the sensors to be attached directly to the main PCB, whereas in the previous design they needed to be on separate boards. The whole module can be powered by a 12V power supply or a 3-cell LiPo battery.

The microcontroller is an ESP-32 based development board with an OLED screen. The screen in combination with the buttons attached to the PCB form the user interface of the module, which we designed. Through the user interface, it is possible to calibrate the sensors, start and stop automatic tension control, adjust the tension setpoint, and manually extend/retract the cable. The board is also WiFi capable and connects to the ROS system. The tension setpoint can be adjusted over the WiFi network, and feedback from the onboard control system can be optionally monitored by the central computer to detect faults.

2.7 VTT Joints

The VTT joints are the connectors for the system; they keep all of the members connected together at the nodes of the truss. In this section we discuss both self-reconfigurable joints and fixed topology joints.

Recall the complete VTT edge module, pictured in Figure 2.34, which comprises a core linear actuator module and two VTT joints, affixed on each end of the linear actuator. The VTT joints passively enforce the truss constraints, allowing an arbitrary number of VTT edge modules to be connected together at a node in the truss such that they are constrained to rotate about the same center point. This is achieved by all of the VTT joints at a node latching their special linkages together to form a chain. The VTT joints also enable the reconfiguration of the truss; by rearranging the chain of connections between the joints at a node, the node can be made to split into two, or two nodes can merge into one.

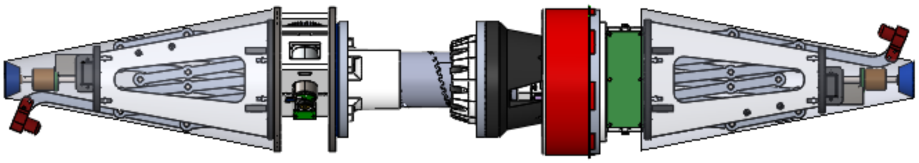


Figure 2.34: A VTT edge module, which has a VTT joint on each end.

Initial Reconfigurable Joint Prototype

Figure 2.35: A demonstration of the “chain of joints” concept to merge and split two nodes.

This six-bar linkage was the initial design for the VTT joint. It was based on Hamlin's CMS joint from Tetrobot, with a peg-and-hole connector added to allow multiple joints to connect together. We demonstrated the docking capabilities of this joint in our demo for IROS 2017, as shown above in Figure 2.35. This version had no actuators, and hence required the truss members to be in very specific locations to achieve docking.

Active Piston-Latch Design

In 2018, we added actuators and a latching system to the design from the IROS demo. It features three motors: one to control the azimuthal angle of the linkage, one to control the opening angle of the linkage, and one to control the latching connector to allow connection and disconnection of the VTT joints. The prototype is shown in Figure 2.36.

A piston-like latch was designed to allow for a simple, yet strong connection between two edge modules once they are connected. The spherical linkage was analyzed and designed to allow two edge modules to be separated by a minimum angle of 24º and a maximum angle of 156º.

This prototype demonstrated the feasibility of a reconfigurable joint, but there were issues with the strength of the linkage and the reliability of the latching system. The small minimum angle was achieved by having the pin joints of the linkage pass over each other, so the links could not all be coplanar. The latching system would have required precise alignment in both degrees of freedom of the VTT joint and had a small area of acceptance. We addressed these issues in the next prototype.

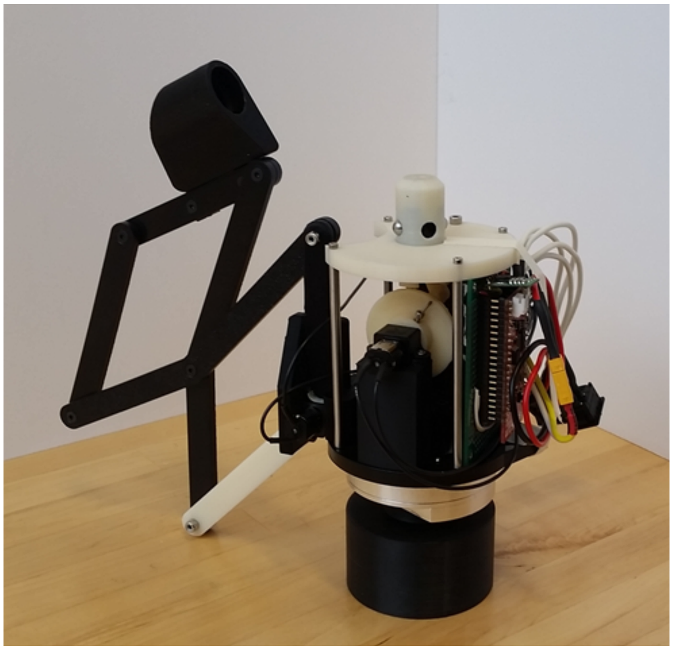


Figure 2.36: Our initial active piston-latch VTT joint.

Six-Bar Linkage with Sliding Latch

Later in 2018, we developed a new latching system with a larger area of acceptance, and we investigated some alternative linkage geometries that had better strength properties. These prototypes omit the azimuthal and linkage actuators for simplicity. The plan was to reintroduce these actuators once the kinematic and structural issues were resolved.

Latching Mechanism Redesign

The new latching mechanism improves the reliability of the latching procedure and the strength properties of the edge module. It is shown in Figure 2.37. The mechanism comprises a fixed central post, a sliding sleeve bearing that travels along the post, and a pivoting catch which is fixed to the distal end of the linkage. The top of the fixed central post makes sliding contact with the sphere, transmitting any compressive load in the truss member to the sphere regardless of the state of the latch. To latch two edge modules together, the pivot catch on the end of one linkage approaches the central post on the other edge module. Once contact is made, guiding features on the catch help align the linkage plane with the axis of the post, increasing the area of acceptance. Next, the linkage extends until the catch is fully aligned with the post, so precise position is not required on the part of the linkage actuator. Finally, the sleeve slides upwards and captures the catch. As a result, the catch is free to pivot about the central post, but the post axis is held in the correct position for the truss constraints.

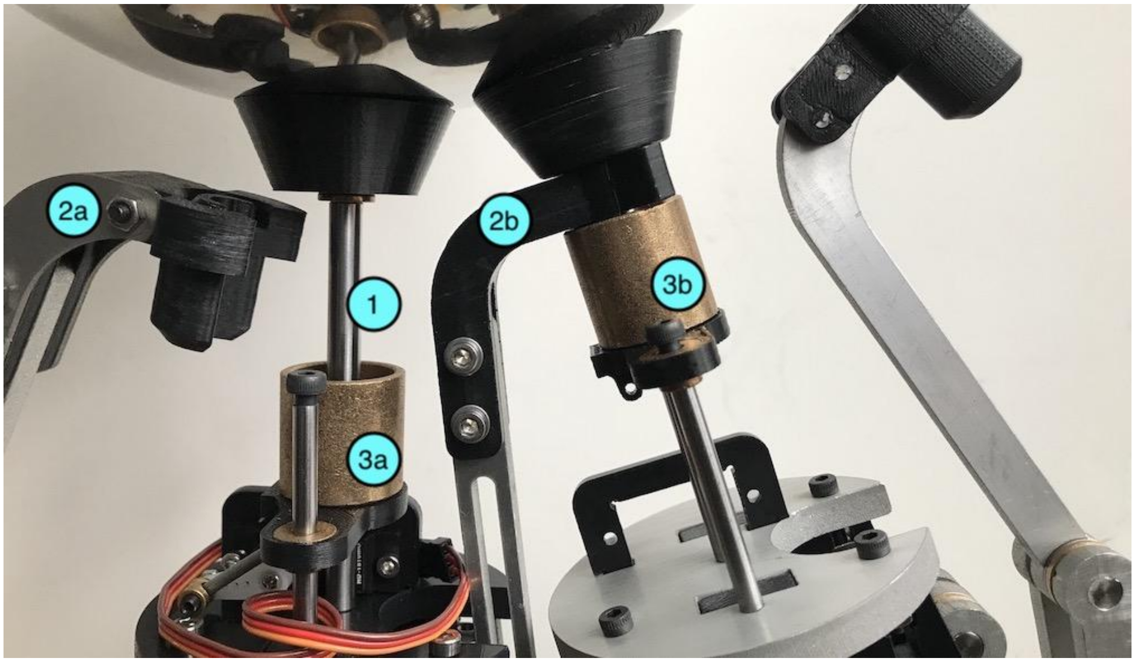


Figure 2.37: Three VTT joints demonstrate the latching procedure. A VTT joint offscreen to the left is preparing to latch with the central VTT joint, which itself is already latched to the VTT joint on the right. A pivot catch (2a) on the end of a linkage (off screen to the left) approaches the central post (1). The sleeve (3a) is in the down position, in preparation for latching. The second

pivot catch (2b) is already latched in place by the sleeve (3b), it can pivot about its post but is otherwise constrained.

Although the first iteration of this design had a very small interface and did not support bending moments well, this aspect was later improved. This concept worked well enough that all of our future prototypes used some minor variation of this design.

Six-Bar Linkage Redesign

We adjusted the placement and the link lengths of the linkage to achieve a minimum angle of 18° and maximum angle of 155° without needing the pin joints to cross over each other, as shown in Figure 2.38. This allowed us to use a much stronger design for the pins and keep the linkage forces coplanar. The downside of this design is that it needs a much larger amount of space in the edge module direction. It extends away from the sphere center a distance of about 2.25 times the sphere diameter, while the previous design only extends about 1.6 times this distance. Three of these VTT joints were manufactured and sent to the URAI 2018 conference for a demo.

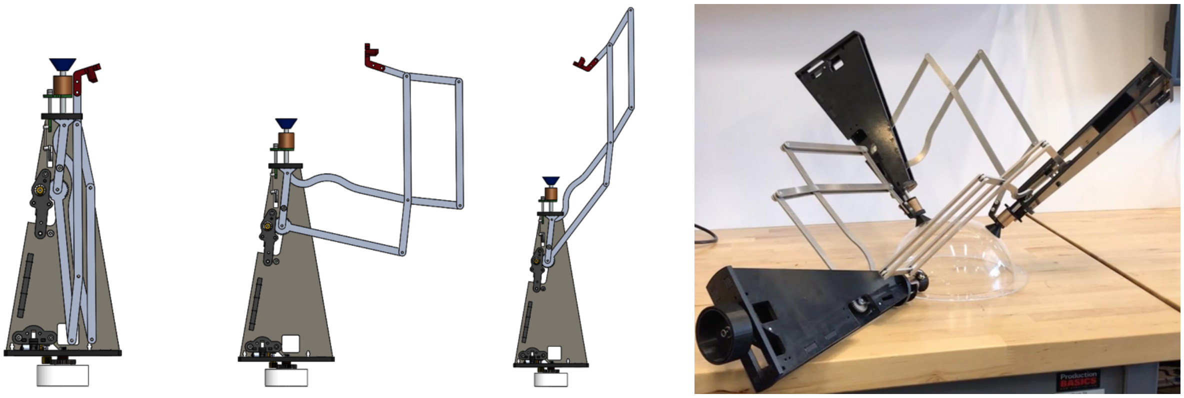


Figure 2.38: On left: the six bar VTT joint at angles of 18°, 90°, and 155°. On right: the three six bar VTT joints demonstrated at URAI 2018.

Twelve-Bar Linkage with Sliding Latch

In order to preserve the favorable pin joint design while shrinking the extents of the linkage, we designed a novel twelve bar offset planar hinge, shown in Figure 2.39. The extra links allow the linkage to extend further although each link is shorter overall. The kite-shaped part of the linkage introduces a shape change at large angles that improves the clearance with the sphere, so the maximum angle can be much larger. The minimum angle is slightly worse at about 22°, but the maximum angle is 165°, and the furthest extent is only 1.9 times the sphere diameter. Additionally, there is more space above the linkage to widen the latching mechanism. We used this design in several of our experiments, and the SNU team built six identical VTT joints using this design to validate it in their tetrahedron test rig.

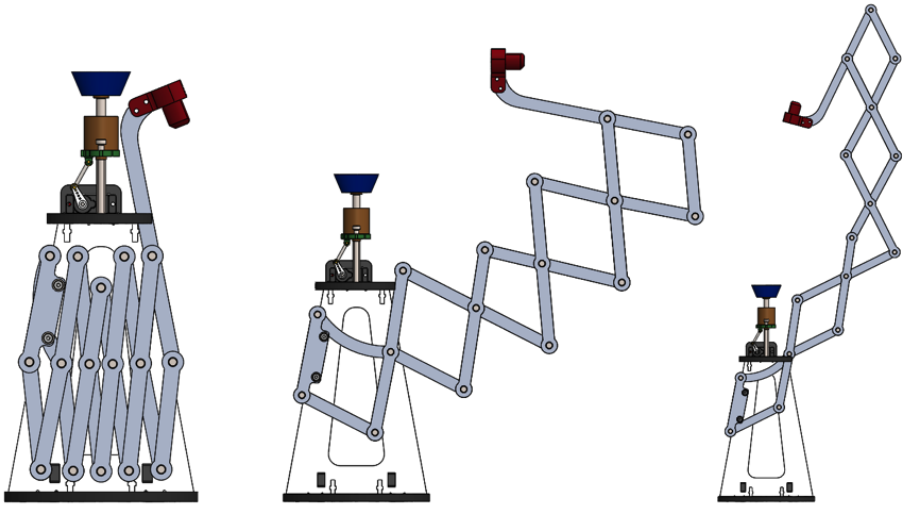


Figure 2.39: The twelve bar VTT joint at angles of 22°, 90°, and 165°.

Fixed Passive Six-Bar Linkage

The previous design worked well when the members were well aligned and the internal forces in the truss were small. However, when we performed the overconstrained motion test (discussed in the next major section), we generated larger internal forces that the previous design could not handle. This year, we attempted to resolve this issue. We returned to a six-bar design, this time with very different linkage parameters compared to the original six-bar design.

This design was focused on having much more favorable load paths for the internal forces in the system. Since we were interested in pursuing some non-reconfigurable truss topologies for our locomotion tests, we made the design simpler and more robust by removing the latching system from this design. A chain of three of these joints around a sphere can be seen in Figure 2.40.

Figure 2.40: A chain of three passive six-bar linkage joints.

We manufactured 24 copies of this linkage design, which we used for our tetrahedral and octahedral locomotion tests.

Reconfigurable Six-Bar Linkage with Improved Latch

Our next plan was to reintroduce the sliding latch system to the more robust six-bar linkage geometry while maintaining the favorable structural properties and robustness. We made some initial designs and prototypes towards this idea, but due to COVID-19 we dropped this aspect of the project to improve the spiral zipper actuator and focus on the locomotion results.

3. VTT Software

3.1 Software Control Architecture

Control Framework

We developed two models to describe a variable topology truss: a graph model and a kinematics model. The graph model is used to describe the topology connection among all links and the kinematics model is used to describe the motion of the robot.

A variable topology truss can be easily modeled as an undirected graph $G = (V, E)$ where V is the set of vertices or nodes of G and E is the set of edges or members of G. The set of all vertices $V$is separated into two subsets V_F and V_C where V_F contains all the fixed nodes and V_C contains all the controlled nodes. Then the kinematics model is built to describe the motion of all controlled nodes, which can be written in a general form.

A feedback position controller based on the resolved motion rate control method is developed to control the states of all controlled nodes. This controller computes the input for a nonlinear model first, then converts this input into corresponding commands for each motor velocity controller on each member. The motor velocity controller gets the angular position from a quadrature encoder and controls the motor to run at a desired angular velocity. The control structure is shown in Figure 3.1.

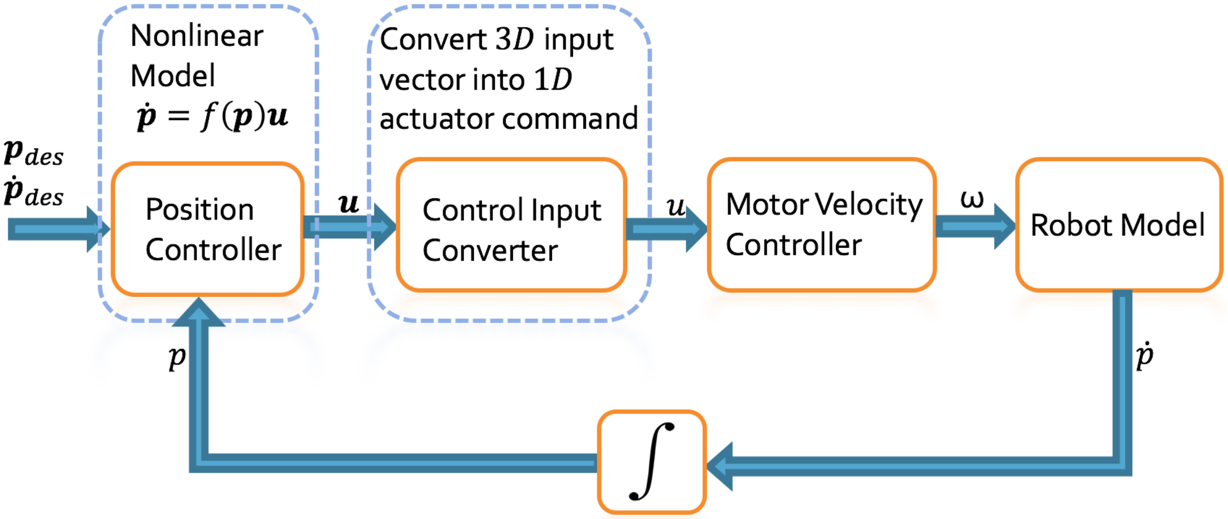


Figure 3.1: The control loop for position control

ROS Package and Hardware Implementation

We developed a ROS package for the variable topology truss system. The robot manager can keep track of the robot states and provides multiple ROS services and actions to control the robot or get information about the robot. The Boost Graph Library (BGL) is used to track the topology of a variable topology truss and Boost ODEINT library is used to simulate the robot motion given control input. The simulation result is visualized in RViz. The controller in the previous section is already implemented in C++ and it is straightforward to construct a control object for motion control. The communication between the simulator and ROS control package is designed in a distributed way, so that the same controller used in the simulator can be applied on the hardware directly.

For hardware implementation, we have demonstrated the geometry reconfiguration actions on a spiral zipper arm which is similar to a simple variable topology truss. Each edge module has a STM32F303K8 microcontroller for motor velocity control and an ESP32 for wireless ROS communication. The structure is shown in Figure 3.2 and each edge module and the corresponding object in the simulator subscribe to the same ROS topic when under control.

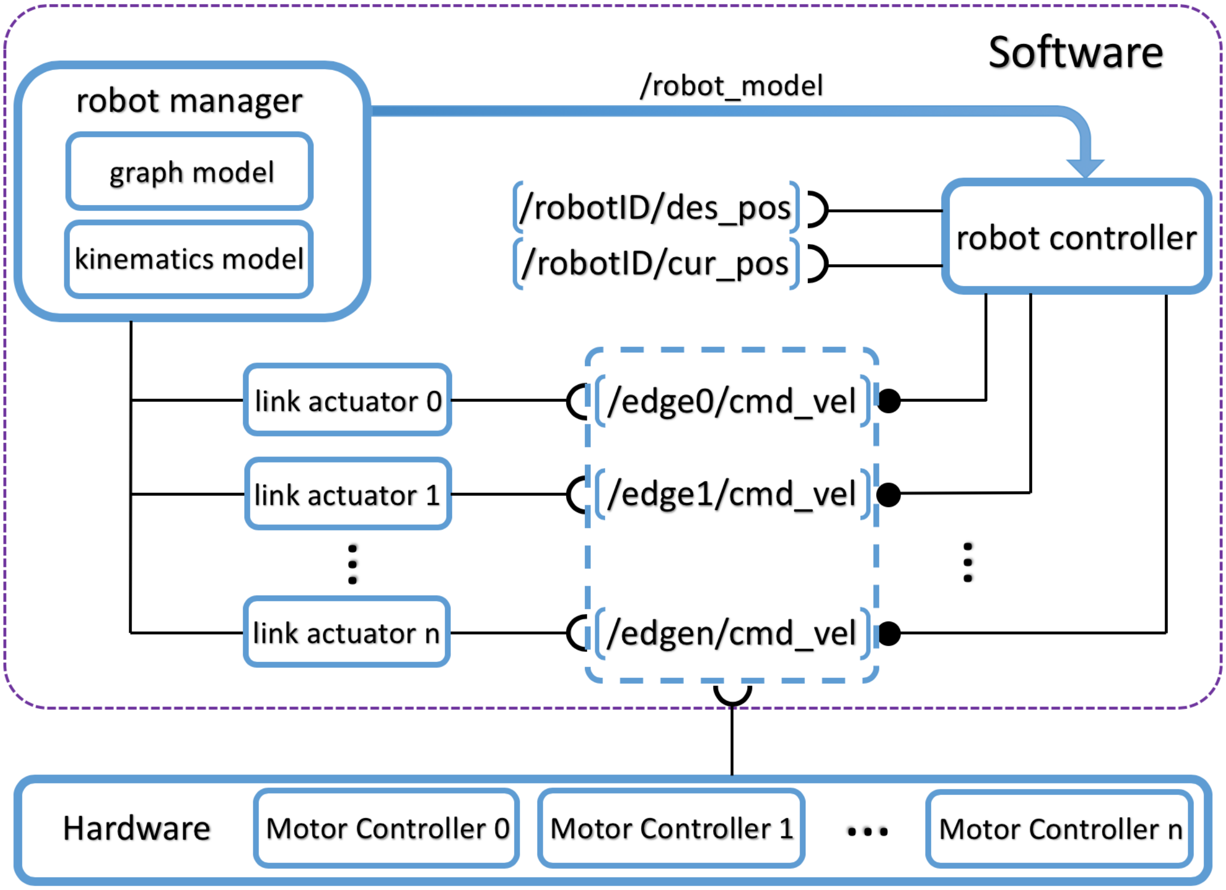


Figure 3.2: Control Software Architecture

3.2 Planning and Algorithmic Results

Topology Selection and Planning

The variable topology truss (VTT) is capable of changing the way the truss members are arranged in the truss (i.e., the truss topology). From the beginning of the project, we studied the capabilities and limitations of this reconfiguration process. For this analysis, we consider only the actions that result in high-level topological changes, and ignore details concerning the exact location and shape of the truss.

We determined that a minimum of 18 members are required for reconfiguration. During reconfiguration, the number of truss members must be preserved and the truss must remain rigid. From these constraints it can be determined that at least 18 members are required, resulting in trusses that transition between having 7 and 8 nodes. There are only a small number of unique topologies for 18 member trusses, but adding additional members rapidly increases the number of topologies and the potential reconfiguration actions.

We began to enumerate all rigid reconfigurable truss topologies with a small number of members. These trusses satisfy generic rigidity and are capable of at least one topology change. By identifying these trusses, we reduce the search space to only the potentially interesting topologies. For example, there are more than 20,000 truss topologies with 21 members and 9 nodes, but fewer than 1,500 are rigid and reconfigurable.

To understand the reconfiguration capabilities of VTTs, we construct topology networks, which describe the possible reconfiguration actions. A topology network is a graph where the vertices represent topologies, and the edges represent reconfiguration actions that take one topology to another. The result for reconfigurable 18 member trusses is shown in Figure 3.3. Initially, we were only able to compute the results for all trusses with 18-23 members.

We performed investigations to find trusses with useful topologies for the required tasks. This first involves identifying truss topologies which satisfy desired properties, such as exhibiting a quadruped structure which is suitable for walking or a multilayer structure which is suitable for shoring high ceilings. Then, we search for a set of topologies which together can satisfy all of the requirements while minimizing their relative distance on the topology network. Through this process we have identified some candidate structures to investigate further. Some of these results are shown in Figure 3.4. However, since this analysis ignores many details about the geometric shape change of the truss, additional analysis will be required to ensure collision free paths exist between the target topologies.

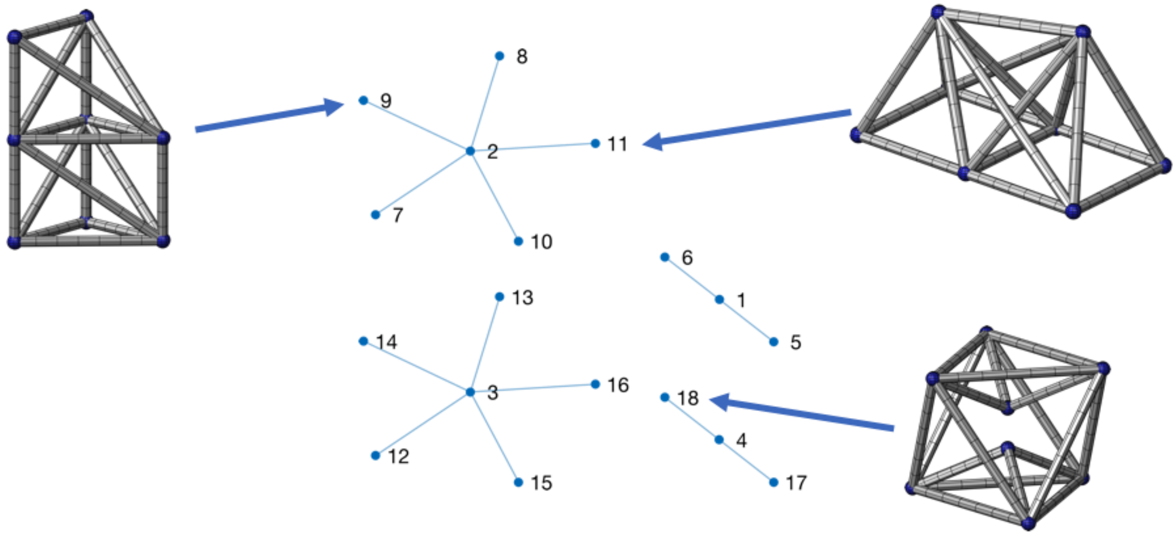


Figure 3.3: The topology network for reconfigurable 18 member trusses. For example, to reconfigure from topology 9 to topology 11, the VTT must first reconfigure into topology 2. Topology 18 cannot reconfigure into topology 11 because there is no path in the graph.

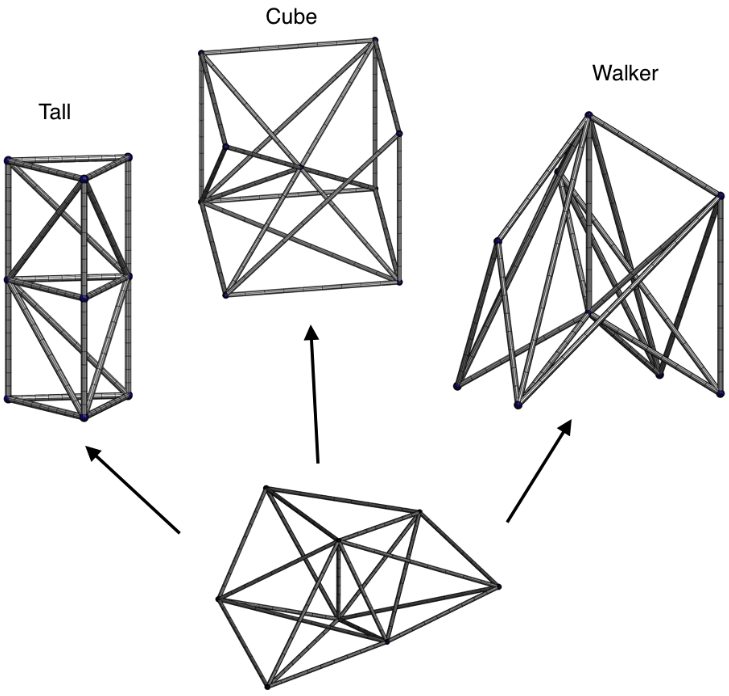


Figure 3.4: A possible set of target topologies for a shoring mission. A tall configuration for shoring ceilings, a cube configuration for rolling locomotion over smooth surfaces, a walking

configuration for locomotion over rough terrain, with all configurations accessible by reconfiguring through the intermediate configuration shown below.

Extension of Results

This year, we greatly improved the efficiency of the topology network generation program. We extended our results to include trusses up to 29 members, which includes many more useful topologies. For example, we can now find a path of topologies to connect the tower and dome configurations shown in Figure 3.5, both of which have 27 members. Such a path is shown in Figure 3.6.

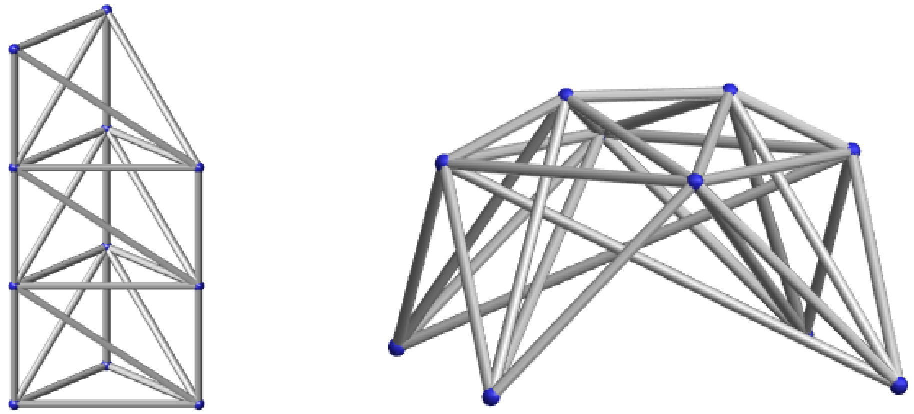


Figure 3.5: Two potential goal configurations for a VTT.

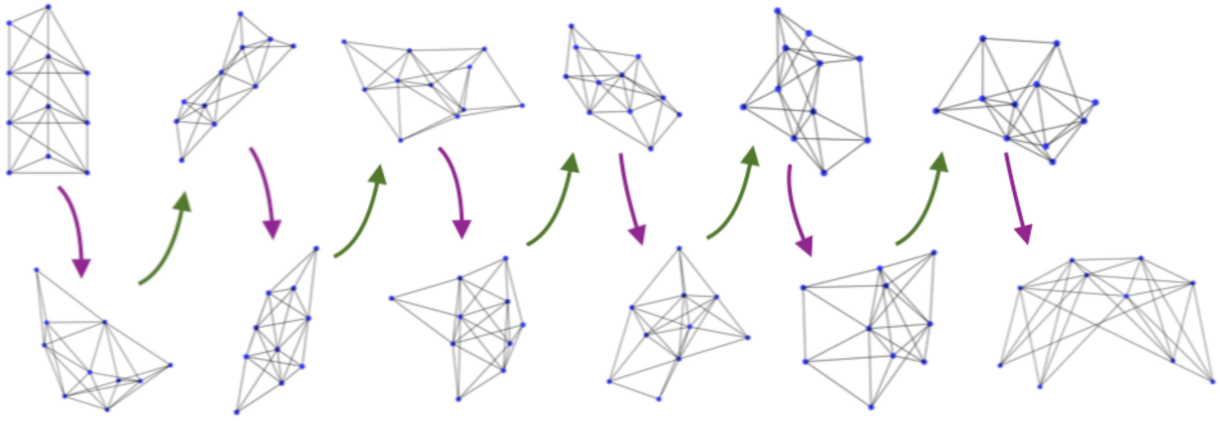


Figure 3.6: A sequence of topologies obtained by merging and splitting nodes that takes a truss from the tower configuration to the dome configuration.

In fact, we found that nearly all reconfigurable 27 member topologies are mutually reachable. The result for 18 members above shows four disconnected connected components of the topology network. A similar situation arises for 21 and 24 member topologies. There are many separate connected components, and it is often impossible to reconfigure from one topology to another. Once 27 members are reached, these connected components coalesce, and it is much easier to find paths connecting different topologies. We expect this property to continue to hold for larger trusses as well.

3.3 Geometry and Topology Reconfiguration Planning

In this section, we discuss methods that can plan paths for general shape-change operations while avoiding collisions between the members

Sampling-Based Probabilistic Planner

From the results in the previous section, we can identify some target topologies that satisfy the requirements of our shoring tasks. From the topology network, we obtain information about the high-level actions that need to be taken to reconfigure. For example, we may find that at a certain step we need to merge two nodes in the truss. The geometric path planning step determines exactly how the truss should be actuated to bring these two nodes together while avoiding collisions. An example of this is shown in Figure 3.7.

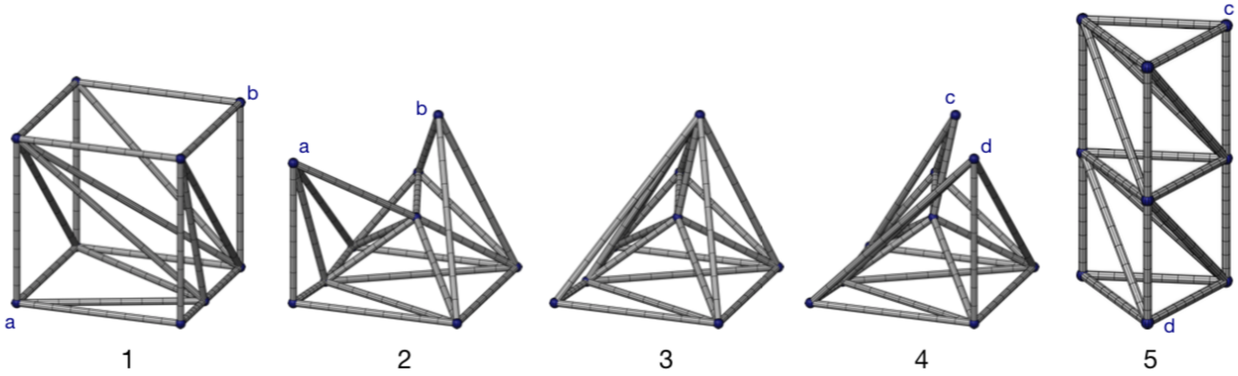


Figure 3.7: To reconfigure from the cube shape to the tower shape, we must merge nodes (a) and (b) and split them into nodes (c) and (d). Geometric path planning takes place between steps 1 and 2 to bring the nodes close together, and then once again between steps 4 and 5 to reach the target configuration.

The first step to solving a path planning problem is to develop a collision checker for truss configurations. We have developed a collision checker that, given a description of the truss topology and the node locations of the truss, determines whether the truss is in collision with itself. It considers actuator limits, member-member collision, and minimum angle constraints at the nodes. Environmental contacts can also be considered.

We used this collision checker to test traditional probabilistic planners, testing our own custom implementations of PRM (Probabilistic Roadmap) and RRT (Rapidly Exploring Random Trees). We have also integrated this collision checker with OMPL (Open Motion Planning Library) to test with state-of-the-art implementations of probabilistic path planners.

The techniques discussed thus far are sampling-based methods, which can have difficulty with high-dimensional configuration spaces. For an 18 member truss, the dimension of the configuration space is at minimum 15 (and up to 24 depending on topology and environmental constraints), so even the smallest reconfigurable trusses have large configuration spaces. To avoid this issue, we later developed alternative methods for path planning. We have developed an analytic collision checker for straight line motions which gives the exact minimum distance or time of collision for straight line motions between configurations, considering member-member collisions and minimum angles. We have also investigated cell decomposition methods for path planning in lower dimensional subspaces of the configuration space, which we will discuss in detail later in this section.

The point-based collision checker is implemented with fast and efficient performance, but the high dimensionality of the configuration space means probabilistic methods are not feasible for long paths in the full configuration space. We have obtained good results for short paths and lower-dimensional projections of the configuration space. The other methods presented in this section are able to solve more challenging planning queries.

Grid-Space Based Planner

The variable topology truss system is able to adapt itself with respect to different environments and tasks, including changing its overall geometry and topology. This year, we have continued looking into its reconfiguration ability and developing motion planning algorithms for reconfiguration tasks. Inspired by the DNA replication process, we developed a motion planning framework to enable nodes to navigate in a variable topology truss. During DNA replication processes, the topology of DNA can be changed by cutting and resealing strands as tanglements form. In a variable topology truss, a node connected with more than six members can be split into two nodes and two separate nodes can also merge into a single one shown in Figure 3.8.

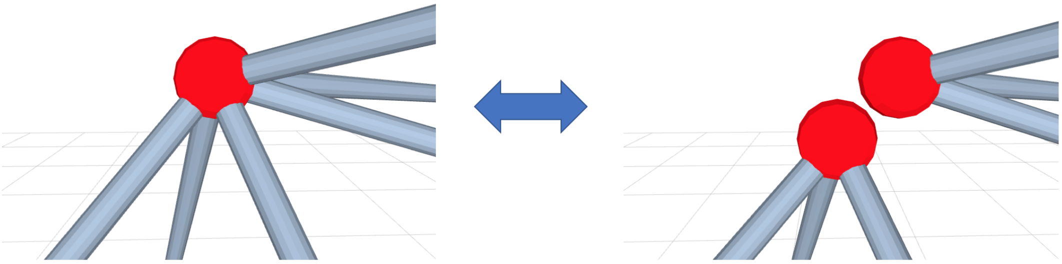


Figure 3.8: A single node with at least six members can be split into a pair of nodes and two separate nodes can also merge into a single node.

In a variable topology truss, the position of a node can be changed by both geometry reconfiguration actions and topology reconfiguration actions. Three possible atomic actions can be applied on a single node: Move, Merge and Split. For some simple reconfiguration tasks, changing the position of a node can be achieved by moving the corresponding node directly to the destination. While for some tasks, there is no way to move a node directly to the destination because the destination is outside the reachable workspace of this node. One example is shown in Figure 3.9. However, these tasks can be accomplished if we change the topology of the truss during the motion. Similar to DNA replication, a node can be temporarily split into two or more nodes so that each of them can be moved to go around the obstacle and then merge them into one single node.

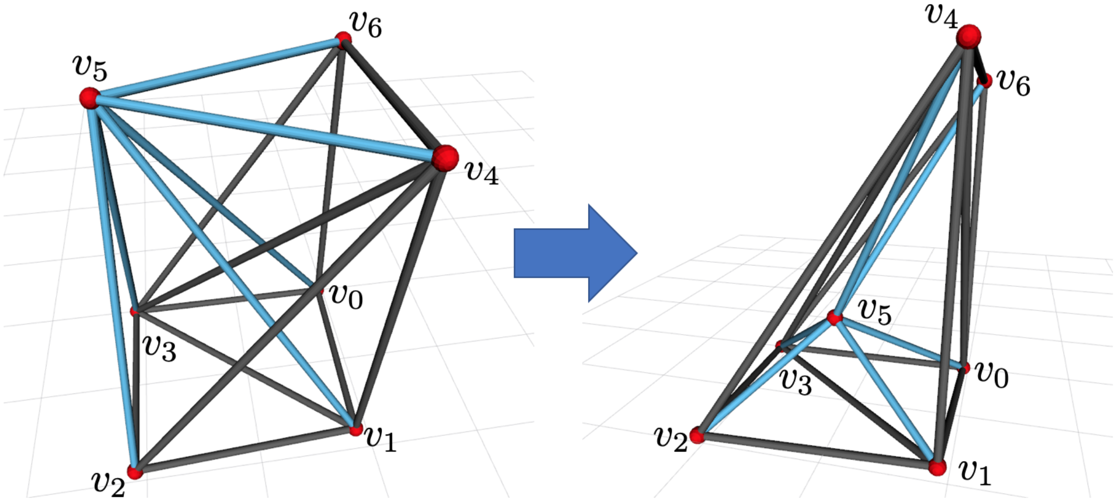


Figure 3.9: Moving node v_5 from an initial position to a goal position is impossible with only geometry reconfiguration actions because members connected with node v_5 will collide with member $\{v_3, v_4\}$.

Planning in discrete space makes it possible to efficiently plan a sequence of discrete actions for a complicated motion task. The variable topology truss system has a very high dimensional configuration space; an unsuitable space discretization can require a very large amount of memory which makes the planning problem intractable. We developed an approach to discretize the space nonuniformly with respect to the density, and different parts of the space may have different resolution. With this approach, a truss can be represented in a grid space with much fewer configurations.

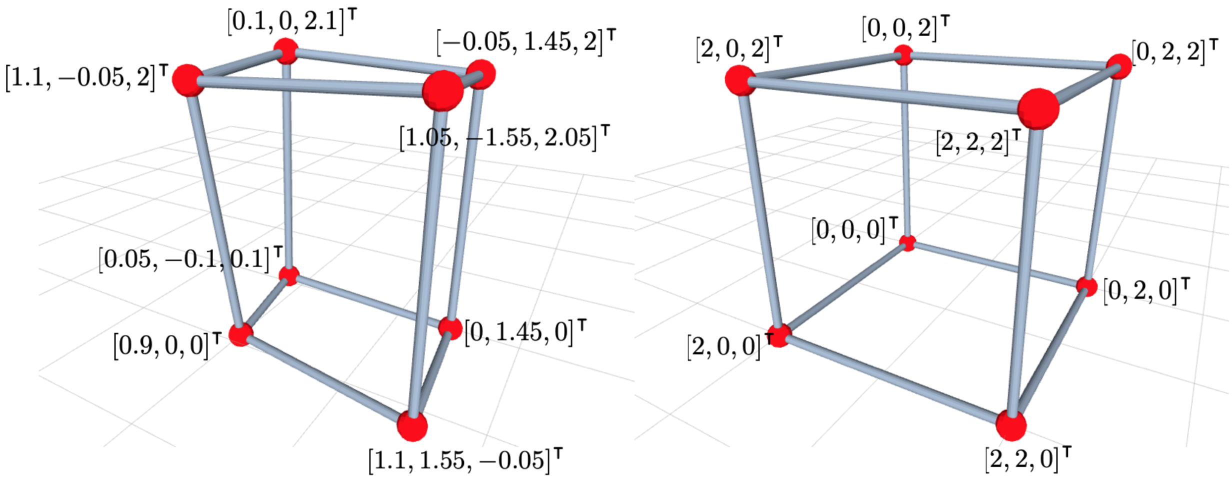


Figure 3.10: A truss in Cartesian space is shown on the left and the equivalent cubic truss in grid space is shown on the right.

Collision-free motion is a critical requirement for this motion planning, so we developed a new efficient algorithm to test straight-line collision. When moving a node, every involved member will sweep a triangle area shown in Figure 3.11. If any other member $\bar{e}$ collides with this moving member $e$during the motion, then $\bar{e}$ will intersect with the triangle swept by e. The Möller-Trumbore ray-triangle intersection algorithm is used here to check the intersection.

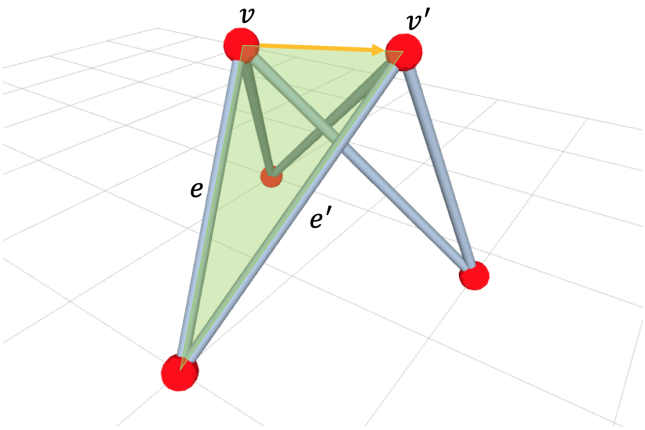


Figure 3.11: The light green triangle is swept by member e when moving node v to the new location v' along the yellow trajectory and the new member is e'.

A graph search motion planning algorithm based on A^* is used to generate a sequence of discrete actions for a given task. One example is shown in Figure 3.12. We first move the node to a closer location and then split it into a pair of nodes. Then, we move each of them in different directions to bypass the obstacle member and merge them in the end.

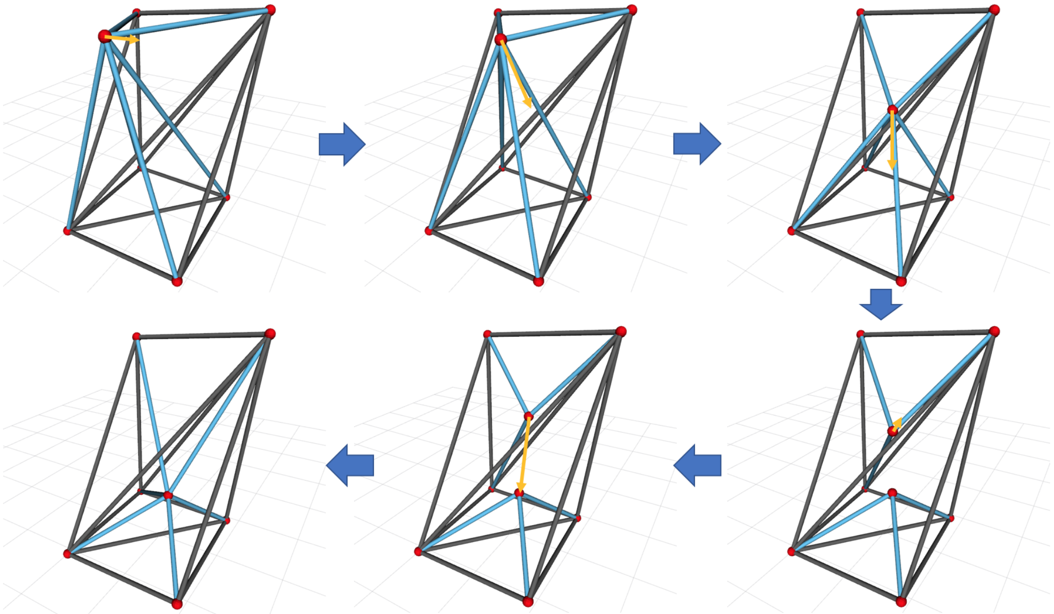


Figure 3.12: The sequence to change the position of a node connected with all blue members is shown and the motion directions are denoted by the yellow arrow.

This grid-space based motion planner can decrease the search space so as to generate a sequence of efficient discrete actions efficiently, and topology reconfiguration actions (Merge and Split) and geometry reconfiguration action (Move) are all embedded in the discrete actions. In this framework, we do need to design a good heuristic function to speed up the searching process.

Configuration Space of Nodes

As mentioned, a VTT is fully represented as an undirected graph $G = (V, E)$ where $V$is the set of vertices of $G$and $E$is the set of edges of G: each member can be regarded as an undirected labeled edge $e \in E$ of the graph and every intersection among members is a reconfigurable node $v \in V$. The configuration space of a single node $v$is $\mathbb{R}^3$ and the state of a member can be defined by the positions of its two nodes. The configuration space of a single node is composed of its obstacle region and free space. We proved that the obstacle region of a given node can be fully defined by the states of all not connected members, and this region is constructed by all obstacle polygons generated by these members. For example, in Figure 3.13, node v_0 cannot move to the region defined by one of its neighbors node v_1 and a member (v_6, v_8) or collision must happen, and the whole obstacle region is composed of multiple polygons derived in a similar way.

It is shown that the free space of a single node is usually separated by its obstacle region. We provide a boundary search algorithm to find the enclosed free subspace containing the current node position out of these obstacle polygons. We first check the intersection of all polygons to

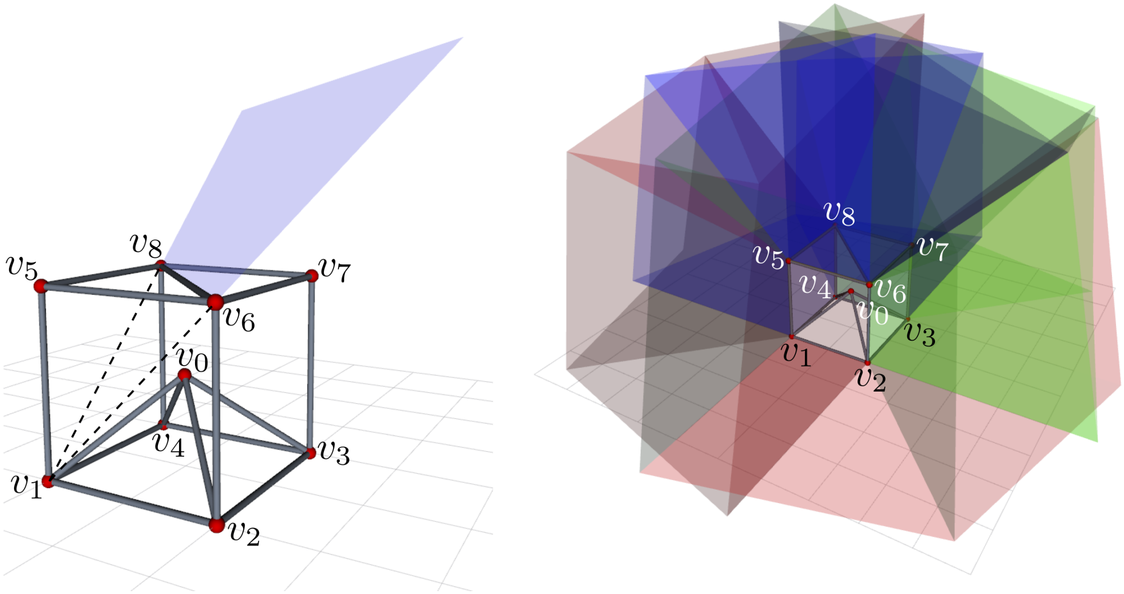


Figure 3.13: Left: Given node v_0, one of its neighbors v_1 and a member (v_6, v_8) can define the blue polygon which is part of the obstacle regions of node v_0. Right: The obstacle region of node v_0.

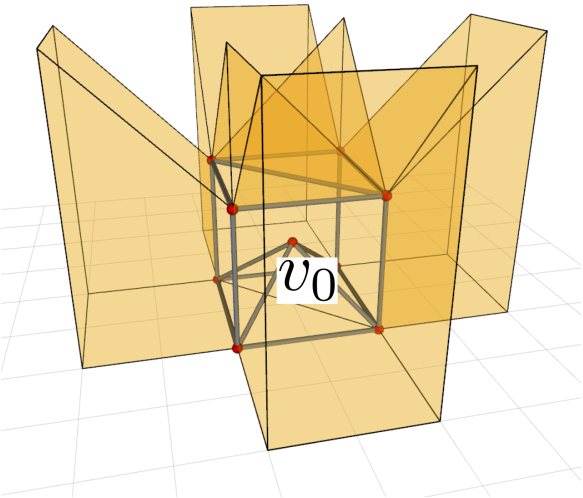


Figure 3.14: The enclosed free space containing the current position of node v_0.

generate a new set of polygons, then search the boundary starting from the closest polygon efficiently. Figure 3.14 shows the resulted enclosed free space of node v_0 and this node can move freely without collision inside this space as long as its path is fully inside this space.

The configuration space of a node is highly affected by these not connected members, resulting in a strongly coupled relationship with other nodes: moving a node can significantly

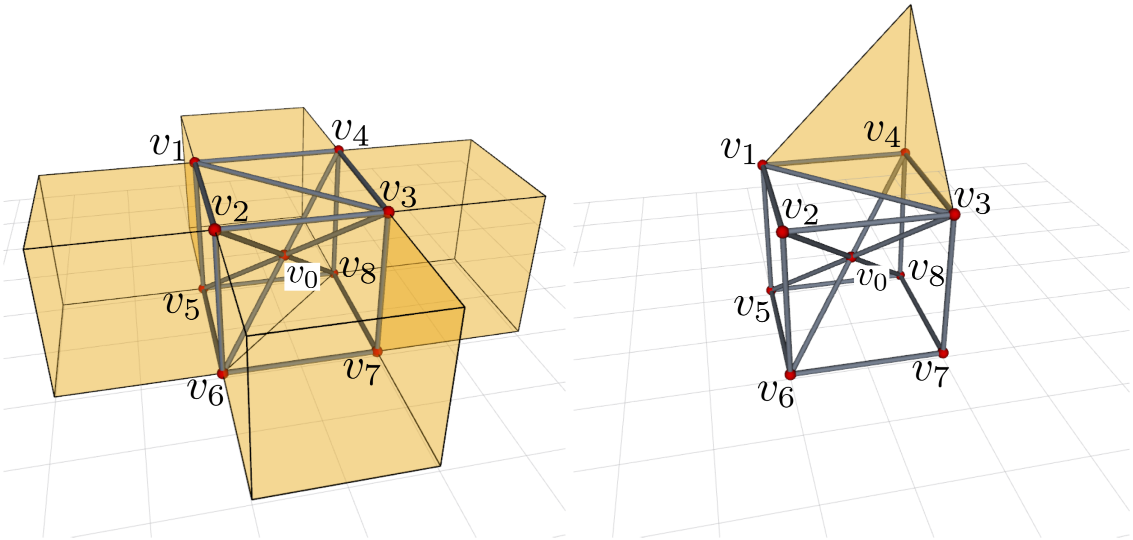


Figure 3.15: Left: The enclosed free subspace contains the current position of node v_0. Right: Another enclosed free subspace is separated by its obstacle region.

change the configuration space of other nodes. This is a challenge for geometric path planning when multiple nodes are involved.

The free space of a node is usually not connected and when this node needs to traverse from one enclosed free subspace to another, topology reconfiguration actions are needed. We can compute all the enclosed free subspace by repeatedly applying the boundary search algorithm until all polygons are included in exactly two enclosed subspaces. Figure 3.15 shows two enclosed free subspaces of node $v_0$in a simple cubic truss, and node $v_0$cannot go to this pyramid space with only geometry motion planning. In total, there are 33 enclosed free subspaces for node $v_0$above the ground.

Geometry Reconfiguration Planning

For a single node, we can compute its configuration space explicitly, including its enclosed free subspaces and obstacle region. The enclosed free subspace containing the current position of this node is usually not convex. We pass this space into a function of the Computational Geometry Algorithms Library (CGAL) to decompose the space into several convex cells. The decomposition result of the space in Figure 3.14 is shown in Figure 3.16. In each cell, the node can move freely without considering collision. With all these convex cells, a graph where all these cells are nodes and each edge represents the connection of two adjacent cells can be built easily. The cost of an edge is the distance of the trajectory of the node has to traverse from one cell to another. Then for a node, given its initial position and goal position, a path can be found using Dijkstra Algorithm. For example, the planned path for node v_0 is shown in Figure 3.16. We also proved that this planning algorithm is complete.

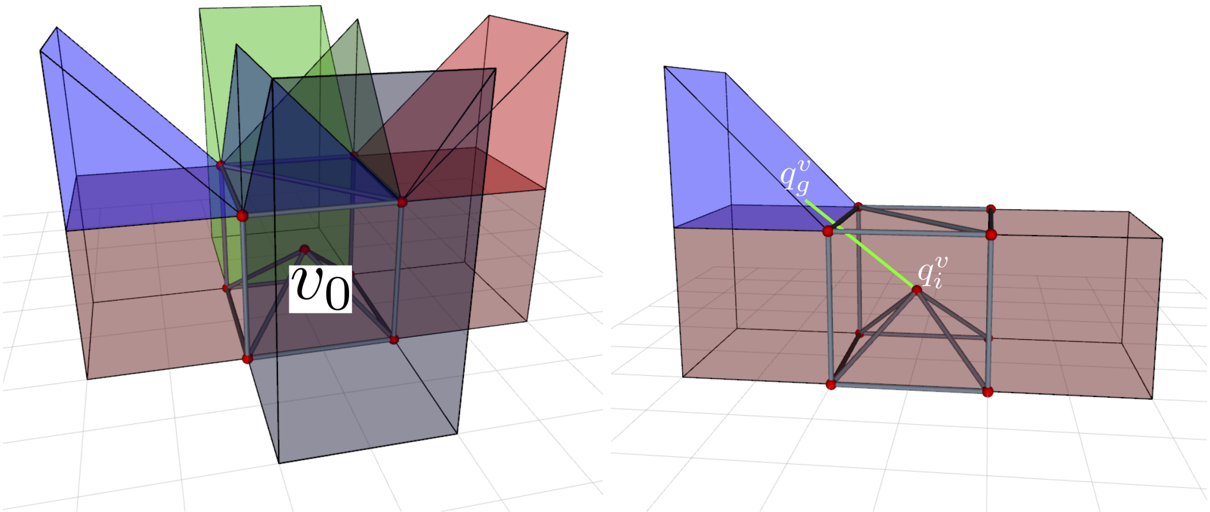


Figure 3.16: Left: The enclosed free subspace containing the current position of node v_0 is decomposed into multiple convex polyhedrons. Right: The path planned for v_0 is shown as the green line as well as the involved cells.

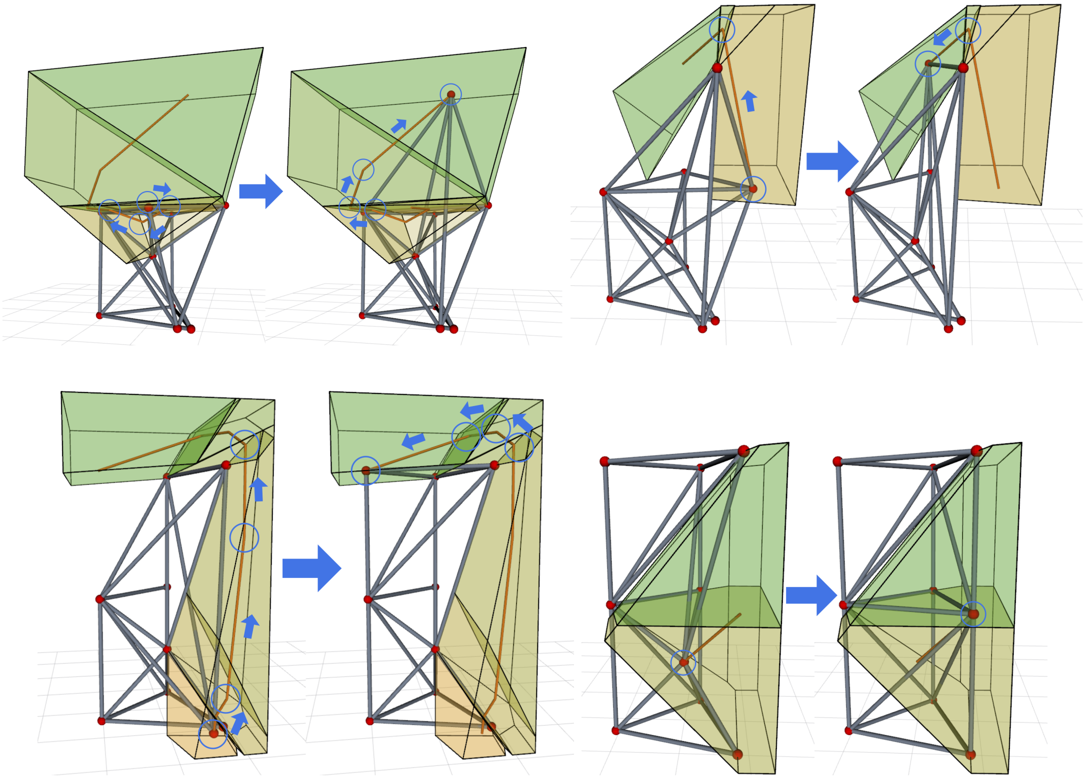


Figure 3.17: Four nodes are moved sequentially in order to change the shape of a VTT from a cubic truss to a tower.

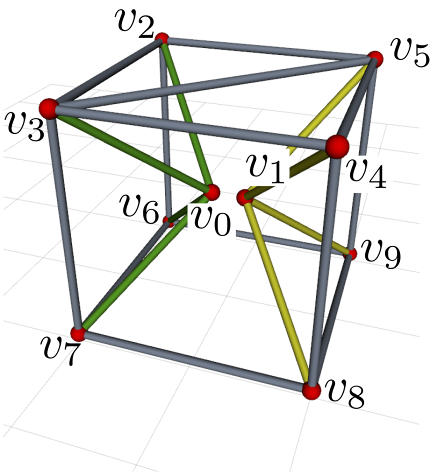


Figure 3.18: When computing the configuration space of a group of nodes containing v_0 and v_1, all moving members (green members and yellow members) are ignored.

When changing the shape of a VTT, multiple nodes can be involved. We have shown that this shape morphing problem can be solved efficiently by planning the motion of nodes one by one. One example is shown in Figure 3.17 and there are four nodes involved in this task. All our algorithms are implemented in C++ and the total planning time for this task is around 4 seconds.

We want to improve the efficiency by moving multiple nodes at the same time. This is hard because motion of nodes are strongly coupled. We extend the algorithm to compute the free space of a single node to a pair of nodes by ignoring the moving members. For example, when computing the group free space for node v_0 and v_1, we ignore all members (green members and yellow members) controlling their motions and apply our boundary search algorithm for these two nodes respectively to derive the group free space shown in Figure 3.19. Then for motions of these two nodes, as long as they are moving in their respective group free spaces, only self-collision in the group is possible, namely the collision can only happen among these green members and yellow members. The special case is when the two nodes in a group are connected by a member. This is an extra case when doing collision check.

We incorporated this result with sampling-based planning algorithms, and when moving these two nodes, sampling will only happen inside the group free space, and collision can be detected using our previous approach based on the ray-triangle intersection model. Our work is incorporated with Open Motion Planning Library (OMPL). The performance for the same shape-morphing task shown in Figure 3.17 is much better than our previous work. It takes around 0.86 second in average with 1000 trials for this planner. The planned result is shown in Figure 3.20.

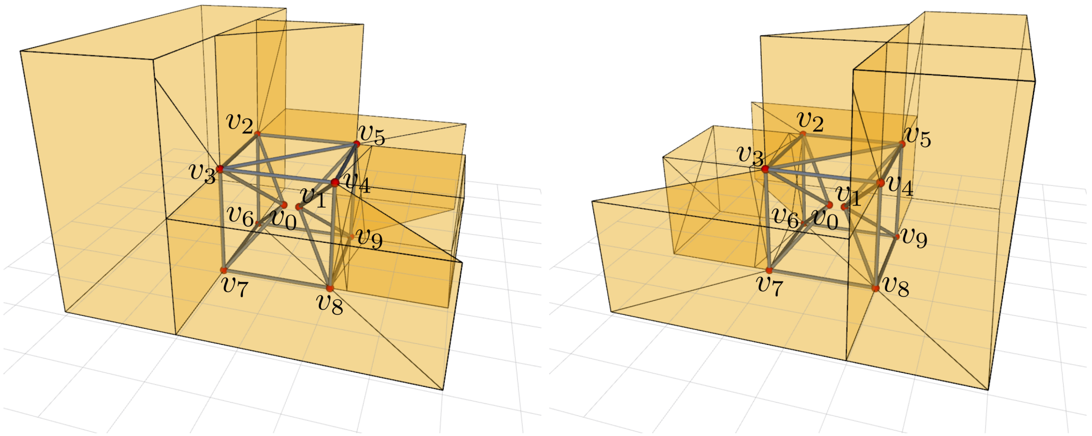


Figure 3.19: Left: The group free space of v_0 is computed with all members controlling v_1 ignored. Right: The group free space of v_1 is computed with all members controlling v_0 ignored.

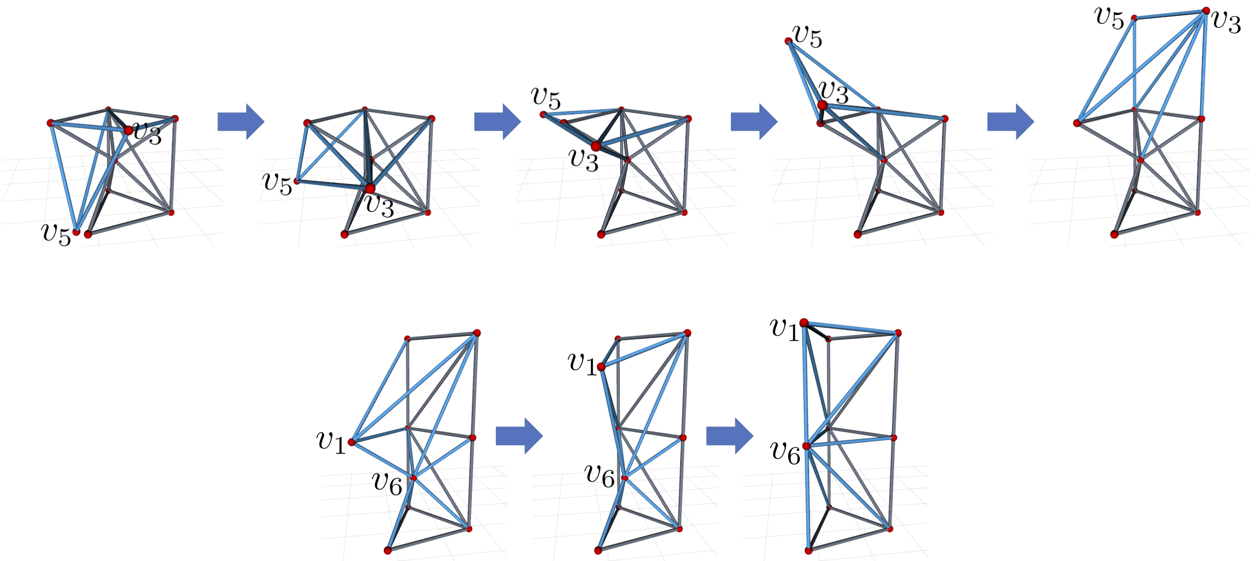


Figure 3.20: Geometry reconfiguration planning result to change a cubic truss into a tower.

Topology Reconfiguration Planning

As mentioned above, a node requires topology reconfiguration actions in order to move from one enclosed free subspace to another one. There can be multiple ways to split a node into two as there are multiple ways to take the member into two groups. Given a valid action, two new nodes can be generated, and their free spaces can be computed accordingly. If there is a position in another enclosed free subspace that is contained by both free spaces, then this

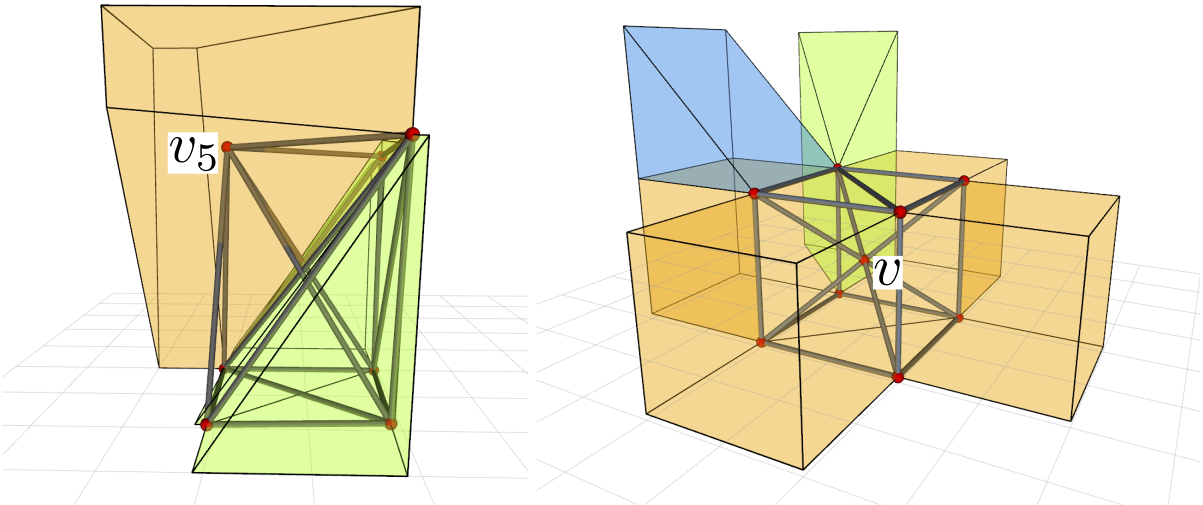


Figure 3.21: Left: The current position of node v_5 is inside the yellow space, and the goal position is inside the green space. Right: Node v has to be split and then merge in the green space, and then be split in a different way in order to merge in the blue space.

two enclosed free subspaces are connected under this action. This is our transition model for this node from one enclosed free subspace to another one. Given the goal position for a node, we can explore its enclosed free subspaces to find a sequence of topology reconfiguration actions that can move this node to the goal position. After splitting the node into two, the previous geometry reconfiguration planning approach is used to plan the motions of these newly generated nodes to go to an adjacent subspace. We can solve the task in Figure 3.12 much faster (around 0.39 second) and two of the involved enclosed free subspaces are shown in Figure 3.21. Another more complex task is also shown in Figure 3.21 that is to move node v from the yellow space to the blue space, and the whole process is shown in Figure 3.22.

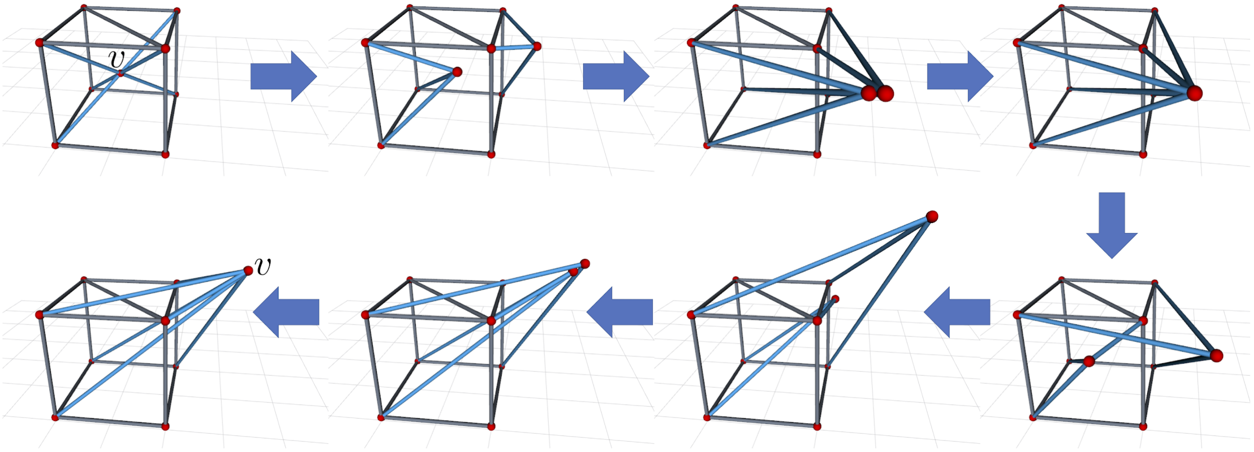


Figure 3.22: The sequence to move v from its initial position to a goal position by traversing three enclosed free subspaces.

3.4 Locomotion Planning

Locomotion is critical for performing search and rescue operations. We developed a polygon-based random tree search algorithm (PRT) for locomotion planning of VTT. This section presents the PRT algorithm and expansion of the algorithm to nonplanar terrain.

Polygon-based random tree search algorithm (PRT)

A polygon-based random tree search algorithm (PRT) provides stable locomotion planning for VTT. Previously, we performed locomotion by making the VTT's center of mass follow the given trajectory. However, we did not consider maintaining a good configuration of the VTT during locomotion, so the VTT's shape was often twisted. Large distortion is not good for VTT locomotion since it is likely to cause hardware constraint violations. Our new algorithm finds a path of desired support polygons to reach the goal, which prevents large distortion during locomotion. The support polygon is the convex polygon made from the nodes on the ground. We selected an octahedron topology for locomotion, because it has symmetry. Figure 3.23 shows the locomotion topology and definition of a support polygon.

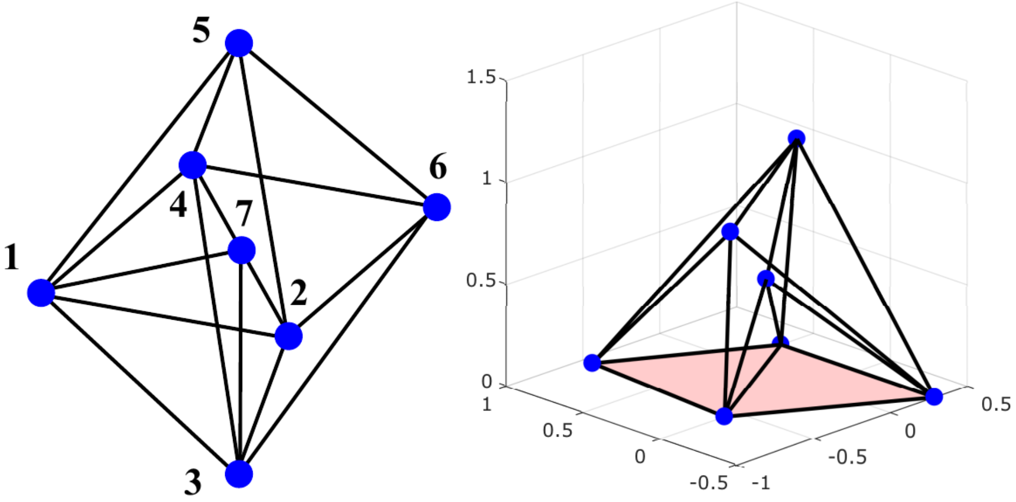
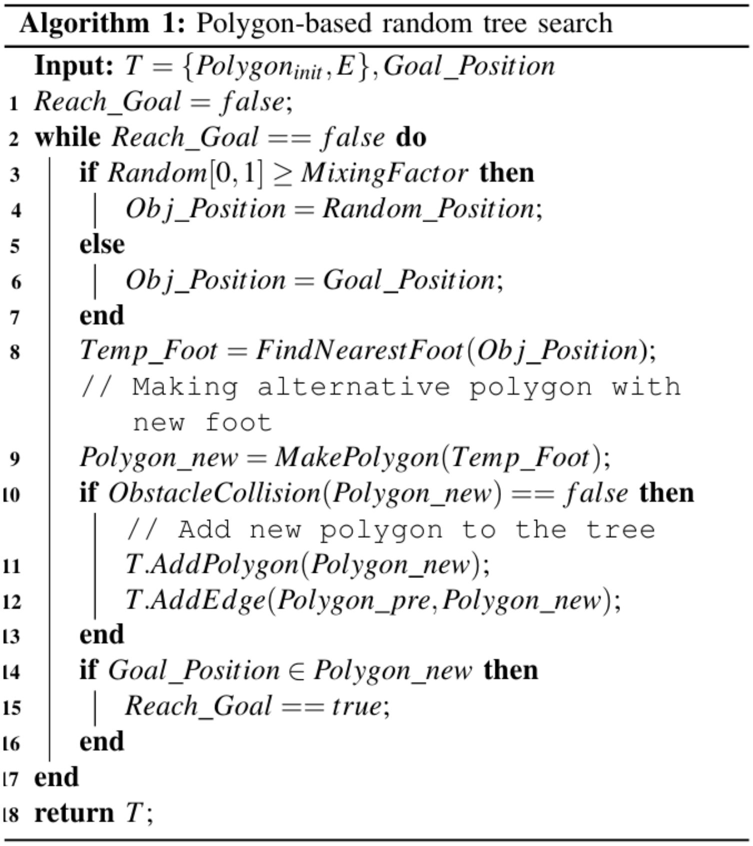


Figure 3.23. Locomotion topology of VTT. (left) Octahedron topology with internal members; (right) Definition of a support polygon.

To determine the desired trajectory to reach the destination point, a rapid random tree search algorithm was modified. PRT identifies a set of desired Polygons by randomly expanding the polygon tree. Nodes in the tree comprise Polygons, which contain Feet that construct the Polygon. A Foot is a positional vector of one vertex that belongs to a Polygon. In the case of an octahedral VTT, the shape of a support polygon is a triangle. Therefore, three Feet are contained in a Polygon.

Algorithm 1 shows the overall algorithm for PRT. For each cycle, the objective point of the algorithm is randomly selected within the operational region. To ensure the algorithm converges well, the objective point is sometimes set as the goal position with a designated probability, which is denoted as MixingFactor in line 3 of Algorithm 1.

The new Polygon is expanded from the previous Polygon sharing two Feet. Figure 3.24 shows the polygon tree expansion schematic. First, the three alternatives of the next Foot are placed around the previous Polygon. The distances between a Foot alternative and two Feet of the previous Polygon that are connected to it are set as the same distance, L_nominal. L_nominal represents the initial member length. The nearest Foot alternative from the objective point, which is randomly determined during the previous step, is selected to set the Foot of the next Polygon. The PRT provides some flexibility to setting the next Polygon by permitting a slight distortion within the distortion margin, which is denoted as a circle in Figure 3.24. The next Foot, which is drawn as a red dot in Figure 4.24, is in the closest position within the distortion margin circle.



Algorithm 1: Polygon-based random tree search

```
   Input: $T = \{Polygon_{init}, E\}, Goal\_Position$
 1 $Reach\_Goal = false$;
 2 while $Reach\_Goal == false$ do
 3     if $Random[0,1] \geq MixingFactor$ then
 4         $Obj\_Position = Random\_Position$;
 5     else
 6         $Obj\_Position = Goal\_Position$;
 7     end
 8     $Temp\_Foot = FindNearestFoot(Obj\_Position)$;
       // Making alternative polygon with
          new foot
 9     $Polygon\_new = MakePolygon(Temp\_Foot)$;
10     if $ObstacleCollision(Polygon\_new) == false$ then
           // Add new polygon to the tree
11         $T.AddPolygon(Polygon\_new)$;
12         $T.AddEdge(Polygon\_pre, Polygon\_new)$;
13     end
14     if $Goal\_Position \in Polygon\_new$ then
15         $Reach\_Goal == true$;
16     end
17 end
18 return $T$;
```

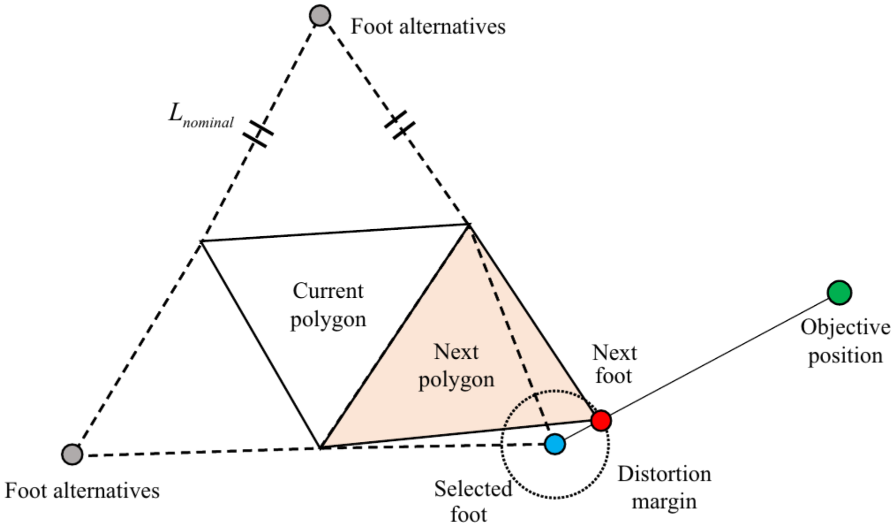
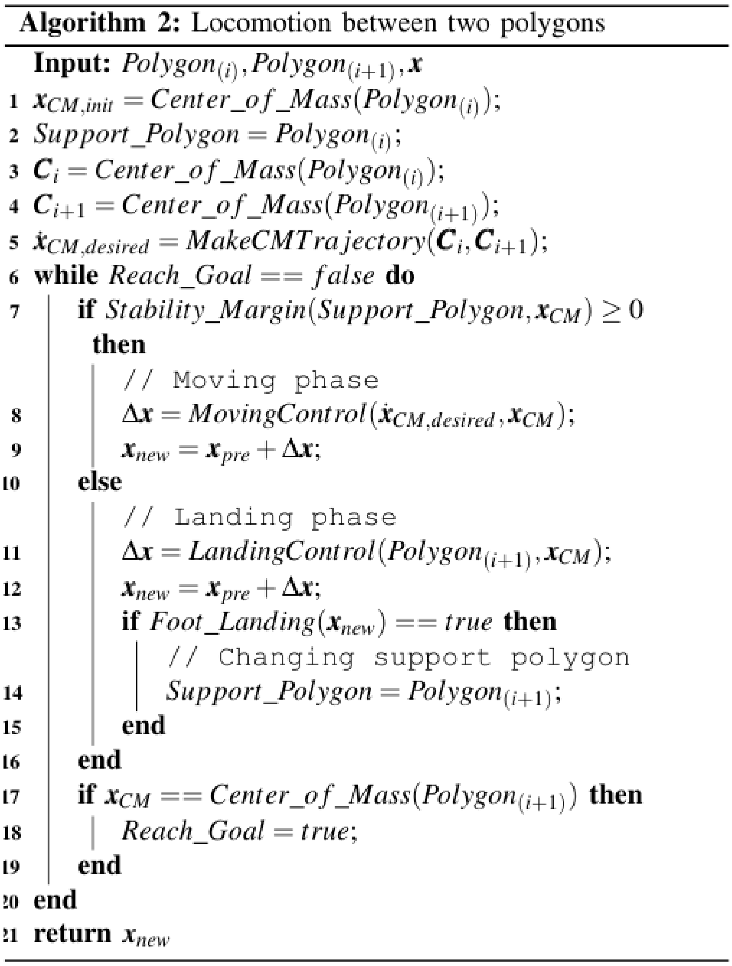


Figure 3.24. Algorithm for finding the next support polygon from the current one.

After the desired set of support polygons are derived using the PRT algorithm, a locomotion algorithm ensures the VTT can follow the derived trajectory. This locomotion algorithm was developed from the concept of our previous VTT locomotion algorithm. The proposed locomotion algorithm provides a non-impact rolling motion that prevents dynamic tip-over motion while following the support polygon trajectory. The overall locomotion algorithm is provided in Algorithm 2.



Algorithm 2: Locomotion between two polygons

```
Input: $Polygon_{(i)}, Polygon_{(i+1)}, \boldsymbol{x}$
$x_{CM,init} = Center\_of\_Mass(Polygon_{(i)})$;
$Support\_Polygon = Polygon_{(i)}$;
$\boldsymbol{C}_i = Center\_of\_Mass(Polygon_{(i)})$;
$\boldsymbol{C}_{i+1} = Center\_of\_Mass(Polygon_{(i+1)})$;
$\dot{x}_{CM,desired} = MakeCMTrajectory(\boldsymbol{C}_i, \boldsymbol{C}_{i+1})$;
while $Reach\_Goal == false$ do
    if $Stability\_Margin(Support\_Polygon, \boldsymbol{x}_{CM}) \geq 0$
     then
        // Moving phase
        $\Delta\boldsymbol{x} = MovingControl(\dot{\boldsymbol{x}}_{CM,desired}, \boldsymbol{x}_{CM})$;
        $\boldsymbol{x}_{new} = \boldsymbol{x}_{pre} + \Delta\boldsymbol{x}$;
    else
        // Landing phase
        $\Delta\boldsymbol{x} = LandingControl(Polygon_{(i+1)}, x_{CM})$;
        $\boldsymbol{x}_{new} = \boldsymbol{x}_{pre} + \Delta\boldsymbol{x}$;
        if $Foot\_Landing(\boldsymbol{x}_{new}) == true$ then
            // Changing support polygon
            $Support\_Polygon = Polygon_{(i+1)}$;
        end
    end
    if $x_{CM} == Center\_of\_Mass(Polygon_{(i+1)})$ then
        $Reach\_Goal = true$;
    end
end
return $x_{new}$
```

This locomotion strategy achieves non-impact rolling by dividing the motion into two phases, a moving phase and landing phase. During the moving phase, the VTT center of mass moves through the desired trajectory, staying inside the support polygon. The VTT is operated with the moving phase until the projected center of mass is inside the support polygon. When the center of mass is on the border of the support polygon, the VTT is controlled with the landing phase.

During the landing phase, the VTT lands the frontal node on the ground without changing the projected position of the center of mass. The frontal node is controlled to land on the vertex of the next desired support polygon. After the frontal node lands on the ground, the support polygon changes, and the VTT is controlled with the moving phase again. Figure 3.25 shows the two locomotion algorithm phases.

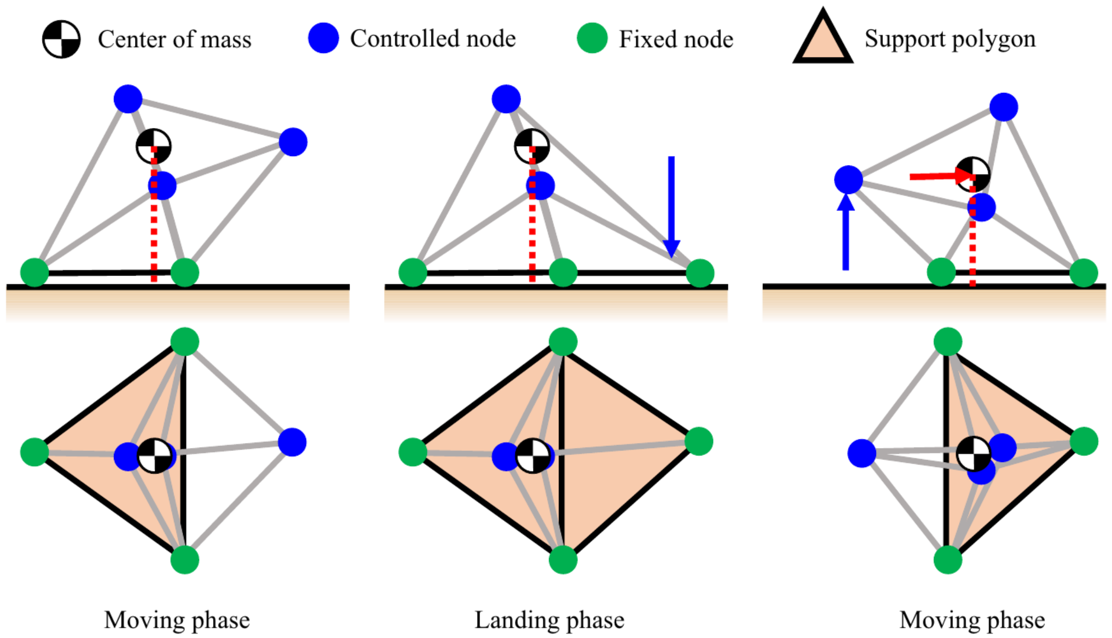


Figure 3.25. Nonimpact locomotion planning of VTT to follow desired support polygon.

We tested the proposed PRT algorithm with a MATLAB simulation, and compared it to the previous locomotion algorithm. Figure 3.26 shows the support polygon planning in various environments with obstacles. By applying the PRT algorithm, we greatly improved the success ratio of locomotion planning. Our previous algorithm finds the center of mass trajectory with a rapid random tree search algorithm. Because our algorithms are based on random search, the success ratio is not 100%. We greatly improved the success ratio with the PRT algorithm by reducing constraint violations. Figure 3.27 shows the difference between the previous algorithm and the PRT algorithm. Distortion is significantly reduced when using the PRT algorithm.

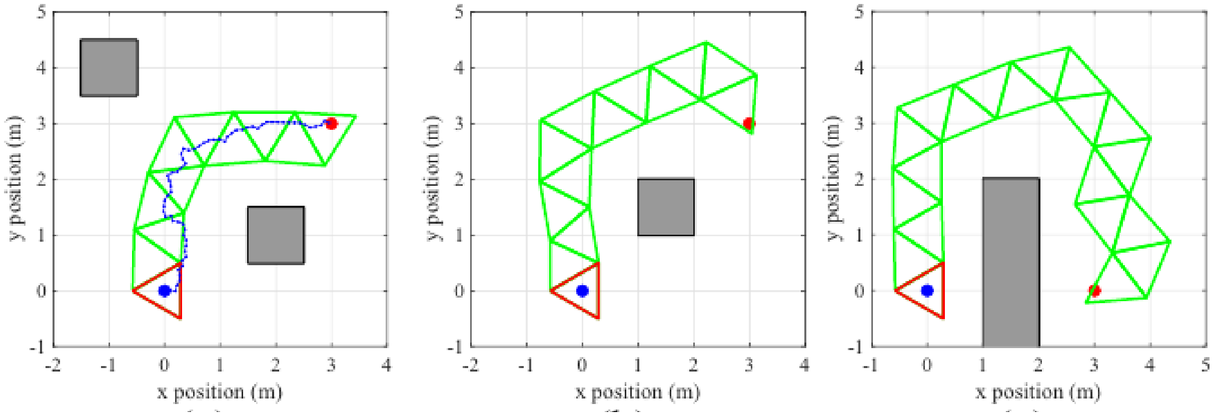


Figure 3.26. Support polygon trajectory planning on various obstacle conditions. (left) Two obstacles on both sides; (b) One obstacle in the middle; (c) One long obstacle in one side.

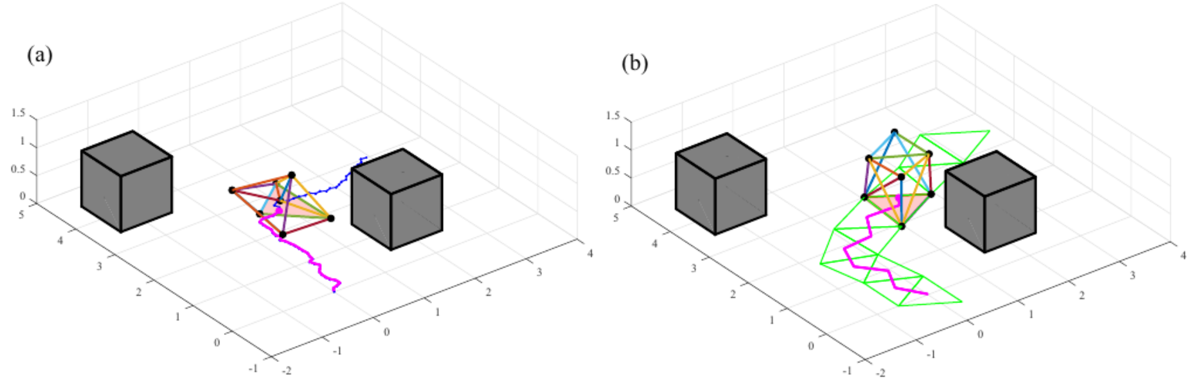


Figure 3.27. Comparison of previous locomotion algorithm and proposed PRT algorithm; (a) Locomotion simulation with the previous algorithm; (b) Locomotion simulation with PRT.

Expansion of PRT to 3-dimensional terrain

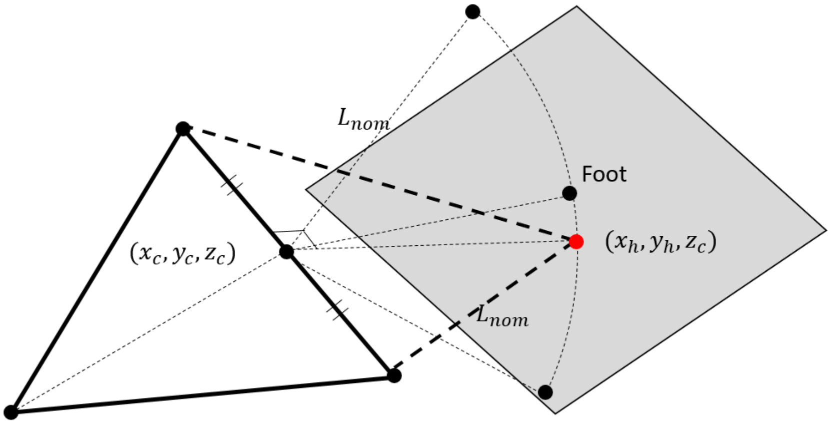


Figure 3.28. Support polygon searching on nonplanar terrain.

The original PRT algorithm only considered support polygon planning on planar terrain with obstacles. However, the object of VTT is performing search and rescue operations, the workspace of which is mostly irregular terrain. We expanded the PRT algorithm to be used on irregular terrain. Figure 3.28 shows the support polygon searching on 3-dimensional terrain. It is similar to the previous one, but finds the next foot of support polygon with a collision checking algorithm. We also consider the friction forces with the ground to avoid slipping. We developed an algorithm for calculating internal force and required frictions. Figure 3.29 shows the force calculation result of VTT on a concave surface. With this expansion, we succeeded to perform locomotion planning on uneven terrain. Figure 3.30 shows the support polygon planning on two examples of uneven terrain, and Figure 3.31 shows the locomotion simulation on two environments.

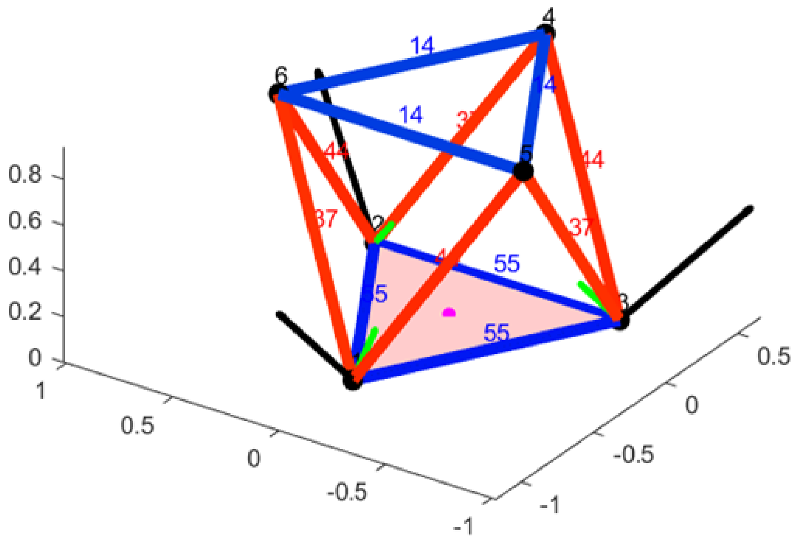


Figure 3.29. Internal force and required friction calculation diagram of VTT.

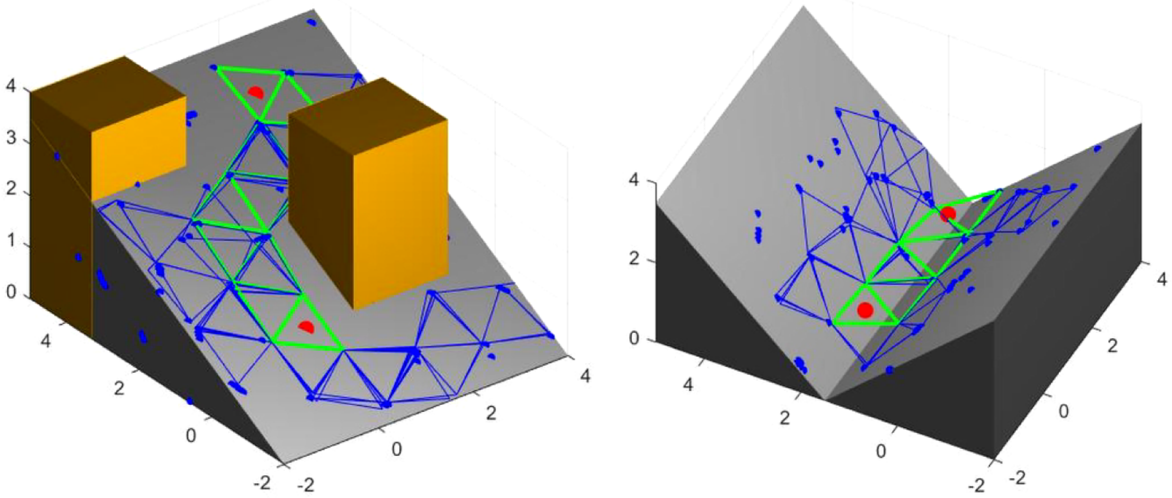


Figure 3.30. Support polygon path finding with expanded PRT algorithm; (a) Inclined terrain with obstacles; (b) V-shaped valley.

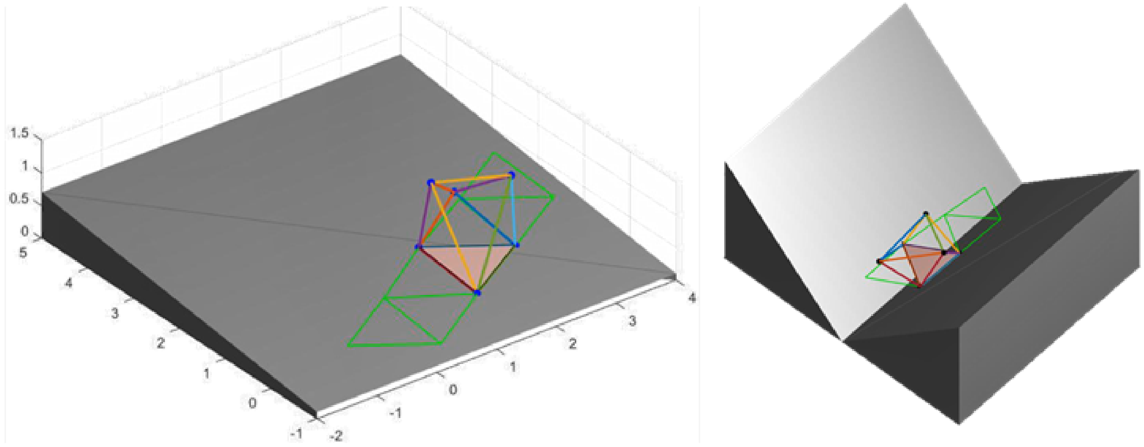


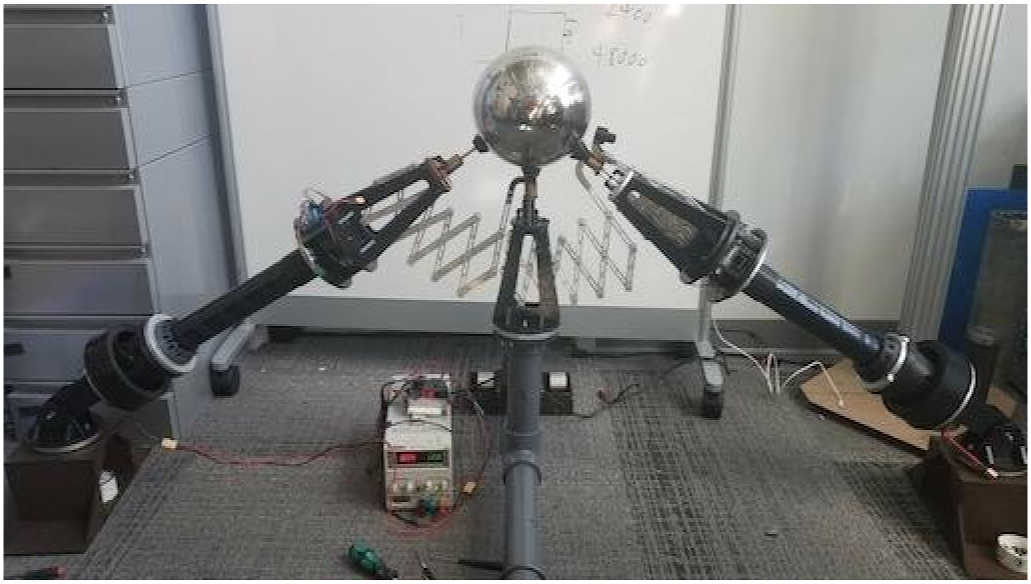
Figure 3.31. Locomotion simulation on two nonplanar terrain; (a) Inclined terrain; (b) V-shaped valley.

4 VTT Experiments and Results

In this section, we present the system-level experiments and results. These are experiments with multiple edge modules operating together to demonstrate our ability to control a truss system.

The main result we were pursuing was a demonstration of octahedral rolling locomotion. The details of this trajectory are discussed further in the Locomotion Planning section.

4.1 Overconstrained Motion Test



Date: 2019 June to August

The planned method for VTT reconfiguration involves merging and splitting nodes in the truss. If the truss is normally statically determinate, then after merging two nodes the truss will be overconstrained. In this test, we investigate the ability to control multiple modules in an overconstrained truss.

We assembled two members together in a setup with one degree of freedom. The node is attached to a hinge that allows it to move on a circular arc. This experiment uses no direct member length feedback, only Vicon feedback on the node position and incremental encoders on the motors. When we tried to make the node follow a trajectory, the system failed. The motions of the individual members were not coordinated enough, causing large internal stresses and breaking the VTT joints.

As a result, we decided to only operate the truss while it is in a statically determinate configuration. It is possible to achieve this by immediately performing node splitting after the node merge. This doesn't impact the achievable truss topologies, but it does place some limitations on the precise sequence of reconfiguration actions.

4.2 Tetrahedron with Fixed Base

Date: 2019 September
This was a basic test to demonstrate the control of a statically determinate structure. We constructed a tetrahedron with a fixed base and moved the top node along various paths. Once again, feedback was provided by an external Vicon camera system.

This experiment was a success. We achieved good trajectory-following performance for both straight line and circular paths.

4.3 Wall Tetrahedron, Two on Ground

Date: 2019 November

Although the previous test demonstrated good performance in a tetrahedral configuration, the forces involved were relatively small, and they were entirely compressive forces. We designed this test to verify the performance of the edge module under tensile loads and test the tension cable system. The setup is intended to be similar to the lifting phase during octahedral rolling.

In this test, one module retracts to lift up two fixed length modules. The module was manually controlled. We tested various levels of internal cable tension and node position to create different internal forces. The experiment was a success. We found that as long as the internal tension cable was set to a tension setpoint about 10N higher than the calculated theoretical tensile force in the member, the module would perform well.

4.4 Fully Actuated Tetrahedron

Date: 2019 December

In this test, six edge modules were attached together into a tetrahedral configuration. This demonstrates the whole system integrated together. This is the first untethered test without any kind of external support structure or external power source.

This test also demonstrated the full ROS software stack. The central computer controls all of the members simultaneously while tracking the location of all of the nodes in the truss. We successfully made the top node follow various trajectories in a similar fashion to the fixed base test. We did not attempt an autonomous tumbling gait because the impact could damage the modules—the strategy for octahedral motion uses a non-impact rolling gait. Instead, we manually assisted a tumbling step to allow it to move through the trajectory gently.

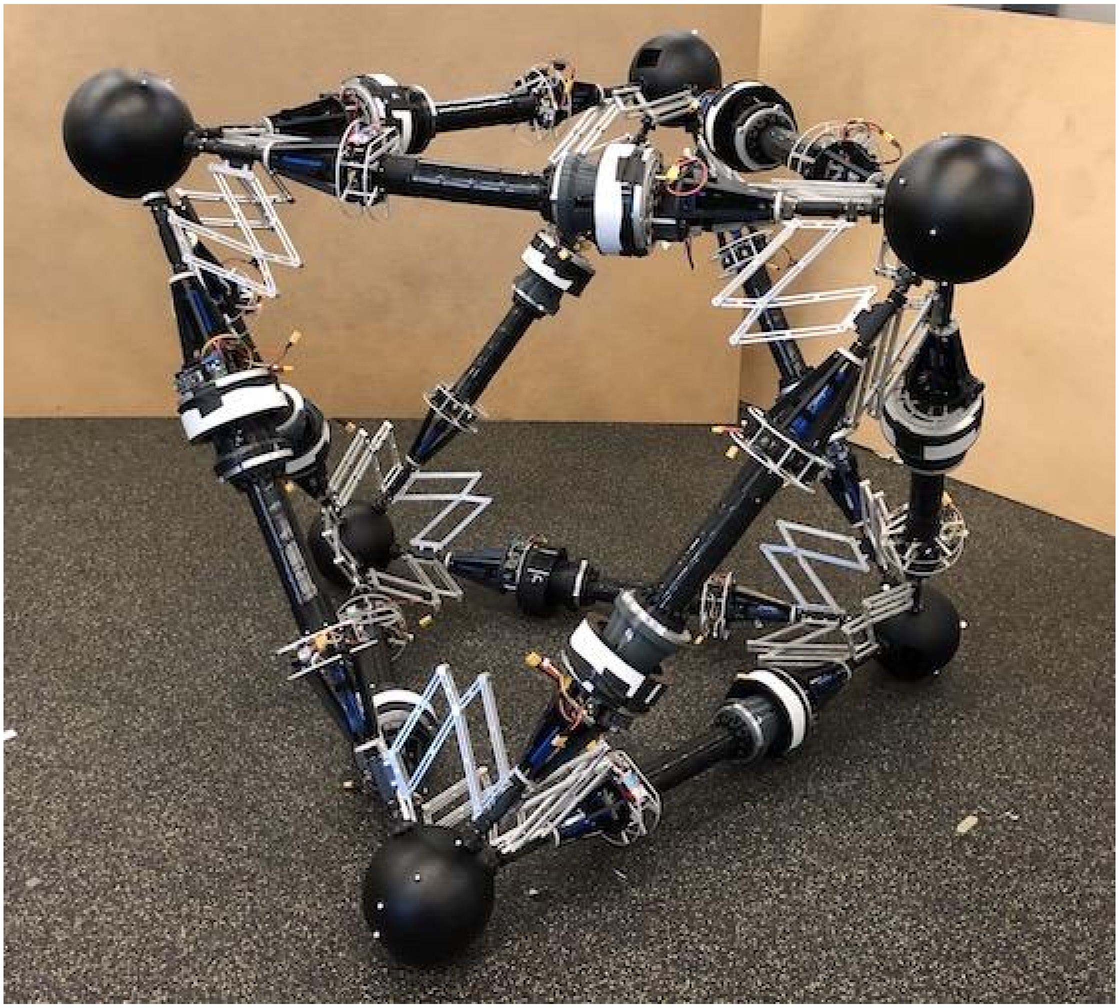
4.5 Wall Tetrahedron, One on Ground

Date: 2020 January

We performed a second wall-mounted tetrahedron test to test the ability of two edge modules to retract and lift a load in a coordinated fashion. Again, the setup is intended to be similar to lifting up one of the corners during octahedral rolling.

This test was successful. We added additional weight to the node to push the limits of the system and make the conditions as similar as possible to octahedral rolling. At the maximum weight, the system was able to lift the node while each upper member was experiencing about 80N of tension.

4.6 Fully Actuated Octahedron



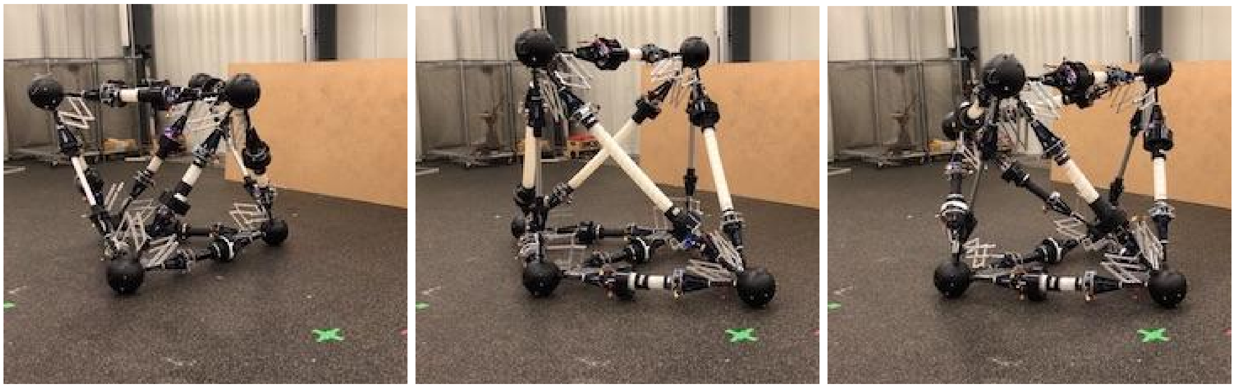
Date: 2020 February-March

In early 2020, we completed the assembly of twelve edge modules and connected them together into an octahedron to test octahedral locomotion. As usual, the robot was controlled using an external motion capture system and central ROS computer.

These experiments had mixed results. We were able to successfully move single nodes at a time. However, when we tried to control the motion of all of the top three nodes together, we had less success. Actuator dead zones lead to oscillations in the controller, which put too much internal stress on the members. Additionally, the weight of the members themselves placed larger than expected internal bending moments on the top three members. The combination of these internal forces was too much for the edge modules to bear, and the system was very fragile.

Shortly after these experiments, we were forced to leave the lab due to COVID-19 restrictions. During the working-from-home period, we evaluated the problems with the edge modules and designed solutions. These designs became the second-generation edge module that was discussed in the previous section. The main change was switching the band material from ABS to nylon. After resuming on-site research activities, we made significant progress on integrating the second-generation edge module into the system.

4.7 Octahedron Rolling Step



Date: 2020 December

As we started to manufacture the second-generation edge modules, we partially replaced the previous modules in the octahedron and continued the full-system experiments. In this test, seven of the modules that would experience high loads were replaced with the second-generation modules, and two modules that did not need to extend were replaced with fixed length aluminum beams.

In this test, we successfully completed a single rolling step. However, our controller did not perform well with the presence of the fixed-length members, and there were still some oscillations.

4.8 Merge and Split Demo

Date: 2021 January

Using the second-generation edge modules with the twelve-bar linkage member end, we performed an improved demo of the reconfiguration mechanism. This demo builds off of the same concept as the demo presented at IROS 2017 (Fig. 2.10), but uses the reconfigurable linkages to alter the connectivity of the truss after merging, rather than simply merging and splitting back to the original connectivity. It demonstrates the reconfiguration capability of the 18-member octahedral truss with two internal tetrahedra, as shown in Fig. 3.3.

Since we lacked the time to develop an actuated version of this reconfiguration mechanism, the latching procedure and linkage angles are controlled by hand in this demo.

Publications:

Spinos, A., Carroll, D., Kientz, T., Yim, M. "Topological Reconfiguration Planning for a Variable Topology Truss." Journal of Mechanisms and Robotics 13, no. 4 (2021): 040901

Liu, C., and Yim, M., “Motion Planning for Variable Topology Truss Modular Robots” Robotics: Science and Systems to appear Feb 2021

Liu, C., Yu,S., Yim ,M. “A Fast Configuration Space Algorithm for Variable Topology Truss Modular Robots,” Proc. of the IEEE Intl. Conf. on Robotics and Automation, (ICRA), Paris France- COVID19 remote, May 31–June 4, 2020

Sumin, P., Jangho, B., Seohyeon, L., Yim, M., Jongwon K., TaeWon S., “Polygon-based Random Tree Search Planning for Variable Geometry Truss Robot,” Proc. of the IEEE Intl. Conf. on Robotics and Automation, (ICRA), Paris France- COVID19 remote, May 31–June 4, 2020

Eugene, P., Jangho, B., Sumin, P., Yim, M., Jongwon, K., TaeWon, S., “Reconfiguration Solution of a Variable Topology Truss: Design and Experiment,” Proc. of the IEEE Intl. Conf. on Robotics and Automation, (ICRA), Paris France- COVID19 remote, May 31–June 4, 2020

Lee, S.H., Carroll,D., Spinos,A.,Yim,M., Seo,T.W., “The Design and Experiment of the Multi-Armed Friction Drive Wheel of the Spiral Zipper for Friction Adjustment,” In 2020 17th International Conference on Ubiquitous Robots (UR). Kyoto Japan COVID19-remote, June 24–27, 2020

Liu, C., and Yim, M., “Reconfiguration Motion Planning for Variable Topology Truss,” Proc. of the IEEE/RSJ Intelligent and Systems 2019, 1941-1948, Macau, Nov. 4–8, 2019

Sumin P., Eugene P., Yim, M., Jongwon K., TaeWon S., “Optimization-Based Nonimpact Rolling Locomotion of a Variable Geometry Truss,” IEEE Robotics and Automation Letters, Vol. 4, No. 2, Apr 2019

Liu, C., Yu, S., and Yim, M., “Shape Morphing for Variable Topology Truss,” In 2019 16th International Conference on Ubiquitous Robots (UR). Jeju Korea, June 24–27, 2019

Jeong, S., Kim, B., Park, S., Park, E., Spinos, A., Carroll, D., ... & Park, F. C. (2018, June). Variable Topology Truss: Hardware Overview, Reconfiguration Planning and Locomotion. In *2018 15th International Conference on Ubiquitous Robots (UR)* (pp. 610-615). IEEE.

D. Carroll and M. Yim, “Truss optimization with random piecewise linear beams,” in 2017 asme international mechanical engineering congress & exposition (IMECE), Tampa, Florida, 2017.

A. Spinos and M. Yim, “Towards a variable topology truss for shoring,” in Ubiquitous robots and ambient intelligence (urai), 2017 14th international conference on, 2017, pp. 244-249.

A. Spinos, D. Carroll, T. Kientz, and M. Yim, “Variable topology truss: design and analysis,” in 2017 ieee/rsj international conference on intelligent robots and systems (iros), Vancouver, Canada, 2017.

Weng, Y., and Yim,M. “Kinematics of Variable Topology Truss Using Affine Coordinate Transformation,” *Proc. of the 14th Intl. Conf. on Ubiquitous Robots and Ambient Intelligence* (URAI) Jeju, Korea, June 28 – July 11, 2017