공학박사학위논문

가변 토폴로지 트러스 로봇의 안정적인 주행 알고리즘 개발

Stable Rolling Locomotion for Variable Topology Truss Robot

2020년 8월

서울대학교 대학원

기계항공공학부

박 수 민

Abstract

Variable Topology Truss (VTT) is truss structured modular robot that can self-reconfigure its topology and geometric configuration, which can be usefully applied to rescuing work in disaster site. In this thesis, design of VTT is introduced and stable rolling locomotion algorithm for VTT is proposed.

To achieve self-reconfiguration feature, VTT are composed specially designed members and nodes. VTT's members consist of Spiral Zippers which are novel linear actuators that has high extension ratio, light weight and high strength. VTT's nodes consist of Passive Member-Ends and Master Member-Ends. Passive Member-Ends are linkage type spherical joint with large angle range that can accommodate many members. Master Member-Ends are spherical manipulators that built in Sphere and it move member to change topology of VTT.

Rolling locomotion of VTT is achieved by controlling the center of mass by geometric reconfiguration. However, the locomotion planning is complex problem, because VTT is parallel mechanism with high degree of freedom and many constraints, which makes it difficult to predict and avoid constraints for feasible planning. Thus, it needs stable algorithm that can find locomotion trajectory even in complicated and large environment. In addition, since VTT has many sophisticated components, the algorithm must prevent VTT being damaged from ground by tumbling.

To meet the requirements, proposed locomotion algorithm is composed of 3 steps; support polygon planning, center of mass planning and node position planning. In support polygon planning, support polygon path is planned by newly

proposed random search algorithm, Polygon-Based Random Tree (PRT). In center of mass planning, trajectory of desired projected center of mass is planned by maximizing stability feature. Planned support polygon path and center of mass trajectory guide VTT to have good-conditioned shape which configuration is far from constraints and makes locomotion planning success even in complex and large environment.

In node position planning, Non-Impact Rolling locomotion algorithm was developed to plan position of VTT's nodes that prevent damage from the ground while following planned support polygon path and center of mass trajectory.

The algorithm was verified by two case study. In case study 1, locomotion planning and simulation was performed considering actual constraints of VTT. To avoid collision between VTT and obstacle, safety space was defined and considered in support polygon planning. The result shows that VTT successfully reaches the goal while avoiding obstacles and satisfying constraints.

In case study 2, locomotion planning and simulation was performed in the environment having wide space and narrow passage. Nominal length of VTT was set to be large in wide space to move efficiently, and set to be small in narrow passage to pass through it. The result shows that VTT successfully reaches the goal while changing its nominal length in different terrain.

Keyword : modular robot, variable topology truss, variable geometry truss, locomotion planning, trajectory optimization, random search algorithm, obstacle avoidance

Student Number : 2013-20675

Table of Contents

Chapter 1. Introduction 1

1.1 Motivation 1

1.2 Previous Truss Type Modular Robot 4

1.3 Previous Research on VTT's Locomotion 8

1.3.1 Heuristic Based Methods 9

1.3.2 Optimization Based Method 10

1.4 Objectives of Locomotion Algorithm 12

1.5 Contribution of Thesis 13

1.5.1 Introduction to Hardware Design of VTT 13

1.5.2 Stable Rolling Locomotion of VTT 15

Chapter 2. Design of Variable Topology Truss 17

2.1 Member Design 18

2.1.1 Spiral Zipper 20

2.1.2 Tensioner 26

2.2 Node Design 28

2.2.1 Passive Member-End and Sphere 29

2.2.2 Master Member-End 36

2.3 Control System 40

2.4 Node Position Control Experiment 44

Chapter 3. Mathematical Model of Variable Topology Truss 47

3.1 Configuration and Terminology 47

3.2 Inverse Kinematics 50

3.3 Constraints 51

3.4 Stability Criteria 64

Chapter 4. Locomotion Algorithm 66
4.1 Concept of Locomotion Algorithm 67
4.1.1 Method for Successful Planning and Obstacle Avoidance 67
4.1.2 Method to Prevent Damage from the Ground 71
4.1.3 Step of Locomotion Algorithm 72
4.2 Support Polygon Planning 73
4.2.1 Polygon-Based Random Tree (PRT) Algorithm 73
4.2.2 Probabilistic Completeness of PRT Algorithm 79
4.3 Center of Mass Planning 85
4.4 Node Position Planning 86
4.4.1 Concept of Non-Impact Rolling Locomotion 86
4.4.2 Planning Algorithm for Non-Impact Rolling Locomotion 89
4.4.3 Optimization Problem of Moving Phase 94
4.4.4 Optimization Problem of Landing Phase 98
4.4.5 Optimization Problem of Transient Phase 99

Chapter 5. Experimental Verification 100
5.1 Case Study 1: Actual VTT Prototype 101
5.1.1 Simulation Condition 101
5.1.2 Obstacle Avoidance Method 103
5.1.3 Simulation Result 104
5.2 Case Study 2: Environment with Narrow Passage 111
5.2.1 Simulation Condition 111
5.2.2 Support Polygon Planning with Varying Nominal Length 114
5.2.3 Simulation Result 117

Chapter 6. Conclusion 126
Bibliography 129
Abstract in Korean 134

List of the Figures

Fig.1.1.1 Rolling locomotion of VTT in disaster site 2
Fig.1.1.2 VTT reconfiguration: (a) Topology reconfiguration.
(b) Geometric reconfiguration for shoring 2
Fig.1.2.1 Early concept of variable geometry truss (VGT) [11] 4
Fig.1.2.2 Tetrobot prototype [10] 5
Fig.1.2.3 Design of TET robot [14] 6
Fig.1.2.4 Design of Odin robot [15] 7
Fig.1.3.1 Heuristic method of locomotion [20] 9
Fig.1.3.2 Locomotion simulation with optimization method [23] 10
Fig.1.3.3 Locomotion algorithm by Usevitch [23] 11
Fig.1.4.1 Schematic of locomotion environment 12
Fig.1.5.1 Design of VTT 14
Fig.1.5.2 Schematic of support polygon path and center of mass trajectory 15
Fig.1.5.3 Schematic of Impact Rolling and Non-Impact Rolling 16
Fig.2.1 Overall design of VTT 17
Fig.2.1.1 Prototype of VTT member 19
Fig.2.1.2 Member design of VTT 19
Fig.2.1.3 Band of Spiral Zipper 20
Fig.2.1.4 Spiral Zipper column (a) without backing, (b) with backing 20
Fig.2.1.5 Design of Slider 21
Fig.2.1.6 Design of Friction Wheel 22
Fig.2.1.7 Design of Support Ring 23
Fig.2.1.8 Spiral Zipper in horizontal configuration (a) without Support Ring,
(b) with Support Ring 24
Fig.2.1.9 Design of Band Management System 25
Fig.2.1.10 Design of tensioner 26
Fig.2.1.11 Tensioner prototype 27
Fig.2.2.1 Passive Member-End and Sphere 28
Fig.2.2.2 Master Member-End prototype 28

Fig.2.2.3 Conventional 3 DOF joint. (a) Ball joint. (b) Universal joint 29
Fig.2.2.4 Node of Odin robot [15] 30
Fig.2.2.5 Node of TET robot [14] 30
Fig.2.2.6 Concentric multilink spherical (CMS) joint [10] 31
Fig.2.2.7 Three CMS joint connected in a row [10] 31
Fig.2.2.8 Design parameter of a CMS joint [10] 32
Fig.2.2.9 Design parameter of a Passive Member-End linkage 32
Fig.2.2.10 Design of a Passive Member-End 33
Fig.2.2.11 Passive Member-Ends with a Sphere: one at minimum angle and the other at maximum angle 33
Fig.2.2.12 Three Passive Member-end on a Sphere 34
Fig.2.2.13 Edge module 35
Fig.2.2.14 Reconfiguration example from cube topology to tower topology 36
Fig.2.2.15 Master Member-End. (a)Master Member-End prototype with Passive Member-End. (b) Master Member-End model 38
Fig.2.2.16 Design of one-way chain 39
Fig.2.2.17 Member moving test with Master Member-End 39
Fig.2.3.1 Control hierarchy of VTT system 40
Fig.2.3.2 Control diagram of VTT 42
Fig.2.3.3 VICON motion capture system. VTT in VICON system area (left). Software visualized VICON system (top right). Sphere with markers (bottom right) 43
Fig.2.3.4 Controller module of VTT 43
Fig.2.4.1 Octahedron VTT for node position control experiment 44
Fig.2.4.2 Position graph of VTT position control experiment 46
Fig.3.1.1 VTT model. (a) Mechanical model. (b) Mathematical model 48
Fig.3.1.2 Representation of octahedron topology VTT 48
Fig.3.3.1 Length of edge 51
Fig.3.3.2 Minimum and maximum angle between adjacent edges 52
Fig.3.3.3 Schematic of dihedral angle 53
Fig.3.3.4 Example of minimum distance between non-adjacent members 55
Fig.3.3.5 Applied forces at edge module 59

Fig.3.3.6 Edge module's mass and center of mass 59
Fig.3.3.7 VTT load calculation example 60
Fig.3.3.8 Test bench for measuring actuation strength 61
Fig.3.4.1 (a) Support polygon of VTT. (b) Schematic of stability margin 64
Fig.3.4.2 Schematic of allowable region of the center of mass 65
Fig.4.1.1 Locomotion simulation with optimization method by Usevitch [23] 67
Fig.4.1.2 Center of mass trajectory following locomotion simulation with octahedron topology 69
Fig.4.1.3 Maximum and minimum member lengths in locomotion simulation 69
Fig.4.1.4 Center of mass trajectory following locomotion simulation with octahedron topology 70
Fig.4.1.5 Maximum and minimum member lengths in locomotion simulation 70
Fig.4.1.6 Schematic of Impact Rolling and Non-Impact Rolling 71
Fig.4.1.7 Schematic of support polygon path and center of mass trajectory 72
Fig.4.2.1 Schematic of RRT structure 74
Fig.4.2.2 Schematic of finding next support polygon 76
Fig.4.2.3 Example of support polygon planning 77
Fig.4.2.4 PRT trees of (0,0) initial position and 0.2 distortion margin 81
Fig.4.2.5 Fig. 4.2.5. PRT trees of (-9,-9) initial position and 0.2 distortion margin 82
Fig.4.2.6 PRT trees of (0,0) initial position and 0.2 distortion margin with 15 obstacles 83
Fig.4.2.7 Probability graph of PRT for each case 84
Fig.4.3.1 Schematic of center of mass trajectory 85
Fig.4.4.1 Support polygon and stability margin 86
Fig.4.4.2 Non-Impact Rolling locomotion process between two support polygons 88
Fig.4.4.3 Schematic of moving phase 94
Fig.4.4.4 Configuration of octahedron topology with 6 nodes and 12 edges.
(a) Nominal configuration (desirable).
(b) Ill-conditioned configuration 95
Fig.4.4.5 Schematic of desired velocity of a node 96

Fig.4.4.6 Schematic of landing phase....98
Fig.4.4.7 Schematic of transient phase....99
Fig.5.1.1 Octahedron topology (12 edges, 6 nodes)....101
Fig.5.1.2 Edge module's mass and center of mass....101
Fig.5.1.3 Start position, goal position, obstacles and initial configuration....102
Fig.5.1.4 Safety space of octahedron VTT....103
Fig.5.1.5 Result of support polygon planning....105
Fig.5.1.6 Center of mass trajectory....106
Fig.5.1.7 Simulation of VTT locomotion (isometric view)....107
Fig.5.1.8 Simulation of VTT locomotion (top view)....108
Fig.5.1.9 VTT constraint graph....109
Fig.5.2.1 Octahedron topology (12 edges, 6 nodes)....111
Fig.5.2.2 Edge module's mass and center of mass....112
Fig.5.2.3 Start position, goal position, obstacles and initial configuration....113
Fig.5.2.4 Waypoints and zones in locomotion environment....114
Fig.5.2.5 PRT trees with generations....115
Fig.5.2.6 Waypoints and zones in locomotion environment....117
Fig.5.2.7 Result of support polygon planning....118
Fig.5.2.8 Center of mass trajectory....119
Fig.5.2.9 Simulation of VTT locomotion (1 sec to 1800 sec)....120
Fig.5.2.10 Simulation of VTT locomotion (2400 sec to 3400 sec)....121
Fig.5.2.11 Simulation of VTT locomotion (3600 sec to 4600 sec)....122
Fig.5.2.12 Simulation of VTT locomotion (4800 sec to 6527 sec)....123
Fig.5.2.13 VTT constraint graph....124
Fig.5.2.14 Length of edge graph....125

List of the Tables

Table 2.1 Initial position of the nodes in octahedron VTT 44

Table 4.1 Condition of cases for probabilistic completeness simulation 79

Table 4.2 Meaning of variables in Algorithm 2 91

Table 4.3 Meaning of variables in Algorithm 3 93

Table 5.1 Constraint parameter of VTT in Case study 1 102

Table 5.2 Constraint parameter of VTT in Case study 2 112

Table 5.3 Nominal lengths for each zone 116

Chapter 1

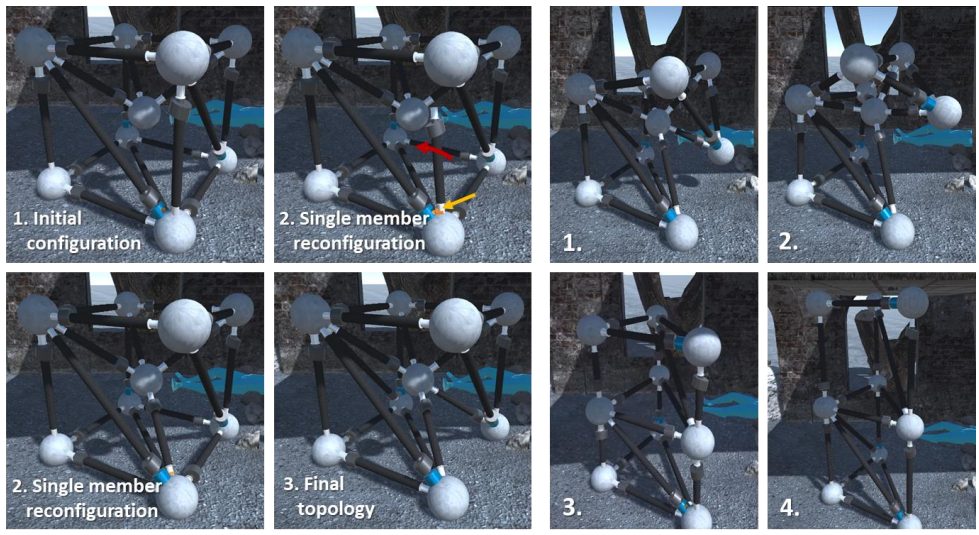
Introduction

1.1 Motivation

When typhoons, earthquake, bombings or explosion occurs, many victims are generated by building collapse. Moreover, weaken building structure can induce second collapse which can endanger both victims and rescuers. Thus, it is important to shore the collapsed building in advance, to prevent second collapse and ensure the safety of rescuers and victims. The shoring typically done by installing wood beam support structure [1] which requires much time and human forces. Variable Topology Truss (VTT) can be a good solution for this situation.

VTT is truss structured modular robot that can self-reconfigure its geometric configuration and topology [2]-[7]. Having truss structure, member of VTT is subjected to pure axial load on its member and can efficiently support heavy load for shoring. Fig. 1.1.1 and Fig. 1.1.2 shows scenario of VTT's rescuing work in disaster site. First, VTT reaches rescuing point by locomotion (Fig. 1.1.1). In this procedure, VTT is in cube topology and locomote by rolling. For rolling locomotion, VTT change its geometric configuration by changing its member length using linear actuator on its members. Second, VTT transform to shoring topology by topology reconfiguration (Fig. 1.1.2.(a)). Here, VTT use built-in manipulator to move members and change connectivity between members and nodes. Third, VTT shores the collapsed structure by geometric reconfiguration (Fig. 1.1.2.(b)). Then, rescuer can approach under collapsed building to rescue victims.

Fig. 1.1.1 Rolling locomotion of VTT in disaster site



(a) (b)

Fig. 1.1.2 VTT reconfiguration:
(a) Topology reconfiguration. (b) Geometric reconfiguration for shoring

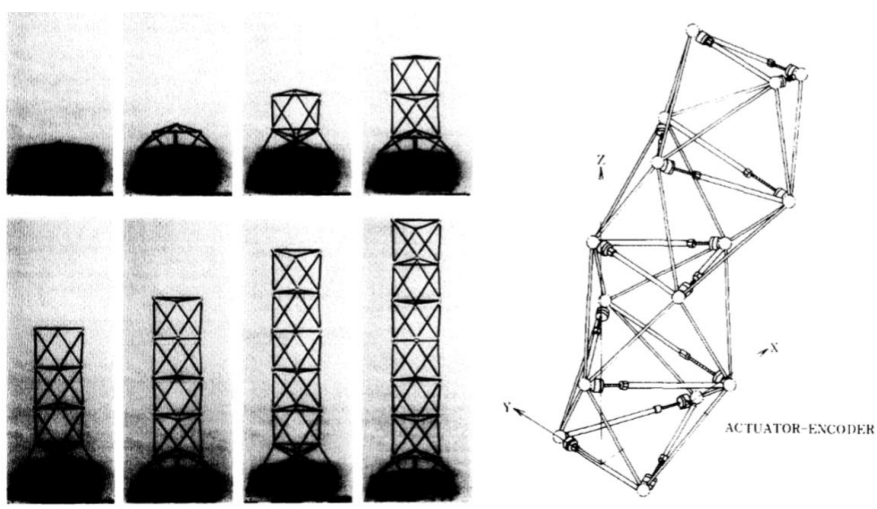
Main topic of this thesis is locomotion of VTT. Locomotion of VTT is complex problem that different from that of conventional robot. First, VTT has high degrees of freedom (DOF). Even the simplest topology of VTT, tetrahedron has 6 DOF. The second simplest topology, octahedron has 12 DOF and the one in the Fig 1.1.1 has 22 DOF. Thus, locomotion is planned in high dimensional space with high level of redundancy which requires sophisticated optimization. Moreover, VTT is basically parallel mechanism with various constraints [8]-[10]. This makes it difficult to predict and avoid constraints for feasible locomotion planning. Although there have been several researches about locomotion of truss type modular robot, they do not fully consider the actual constraints or they were only applicable to simple case: flat ground without obstacle. In this thesis, I will examine actual constraints of VTT and suggest locomotion algorithm that considers the constraints and applicable to various environment.

This thesis is organized as follows. In the remainder of this chapter, early works of truss type modular robot and previous research on the locomotion is introduced. The objectives of locomotion are defined and contribution of this thesis is explained afterwards. In chapter 2, hardware design of VTT and its control system is described. In chapter 3, VTT is mathematically modeled and constraints are explained. In chapter 4, locomotion concept and algorithm are explained. The algorithm is verified with experimental simulation in chapter 5. Finally, conclusions are summarized in chapter 6.

1.2 Previous Truss Type Modular Robot

Earlier concept of active truss structured platform is Variable geometry truss (VGT). VGT is a type of modular robot that has a form of truss which consist of linear actuators at its members and 3 degrees of freedom joint. With its actuators, VGT can flexibly change its geometric shape so that it can be folded into a compact transferrable form and then deployed to achieve its function such as manipulation and shoring [10]-[13]. It can also access and adjust to irregular environments, such as narrow spaces or obstacles [14][15].

VGT concept has been researched for several decades. VGT was first proposed for the design of a space crane arm by Miura et al. [11]. Miura suggested VGT which consist of the repetition of octahedron truss module longitudinally, where the length of lateral member is varied and the length of diagonal member is fixed. He theoretically shows the VGT geometrically reconfigure to be folded and be deployed as manipulator arm. Fig. 1.2.1 shows deployment model simulation and manipulator concept of VGT.



(a) (b)

Fig. 1.2.1 Early concept of variable geometry truss (VGT) [11]:
(a) Deployment model simulation (b) Manipulator concept

Hamlin et al. made early prototype of VGT named Tetrobot [10]. Tetrobot used electric lead screw actuators as linear actuators at its members and used concentric multi-link spherical joint (CMS joint) at its nodes. As in Fig. 1.2.2.(b), CMS joint designed with six bar linkage mechanism so that they function as spherical joint. By controlling linear actuator, shape of Tetrobot can be varied. Furthermore, by rearranging the components manually, it can be changed to different topology to do different function. Fig. 1.2.2. shows two different form of Tetrobot; double octahedron for manipulation and six-legged walker for locomotion. However, this system basically accompanies with many wires for control. Consequently, it has to carry heavy main computer and controller for mobility which makes it lower the efficiency and performance.

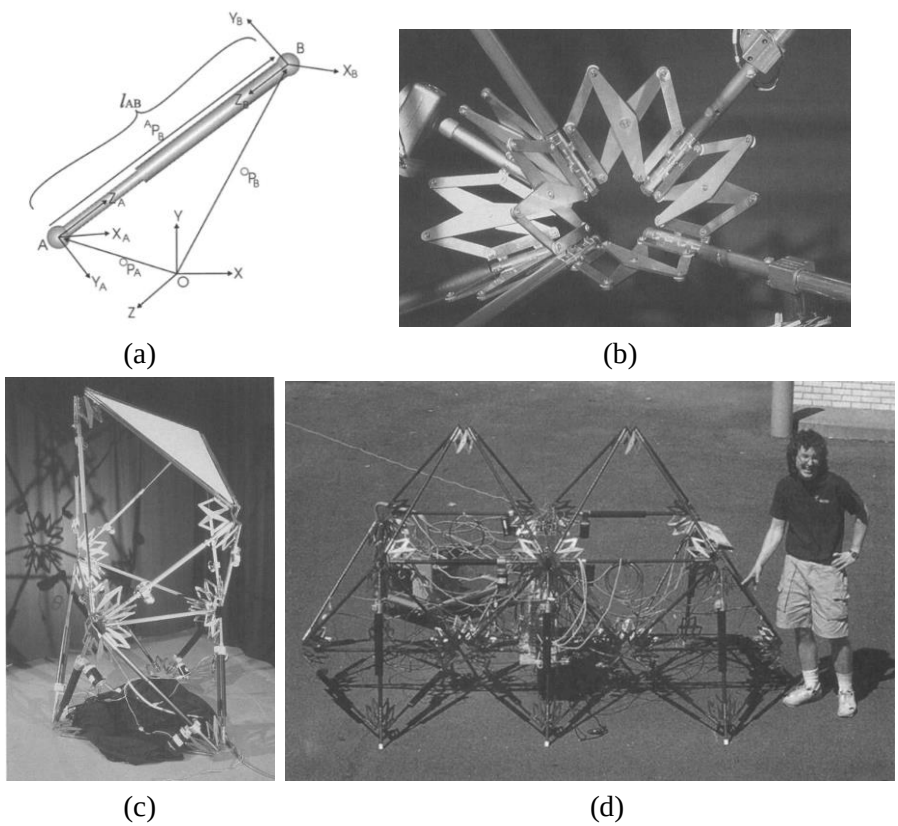
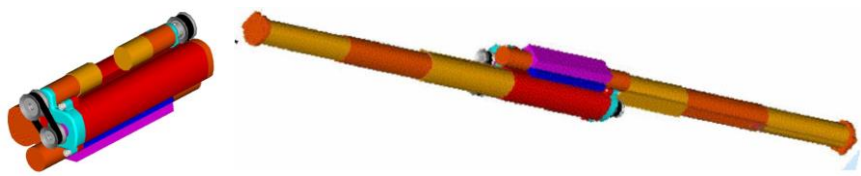
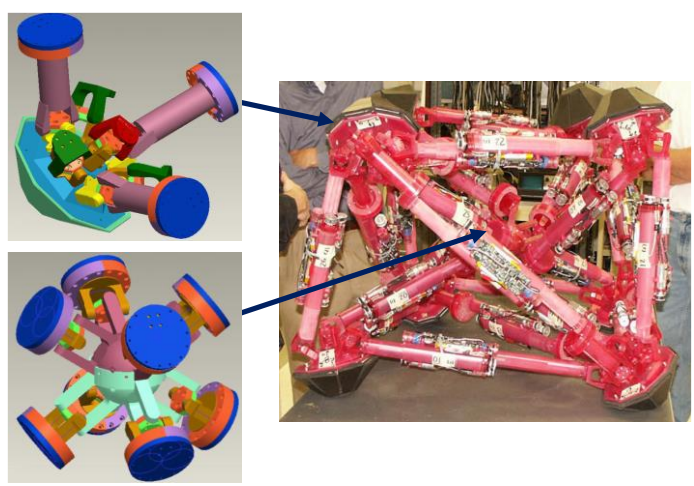


Fig. 1.2.2. Tetrobot prototype [10]: (a) Schematic of electric lead screw actuator (member), (b) double octahedron topology with fixed base. (b) six-legged walker topology

Next generation of VGT improves mobility using wireless control. Curtis et al. in NASA developed VGT called TET robot for space exploration as in Fig. 1.2.3 [14]. TET robot controlled wirelessly using ZigBee protocol. For linear actuation, they developed telescopic linear actuator with a system of nested screws within an exoskeleton. As in Fig. 1.1.3.(a), two actuators are packed so that they can actuated in two-sides. This achieves higher extension ratio of the member. (ratio of maximum length to minimum length). A Node of TET robot is composed of a cluster of universal joints as in Fig. 1.1.3.(b). However, the design of the nodes has to be customized depends on the configuration of the robot. In Fig. 1.1.3.(b), for example, two types of nodes; nodes on the ground and central node was used to form cubic topology with 26 members. This feature makes it difficult to freely change topologies of TET robot.



(a)

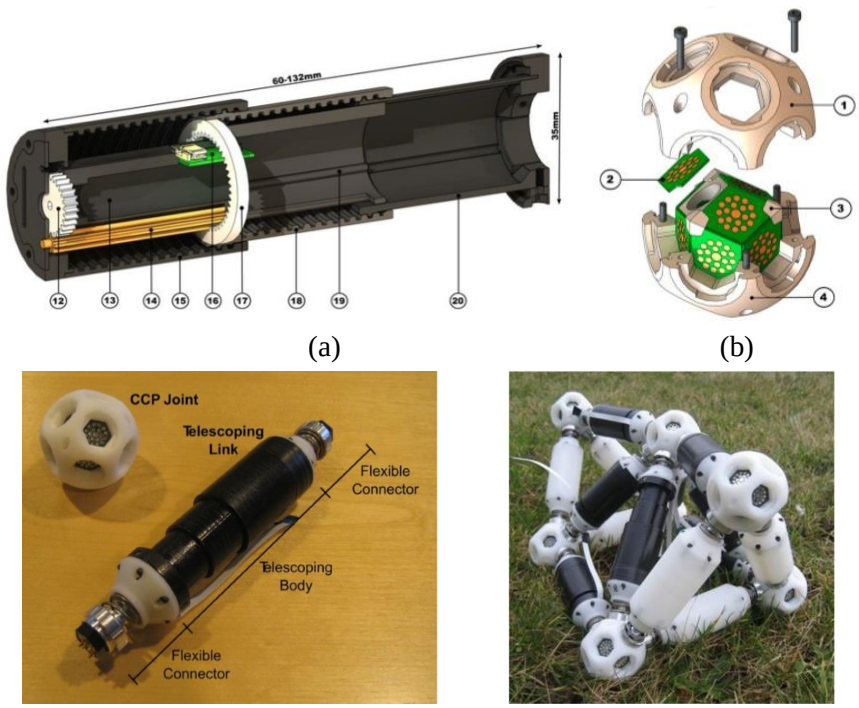


(b)

Fig. 1.2.3. Design of TET robot [14]: (a) Telescopic linear actuator with nested screws within an exoskeleton (member). (b) Cluster of universal joints (node) and assembled TET robot with 26 members and 7 nodes

Lyder et al. developed VGT named Odin that can easily change topologies [15]. Fig. 1.2.4. shows modules and prototype of Odin robot. A member of Odin consists of telescoping body which has 3 stage telescopic linear actuator and flexible connector at its ends. Flexible connector has ball joint for spherical motion when its connected to a node. A node of Odin is cubic closed-packed (CCP) joint. There are hexagon shaped female connectors on surface of CCP joint to connect members. Inside CCP joint are controller and PCB board. Odin can easily connect members and nodes with its specialized connector.

To change topology, previous truss modular type robot must be disassembled and reassembled manually. In this way, they cannot be independent from human assist and only can perform limited function in real application.



(a) (b)

(c) (d)

Fig. 1.2.4. Design of Odin robot [15]: (a) 3 stage telescopic actuator (member).(b) Cubic closed-packed joint (node). (c) Prototype of telescopic actuator (member) and cubic closed-packed joint (node). (d) Assembled robot

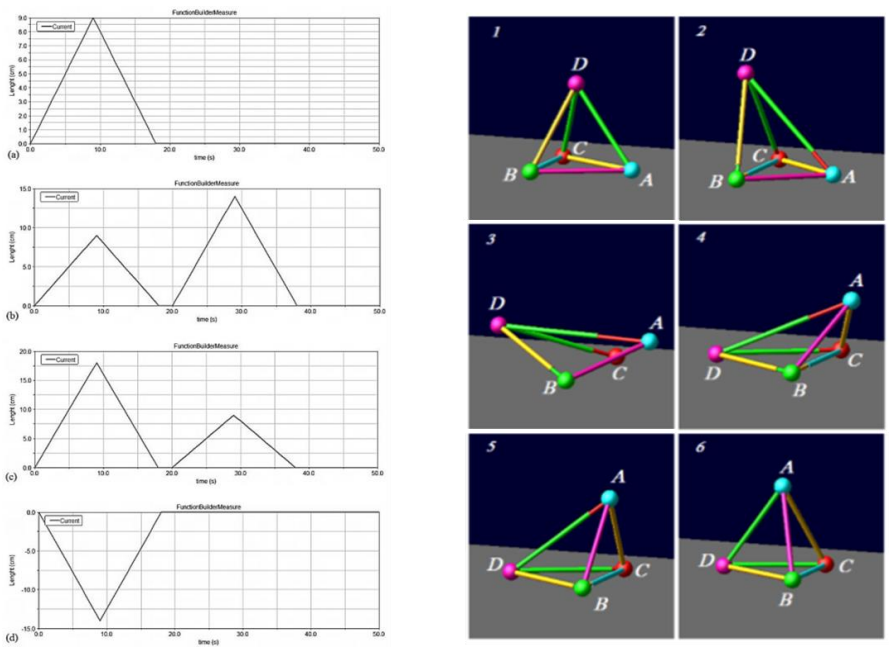
1.3 Previous Research on VTT's Locomotion

There have been several researches about VGT's locomotion and those research provides insight about VTT's locomotion since VTT is kinematically same as VGT during locomotion. VGT's locomotion is achieved by controlling the center of mass by changing its geometric configuration. There is various type of VGT's locomotion such as rolling, crawling and walking [17]-[21]. Among them, rolling locomotion is known to be most efficient and fast [22]. Thus, rolling locomotion will be mainly discussed in this study.

Rolling locomotion is accomplished by tipping the VGT and contacting the ground in front. There have been two kinds of method to achieve rolling locomotion: heuristic based method and optimization based method.

1.3.1 Heuristic Based Methods

Many early works of locomotion were done by planning a gait cycle heuristically. Lee et al. simulated the rolling locomotion of icosahedral VGTs, giving arc trajectories to each control node, so that it could dynamically roll across the ground [17]. Abrahantes et al. simulated tumbling and rolling locomotion of a cube-shaped VGT, giving position profiles of nodes they wanted to move [18][19]. Similarly, Motahari-Bidgoli et al. planned length profiles of members for locomotion of tetrahedron-shaped VGT as shown in Fig. 1.3.1 [20]. These methods executed pre-planned motions of nodes so that the VGT could move one gait rotation cycle. Although their planning method is intuitive, it can only be applied to pre-planned configurations and certain moving patterns. It cannot move in irregular environments and cannot freely follow arbitrary trajectories.



(a) (b)

Fig. 1.3.1. Heuristic method of locomotion by Motahari-Bidgoli et al.[20]:
(a) Velocity profile of members. (b) Rolling locomotion

1.3.2 Optimization Based Method

Another method of locomotion planning is based on optimization algorithms. Usevitch et al. proposed a locomotion algorithm that optimizing the velocity of nodes at every time step when the desired VGT's trajectory was given [23]. The algorithm was verified by simulating the locomotion of a tetrahedral VGT. Once the optimization algorithm was set up, it could be applied to any VGT configuration. It could also be applied to various trajectories and terrains, because it did not depend on a specific gait cycle. Fig. 1.3.2 shows its locomotion simulation shows and Fig. 1.3.3 shows the algorithm.

Although this optimization-based algorithm suggested useful methods for generalizing locomotion, it has several limitations. First, shape of VGT tends to be distorted too much as it moves forward. This shape distortion makes VGT easily violate its constraints which induce planning failure. The other problem is that VGT damaged from the ground during rolling. As VGT rolls, it tumbles to the ground and is subjected to impact. Since VGT is composed of many linear actuators and spherical joints, this impact causes significant damage to those components.

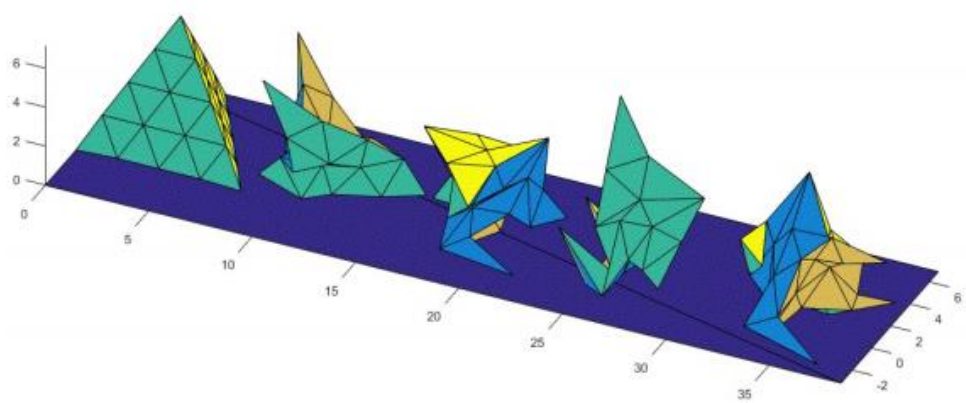


Fig. 1.3.2. Locomotion simulation with optimization method by Usevitch [23]

Algorithm 1 Constraint-Violation Free Motion

function NEXTCONFIG($G, x, \dot{x}_{cm}$)
  D=[]
  Compute $R(x)$
  $Feasible = False$
  while $Feasible == False$ **do**
    $\dot{x}_{temp} = DesiredMotion(D, \dot{x}_{cm})$
    $x_{new} = x + \dot{x}_{temp} dt$
    Active=0
    for $i = 1$ to $N_{constraints}$ **do**
      if $Constraints_i(x_{new}) < 0$ **then**
        D=[D; $\frac{\partial f_i(x)}{\partial x}$]
        $Active = Active + 1$
      end if
    end for
    if $Active == 0$ **then**
      $Feasible = True$
    end if
  end while
  $x(t + dt) = x_{new}$
  $\dot{L} = R(x)\dot{x}_{temp}$
end function

Fig. 1.3.3. Locomotion algorithm by Usevitch [23]

1.4 Objectives of Locomotion Algorithm

Although previous researches provided useful insight about locomotion planning, they only dealt with obstacle-free environment. However, obstacle must be considered in actual case, since various obstacles such as rocks, pits, debris, walls and furniture commonly exist in outdoor and indoor disaster site. In addition, it is necessary to prevent VTT from damaged from the ground during rolling. Considering these issues, I defined the objectives of locomotion algorithm as follows.

Given start position and goal position of VTT,
plan the trajectory of VTT's nodes that reach the goal position by rolling,
while avoiding obstacles, and prevent the damage from the ground

Fig. 1.4.1 shows schematic of locomotion environment. Here, it was assumed that the position and shape of obstacle is known by vision sensor. Note that, position of the nodes defines geometric configuration of VTT, and planned trajectory of VTT's nodes defines motion of VTT over time.

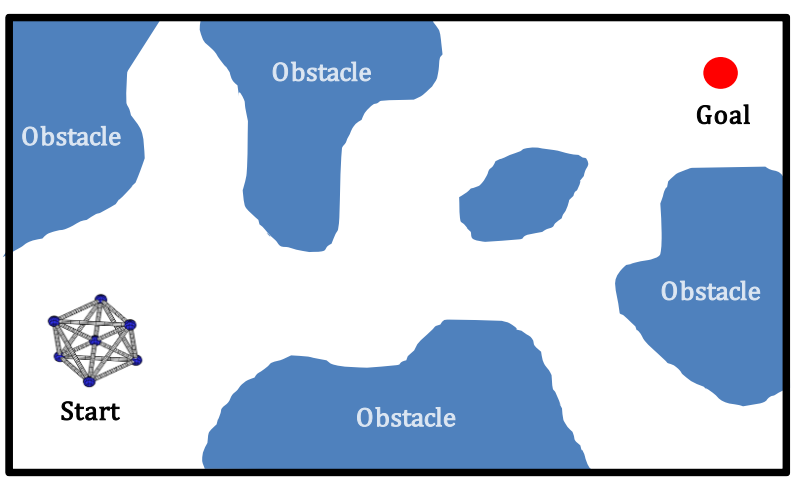


Fig. 1.4.1. Schematic of locomotion environment

1.5 Contribution of Thesis

In this thesis, I will introduce hardware design of VTT for self-reconfiguring its topology. Then, I will propose stable rolling locomotion algorithm that applicable to actual VTT and environment.

1.5.1 Introduction to Hardware Design of VTT

VTT is first truss type modular robot that can self-reconfigure its topology. By topology reconfiguration, it can perform more versatile and practical function in real application as in Fig. 1.1.1. To realize self-reconfiguration, VTT was designed as Fig. 1.5.1. Members of VTT consist of a Spiral Zipper. Spiral Zipper is novel linear actuator that change its length by winding and unwinding its zipper shaped band. Spiral Zipper has very high extension ratio (ratio of minimum length to maximum length) and has much higher strength to weight ratio than conventional linear actuator [16]. Nodes of VTT consist of Sphere, Passive Member-End and Master Member-End. Passive Member-End is a linkage type spherical joint that connect members at a node. Sphere is connected to the other end of Passive Member-End and contact to ground to protect VTT. Master Member-End is a spherical manipulator that is built in Sphere and moves a member for topology reconfiguration. Bottom of Fig. 1.5.1 shows concept of VTT's topology reconfiguration and the function of Master Member-End. With Master Member-End and high extension ratio Spiral Zipper, VTT can flexibly reconfigure its topology as well as its geometric shape.

Fig. 1.5.1. Design of VTT

1.5.2 Stable Rolling Locomotion of VTT

The proposed rolling locomotion algorithm is stable in three aspect. First, proposed algorithm is based on optimization with high planning success rate. Previous optimization method was easily failed due to shape distortion which induce constraints violation. However, this algorithm properly defines the support polygon and the center of mass trajectory so that the robot can always remain in good condition. This makes the configuration is far from the constraint boundary and leads to successful planning even with tight constraints. Schematic of support polygon and center of mass trajectory is shown in Fig. 1.5.2.

Second, proposed algorithm can avoid obstacles. Unlike the locomotion of typical wheeled platform which is controlled by simple steering movement, rolling locomotion is less intuitive to be controlled manually, especially when there are many obstacles to be avoided. Thus, it is necessary to plan the trajectory between start position and goal position considering obstacle. In my algorithm, obstacle avoiding path planning is achieved by support polygon planning using new random search algorithm, Polygon-Based Random Tree (PRT).

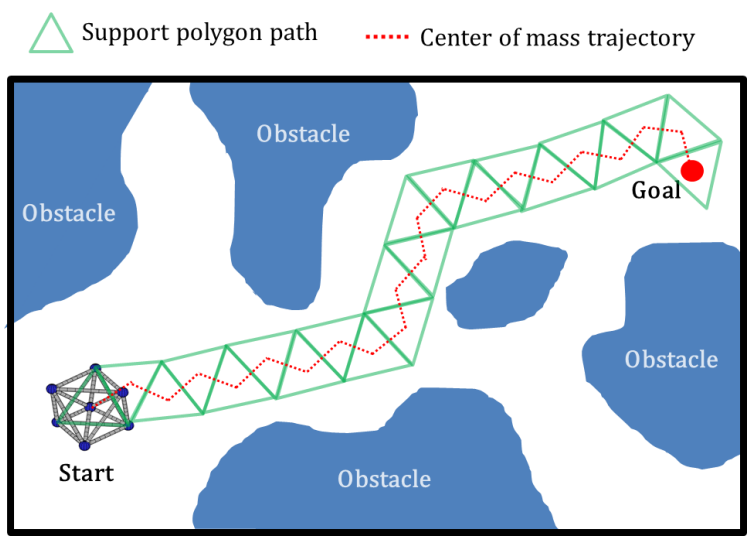


Fig. 1.5.2. Schematic of support polygon path and center of mass trajectory

Third, proposed algorithm can prevent VTT from being damaged from ground. The damage prevention is achieved by maintaining static stability using Non-Impact Rolling locomotion. As illustrated in Fig. 1.5.3, algorithm for Non-Impact Rolling locomotion makes VTT maintain stability and smoothly land on the ground without tumbling and impact from the ground.

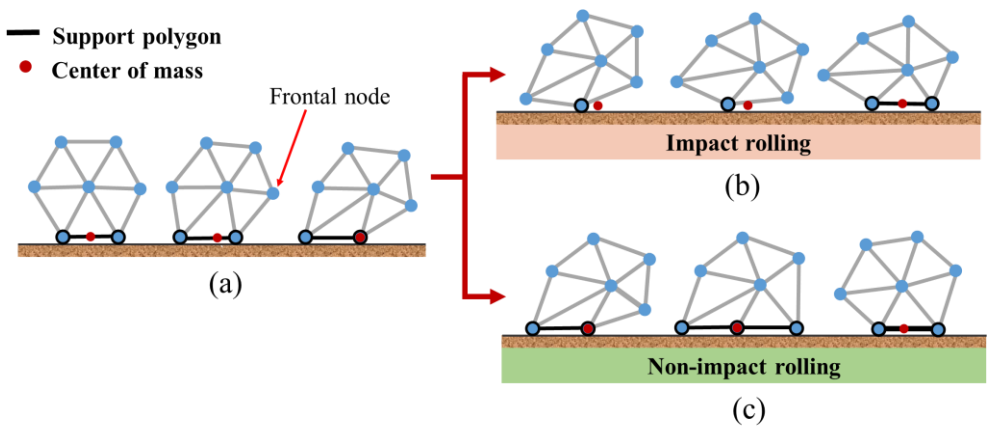


Fig. 1.5.3. Schematic of Impact Rolling and Non-Impact Rolling:
(a) Configuration before frontal node landing,
(b) Impact Rolling motion, (c) Non-Impact Rolling motion

Chapter 2

Design of Variable Topology Truss

The main feature of VTT is topology self-reconfiguration. In this work, self-reconfiguration was realized by high extension ratio members and built-in manipulator that moves the member. With high extension ratio members, VTT can flexibly change its geometric configuration to facilitate topology reconfiguration. With built-in manipulator, VTT can achieves topology reconfiguration by moving members and change connectivity between members and nodes.

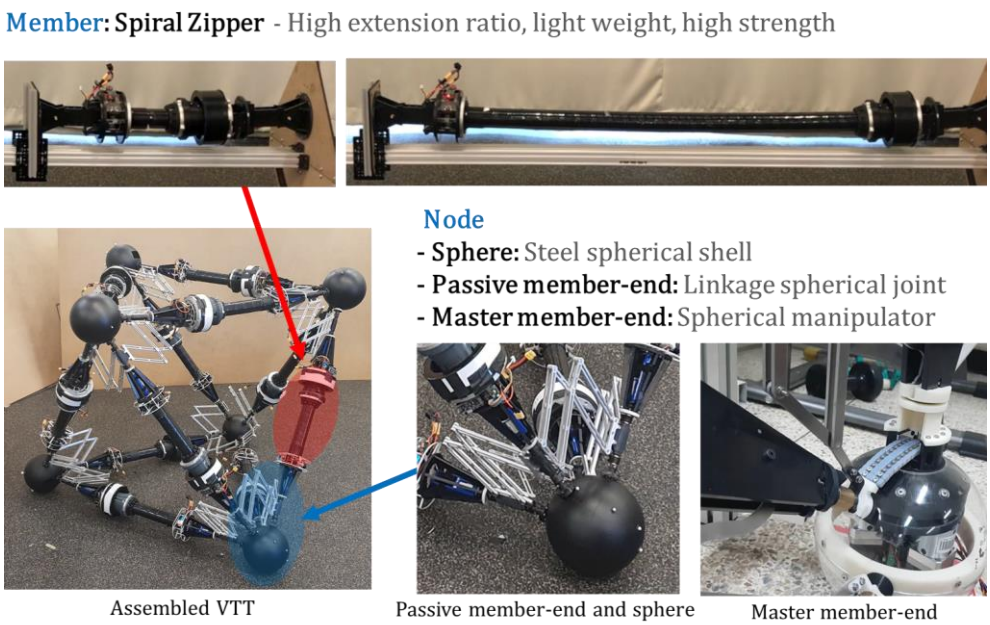


Fig. 2.1. Overall design of VTT

Overall design of VTT is shown in Fig. 2.1. VTT is basically truss structure composed of members and nodes. A member of VTT consists of novel linear actuator, Spiral Zipper. Spiral Zipper has high extension ratio and high strength to weight ratio compare to conventional linear actuator [16].

A node of VTT consists of Sphere, Passive Member-End and Master Member-End. Passive Member-End is a linkage type spherical joint that connect members at a node. Sphere is connected to the other end of Passive Member-End and contact to ground to protect VTT. Master Member-End is a spherical manipulator that is built in Sphere and move a member for topology reconfiguration. Detail design and mechanism of each components is explained in next sections.

2.1 Member Design

Member of VTT requires two functions. The first is to support the axial force, and the second is to control its length to change geometric shape of VTT. To perform the two functions, the member consists of Spiral Zipper system and tensioner. Spiral Zipper is newly developed linear actuator and tensioner is the part that reinforce the tensile strength of band. Prototype of VTT member is shown in Fig. 2.1.1 and its detail design is presented in Fig. 2.1.2

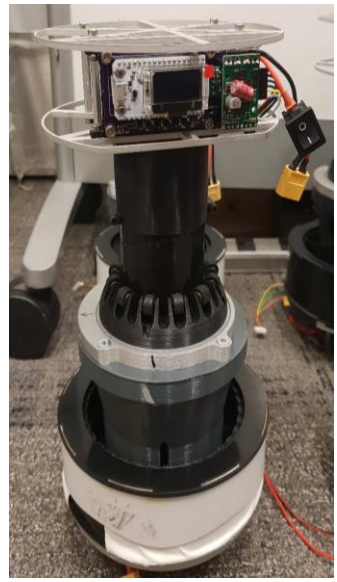


Fig. 2.1.1. Prototype of VTT member

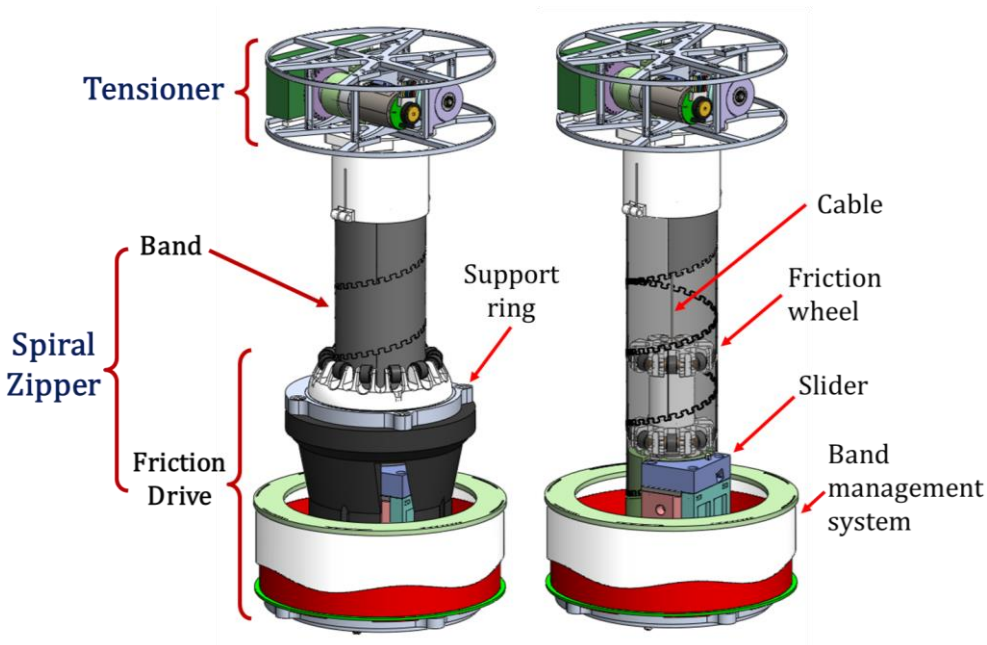


Fig. 2.1.2. Member design of VTT

2.1.1 Spiral Zipper

Spiral Zipper changes its length by winding and unwinding its zipper shaped band using Friction Drive. The band of Spiral Zipper is shown in Fig. 2.1.3. The band has 2D teeth shape on upper and lower edge and is manufactured by laser cutting 1/32-inch-thick ABS. As the band winds in a spiral, its teeth are meshed each other so that the band can maintain column shape. However, because of its 2D shape, the teeth are easily de-meshed by small external shear force. To reinforce the shear

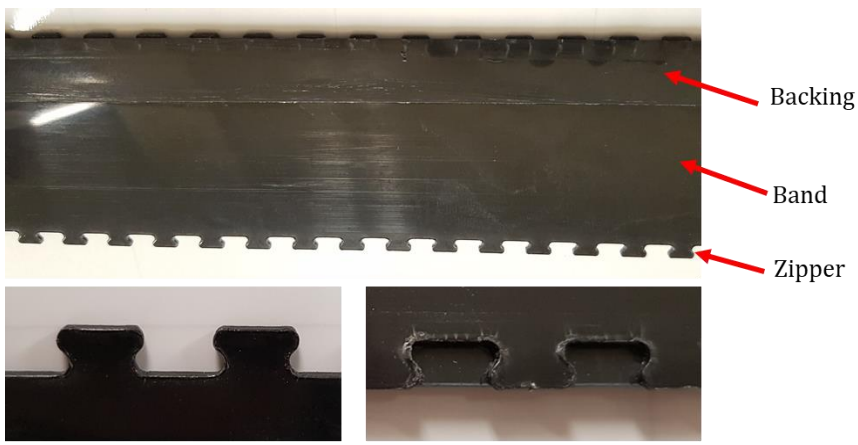
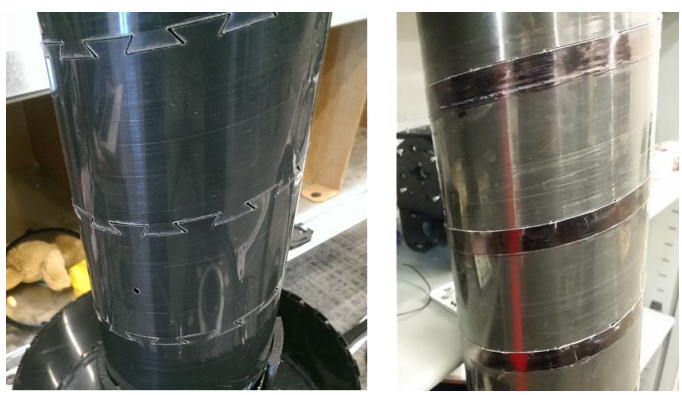


Fig. 2.1.3. Band of Spiral Zipper

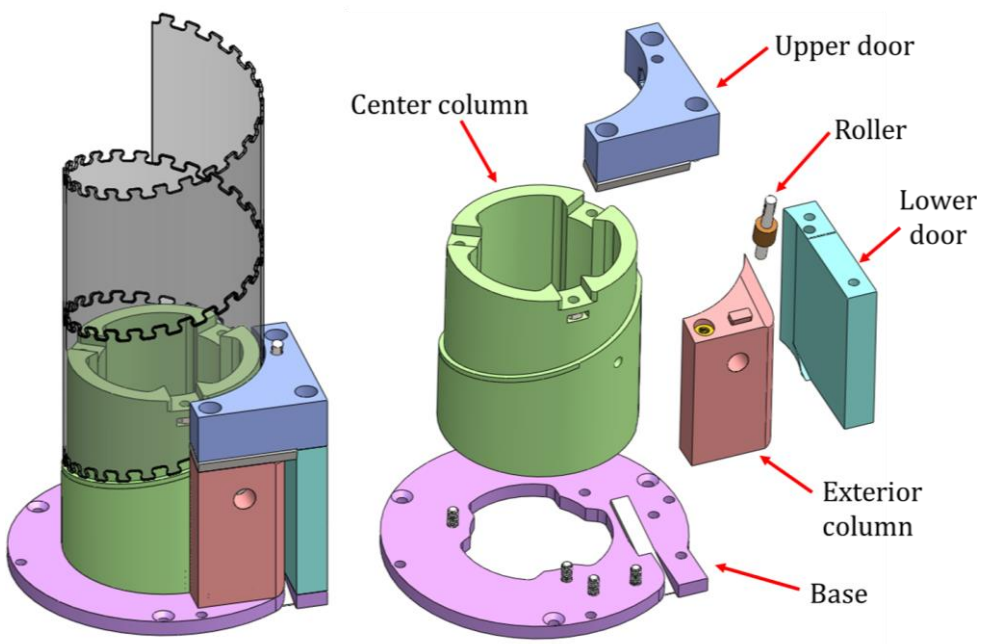


(a) (b)

Fig. 2.1.4. Spiral Zipper column: (a) without backing, (b) with backing

strength of zipper column, backing, a strip of ABS (0.01-inch-thick) is glued on upper teeth of the band as in Fig. 2.1.3. Fig. 2.1.4 shows the zipper column with and without backing. As in the figures, backing prevent the teeth from de-mesh radially and makes the column withstand large shear force.

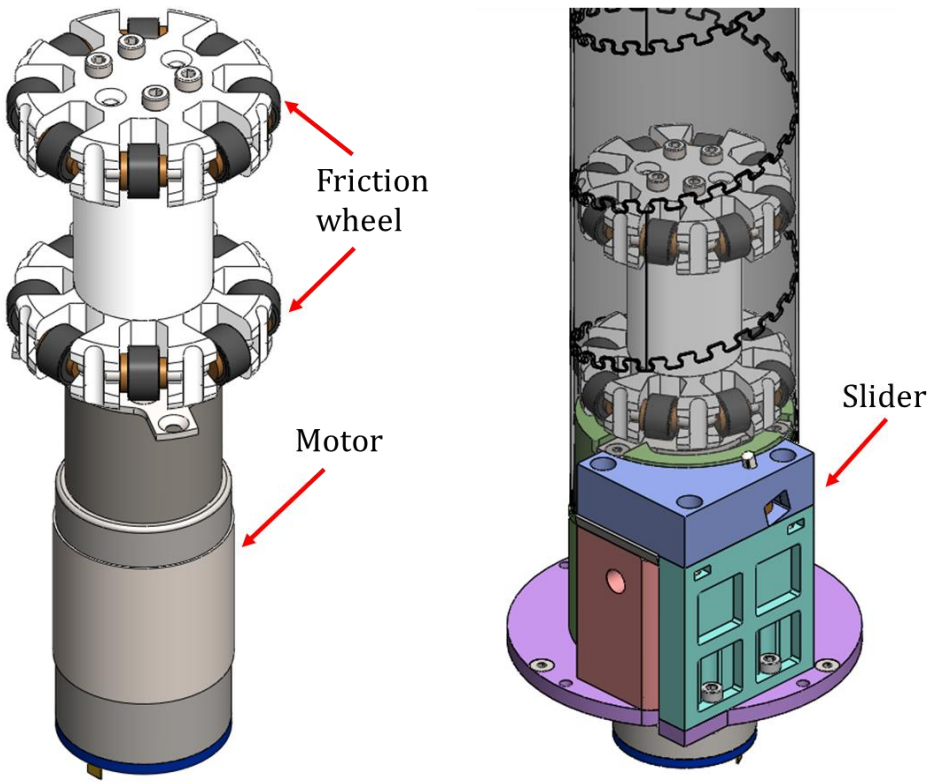
Friction Drive is actuation system that winds and unwinds the band to adjust length of the member. As in Fig. 2.1.2, Friction Drive is composed of Slider, Friction Wheel, Support Ring and Band Management System. Slider is the most essential part of Friction Drive that guide the band to be wind and its teeth to be meshed well. Structure of Slider is shown in Fig. 2.1.5. Center column has slope on the surface and exterior column surround the center column. By moves along the slope, band is winded in desired angle and its teeth meshed exactly. A roller on the upper and lower door is located on meshing point of teeth (the point that the teeth started to mesh) to press and guide it smoothly.



(a) (b)

Fig. 2.1.5. Design of Slider: (a) Assembled model with band, (b) Exploded view

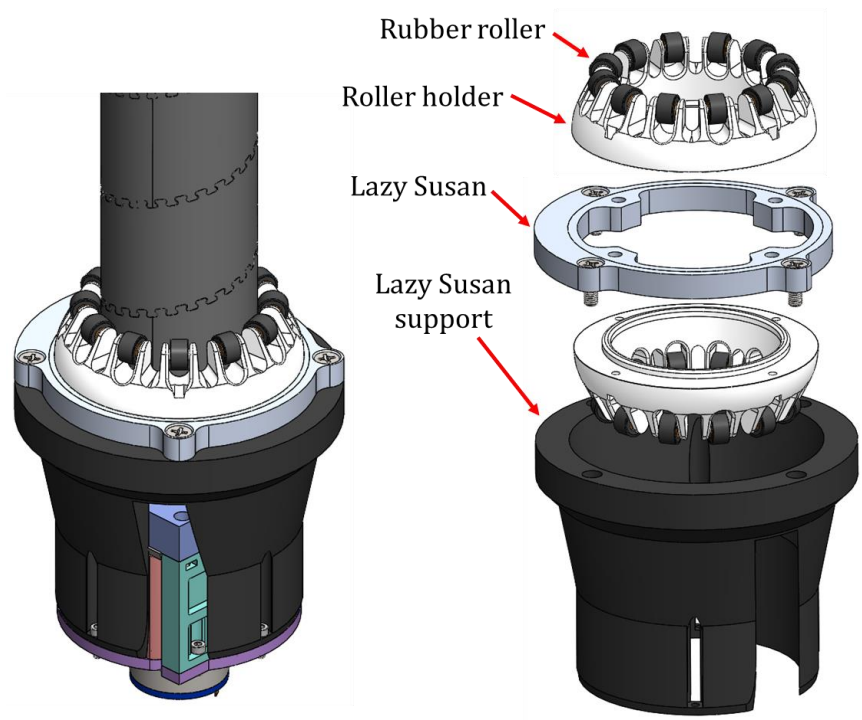
Friction Wheel is the part that rotate the band by friction. Structure of Friction Wheel is shown in Fig. 2.1.6. Friction Wheel is rotated by motor (Pololu 638324). As the wheel rotate, rubber roller on Friction Wheel makes the band rotate together.



(a) (b)

Fig. 2.1.6. Design of Friction Wheel
(a) Friction Wheel and motor, (b) Friction Wheel system inside Slider and band

Support Ring is the part that support the band from shear force and moment. Structure of Support Ring is shown in Fig. 2.1.7. Meshing point of the band is very weak to shear force and moment. To protect the weak part, rubber rollers on the Support Ring surround and contact the meshed part of the band, so the shear force and moment is transfer to Support Ring rather than weak meshing point. In the Support Ring, roller holders contain the rubber rollers and the roller holder is assembled on the inner wheel of the Lazy Susan bearing. When the band is would up by Friction Wheel, rubber roller allows the band to move longitudinally and the Lazy Susan allows the roller holder to rotate with band. Thus, Support Ring does not interfere the band actuation while reinforce its strength.

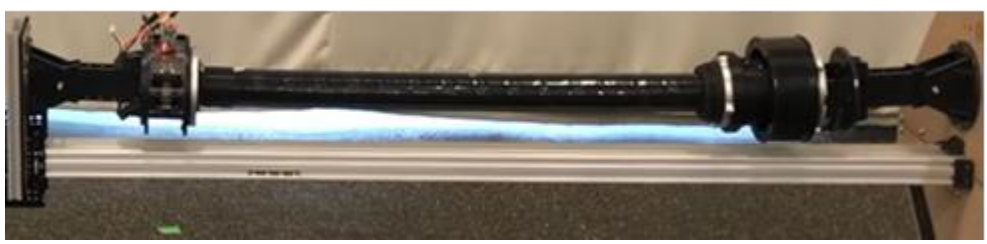


(a) (b)

Fig. 2.1.7. Design of Support Ring:
(a) Support Ring with the band, (b) Exploded view of Support Ring

Fig. 2.1.8 illustrates the necessity of Support Ring. When Spiral Zipper is in horizontal configuration without Support Ring, meshing point is subjected to moment exerted by weight of Friction Drive and the band column sags as in Fig. 2.1.8.(a). In contrast, if Spiral Zipper has Support Ring, it can withstand and maintain straight form as in Fig. 2.1.8.(b).

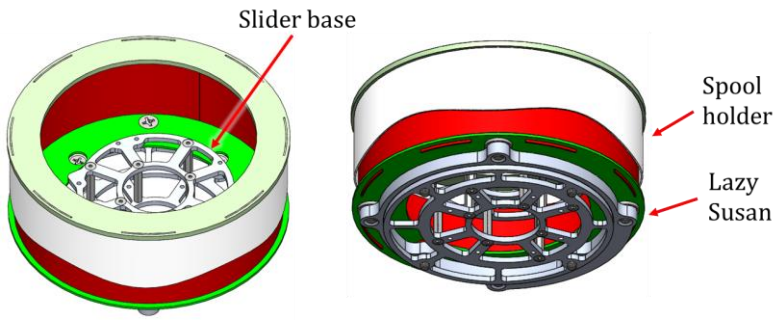
(a)



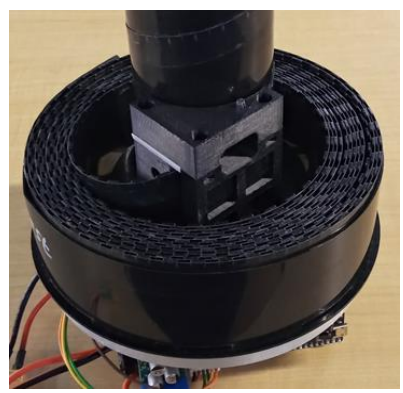
(b)

Fig. 2.1.8.Spiral Zipper in horizontal configuration
(a) without Support Ring, (b) with Support Ring

Band Management System is the part that stores un-winded band. Structure of Band Management System is shown in Fig. 2.1.9. Slider base is the connecting part of Slider and spool holder contains un-winded band. The Slider base and the spool holder are assembled to inner wheel and outer wheel of the Lazy Susan bearing respectively. Therefore, spool holder can rotate independently from Slider part. This helps the band to lie and roll stably in the spool as the band winded and un-winded by Slider.



(a)



(b)

Fig. 2.1.9. Design of Band Management System.
(a) Band Management System model.
(b) Band Management System prototype that contains un-wind band.

2.1.2 Tensioner

Due to its zipper and backing shape, Spiral Zipper has high tensile strength of over 1000 N [16]. Whereas, its tensile strength is so low that it is only about 20 N. With this low strength, VTT can be fall apart even with its own weight. To compensate this problem, Tensioner is used to reinforce the increase strength of the Spiral Zipper. Tensioner utilize cable, which has opposite property of Spiral Zipper: strong in compression but weak in tension.

Tensioner is Series Elastic Actuator (SEA) system that apply constant tensile force to the Friction Wheel by cable. Design of Tensioner is shown in Fig. 2.1.10. SEA is basically torque control system and use torsion spring as a torque sensor

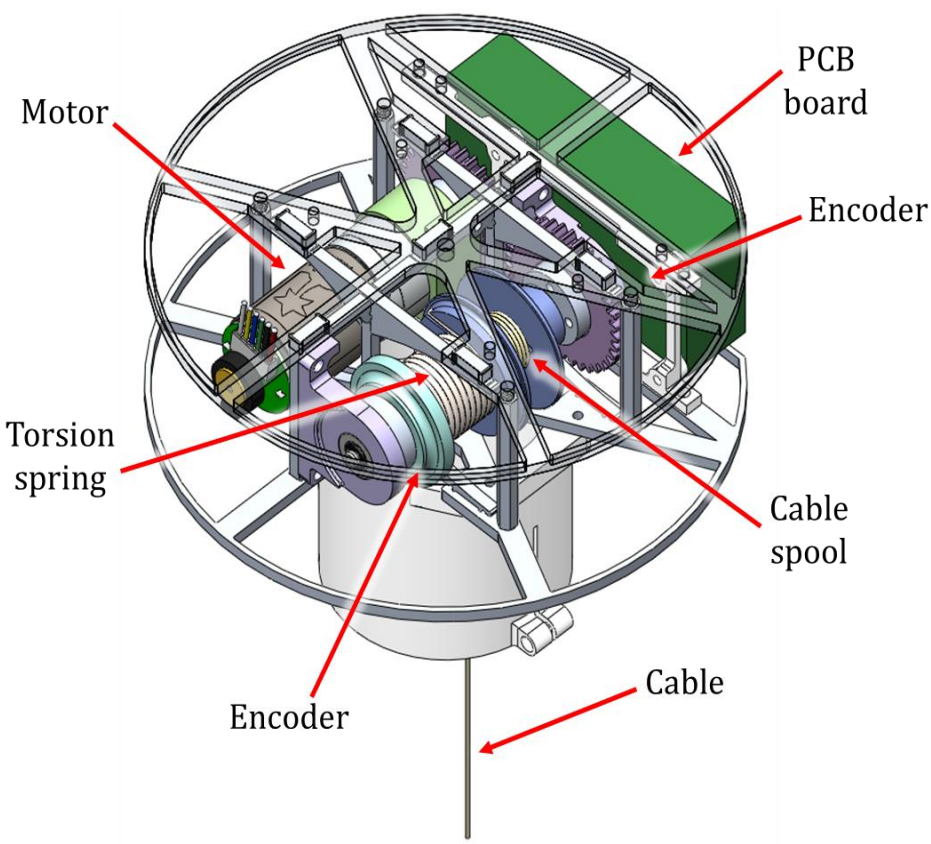


Fig. 2.1.10. Design of tensioner

[24]. The cable of Tensioner is connected to top side of the Friction Wheel. Thus, the Tensioner motor can apply a pulling force on Spiral Zipper to generate a compressive preload to the Spiral Zipper band. When the motor torque is applied to cable spool, torsion spring is deformed. Then, the angle deflection is measured by encoders at the two ends of torsion spring. The measured deflection is used as torque value for torque control. Consequently, the Tensioner applies constant compressive preload to the Spiral Zipper band regardless of the length of band.

Fig. 2.1.11 shows protype of Tensioner. Frame of the Tensioner was manufacture by waterjet cutting the aluminum panel. The cable was made of Kevlar and has a tensile strength of 4440 N which is much higher than the compressive strength of the band.

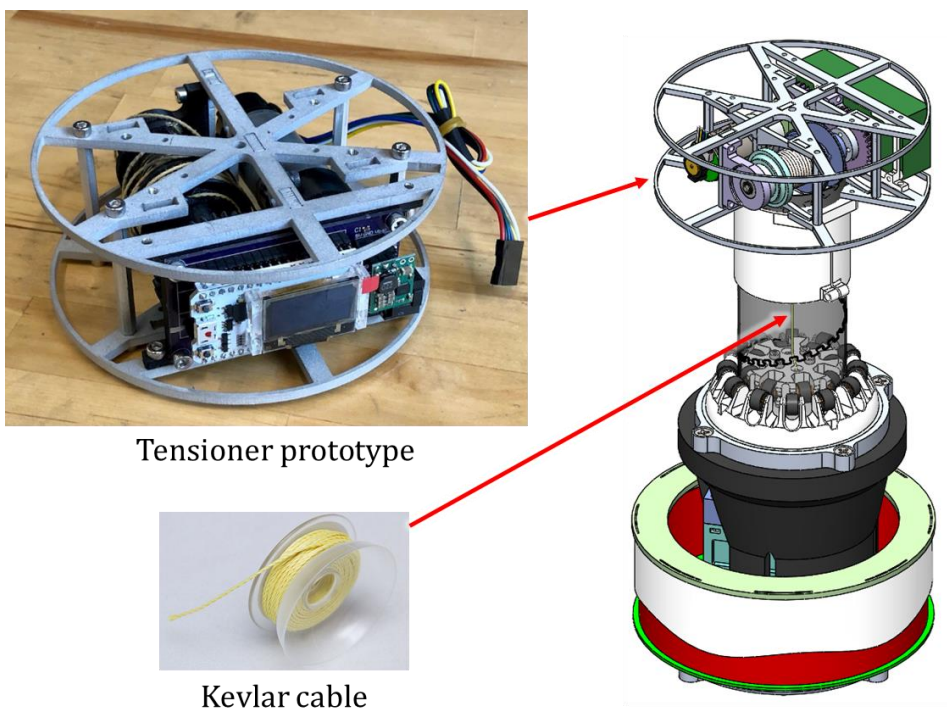


Fig. 2.1.11. Tensioner prototype

2.2 Node Design

Node of VTT has two functions. The first is to connect and give relative DOF to members and it is achieved by Passive Member-Ends. Here, Sphere is attached to the Passive Member-Ends to protect it from ground. The second function is to change the connection between members and nodes for topology reconfiguration, and it is implemented using Master Member-Ends. The design and prototype of Passive Member-Ends, Sphere and Master Member-End is presented in Fig. 2.2.1 and Fig. 2.2.2.

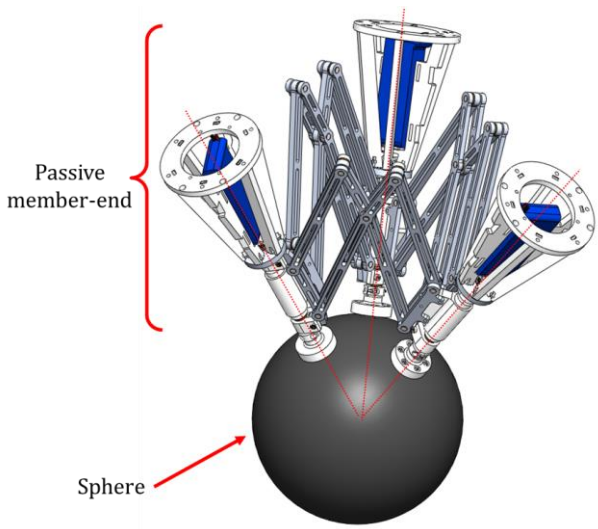


Fig. 2.2.1. Passive Member-End and Sphere

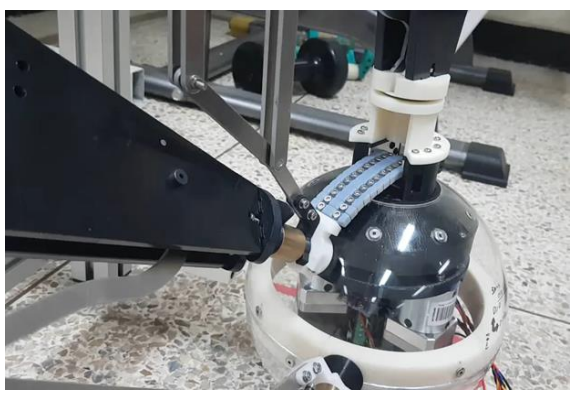


Fig. 2.2.2. Master Member-End prototype

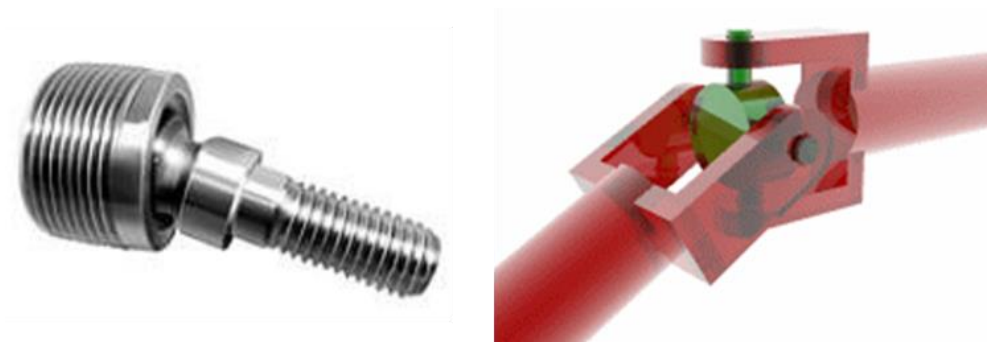
2.2.1 Passive Member-End and Sphere

The Passive Member-End function as a joint to connect members of VTT. To implement the VTT, Passive Member-End requires the functions listed as follows.

Functional requirements (FRs) of Passive Member-End

1. Have large angle range in 3 rotational DOF
2. Accommodate many members (at least 4)
3. Member axes meet at one point

Functional requirements (FRs) 1 and 2 Is to implement various topologies and configuration of VTT, and 3 is to ensure the member of VTT is subjected to pure axial force. Conventional 3 DOF joints are ball joint and universal joint as in Fig. 2.2.3. For reference, universal joint is originally 2 DOF but can be increased to 3 DOF by adding revolute joint which axis is parallel to the arm. However, ball joints and universal joints alone can accommodate only 2 members.



(a) (b)

Fig. 2.2.3. Conventional 3 DOF joint. (a) Ball joint, (b) Universal joint

Some works attempt to accommodate many members by using ball joints and universal joints. Andreas et al developed cluster of ball joints called CCP joint for the node of Odin robot [15] as in Fig. 2.2.4. However, angle range of the joints was small (-24 to 24 degree). Additionally, the member axes do not meet at one point when the joint angle deviate from 0 degree, which generate shear force and moment to the truss. Curtis et al developed cluster of universal joints for the node of TET robot [14] as in Fig. 2.2.5. Although the node of TET robot has large angle range than that of Odin robot, the member axes still do not meet at one point.

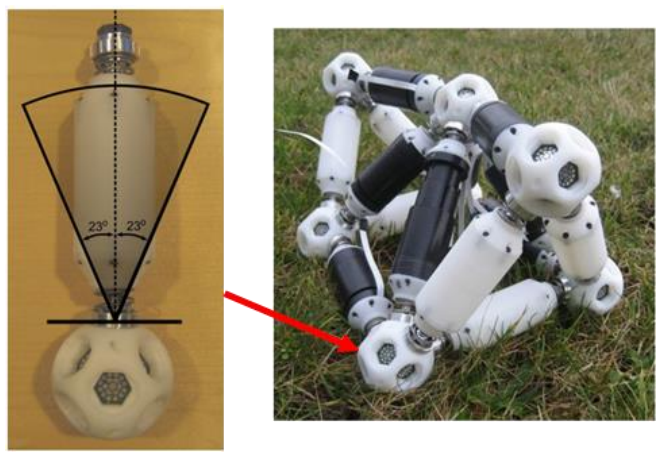


Fig. 2.2.4. Node of Odin robot [15]

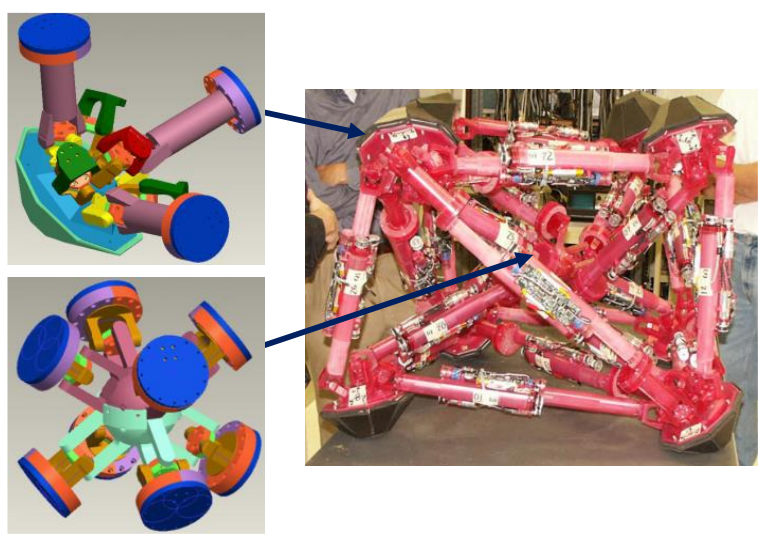


Fig. 2.2.5. Node of TET robot [14]

To solve the problems of conventional joints, Hamlin et al developed novel Concentric Multilink Spherical (CMS) joint [10][25] as in Fig. 2.2.6. CMS joint is basically six bar linkage and has link collar on the leaf to connect truss member. CMS joint satisfies all three FRs. First, it has large angle range in 3 rotational DOF by linkage structure. Second, it can accommodate any number of members by connecting the joint in a row as in Fig. 2.2.7. Third, by specially designed bent linkage, member axes meet at a point.

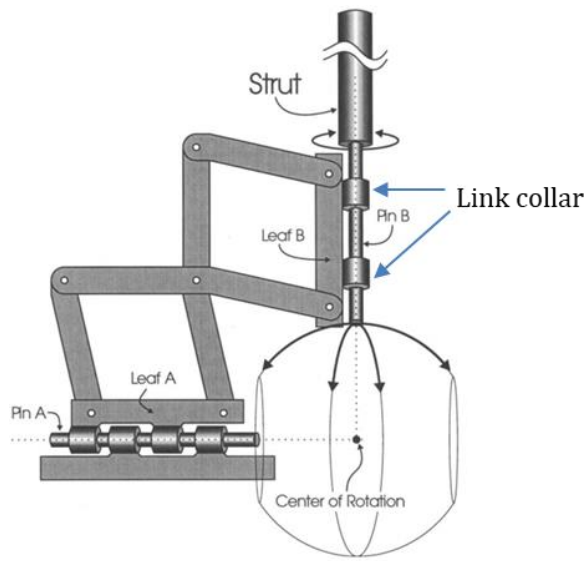


Fig. 2.2.6. Concentric multilink spherical (CMS) joint [10]

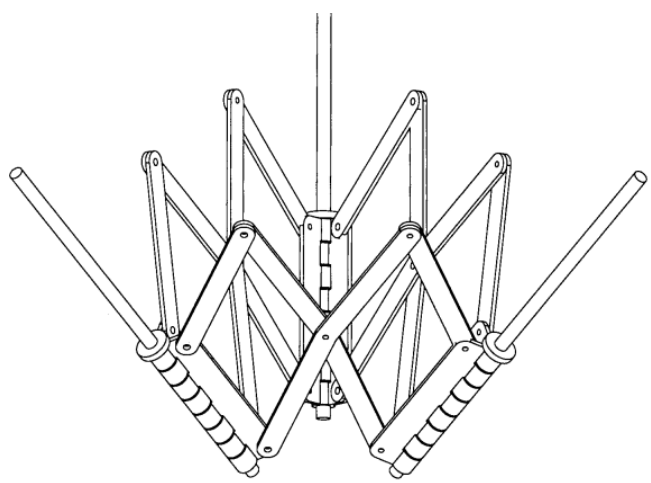


Fig. 2.2.7. Three CMS joint connected in a row [10]

Satisfying all FRs, CMS joint mechanism was used for Passive Member-End. Design of CMS joint is determined by 3 parameters: L_1, L_2, and L_3 as in Fig. 2.2.8. The selected parameters and design of Passive Member-End linkage is presented in Fig. 2.2.9. In Passive Member-End design, new design parameter, end angle offset ϕ_0 was added to give enough space for end hinge.

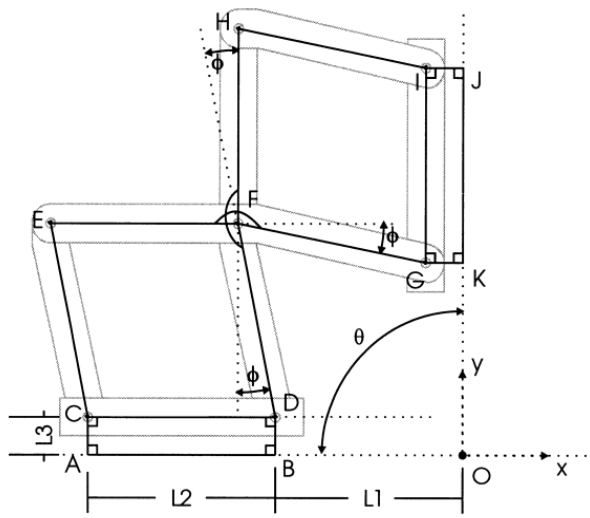


Fig. 2.2.8. Design parameters of a CMS joint [10]

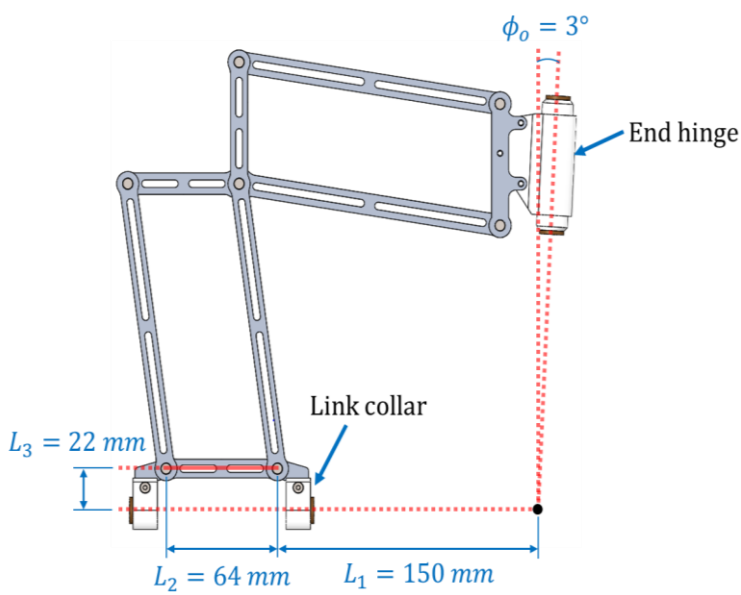


Fig. 2.2.9. Design parameter of a Passive Member-End linkage

Design of a Passive Member-End is shown in Fig. 2.2.10. Passive Member-End is composed of six bar linkages and Passive Member-End body. In the body, Member-Interface is connected to the member, Sphere Interface interact with the Sphere and shaft is connected to the end hinge of another Passive Member-End. A battery is placed inside body support to supply power to Spiral Zipper or Tensioner. Fig. 2.2.11 shows Passive Member-Ends connected to a Sphere. As in the figure, Sphere Interface moves on the surface of Sphere and angle range of a Passive Member-End is 25 degree to 155 degree as represented.

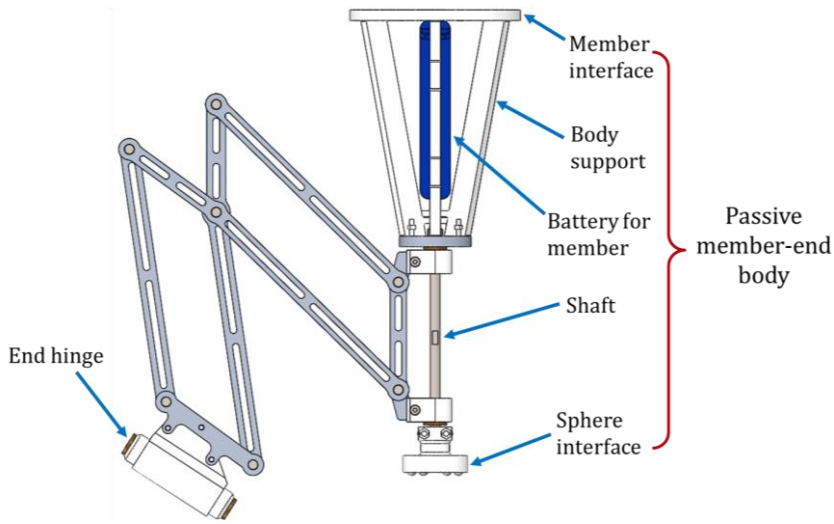


Fig. 2.2.10. Design of a Passive Member-End

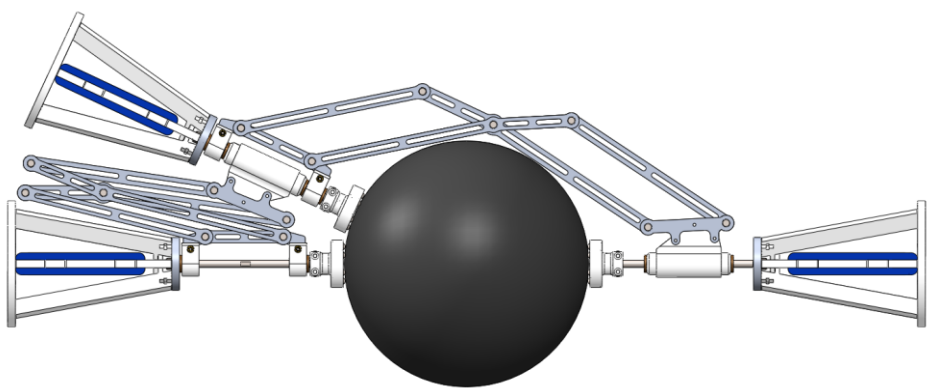


Fig. 2.2.11 Passive Member-Ends with a Sphere:
one at minimum angle and the other at maximum angle

Fig. 2.2.12 shows three Passive Member-Ends and a Sphere connected each other. Among Passive Member-Ends on a Sphere, only one Passive Member-End is fixed to the Sphere, and the others freely moves on the surface of Sphere. The bottom of Fig. 2.2.12 shows two kinds of Sphere Interface, fixed Sphere Interface for fixing to the Sphere, and free Sphere Interface for moving on the Sphere. Fixed Sphere Interface is fixed to the Sphere by screw, and needle bearing is placed in between cup and Sphere to give rotational freedom of cup. Free Sphere Interface has ball roller so it can smoothly move on the surface of the Sphere.

Sphere support the force transferred from Passive Member-End and also protect member from ground. Sphere is made of 0.7 mm thickness steal and its diameter is 200 mm. The diameter of Sphere is larger than that of member structure so the members does not contact the ground.

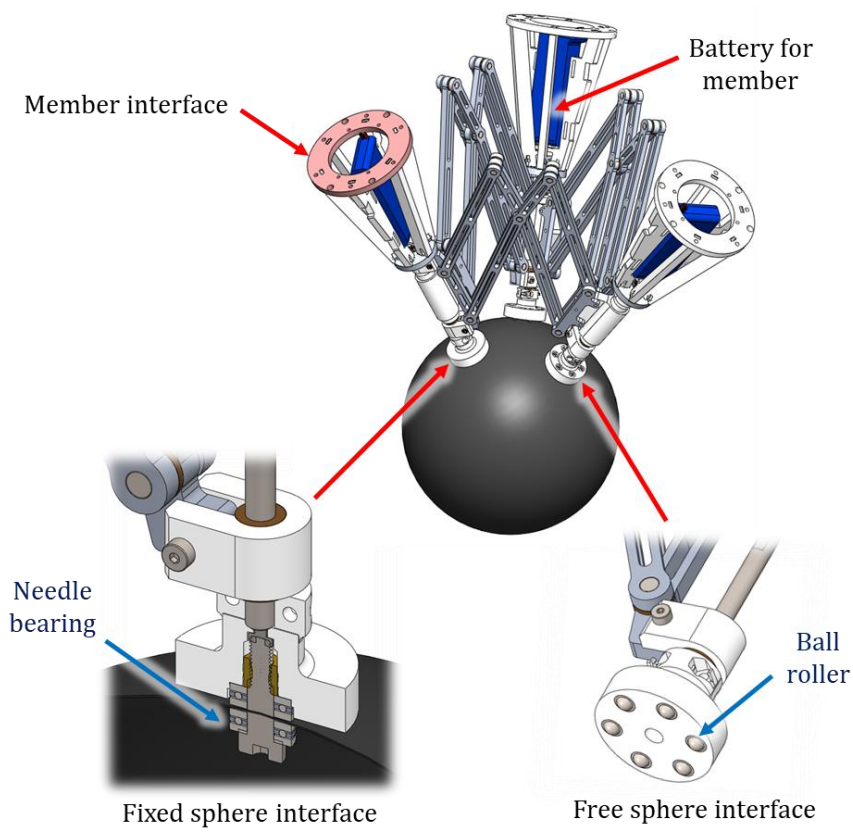


Fig. 2.2.12. Three Passive Member-Ends on a Sphere

Assembly of a member and two Passive Member-Ends forms a edge module which is which is a unit module of VTT as shown in Fig. 2.2.13. VTT can transforms to various geometric shape and topology by controlling the length of edge modules and reconfiguring the connection between them.

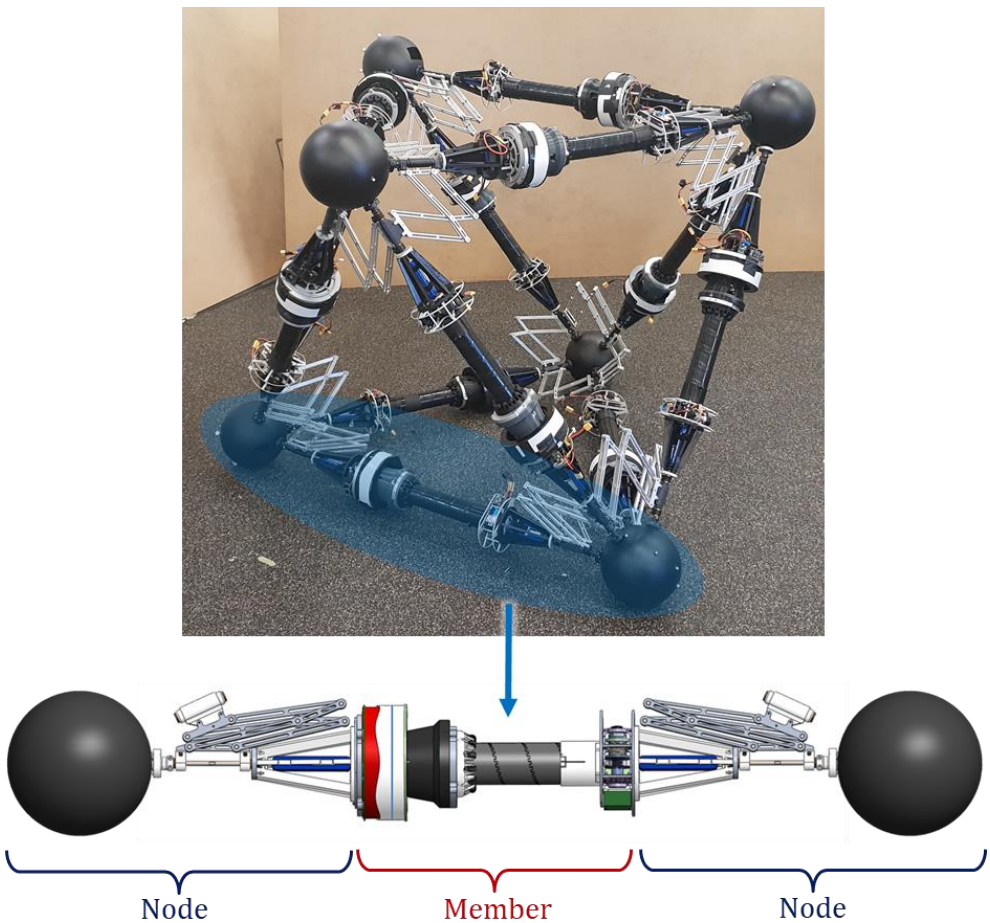
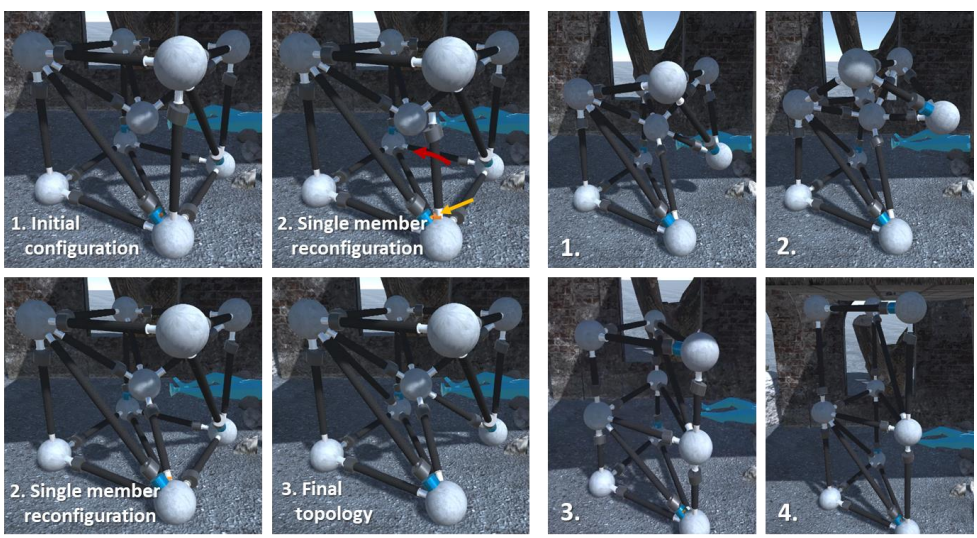


Fig. 2.2.13. Edge module

2.2.2 Master Member-End

Topology reconfiguration of VTT is conducted by moving a member to change connectivity between a member and a node. Fig. 2.2.14. revisit the reconfiguration example from cube topology to tower topology. For topology reconfiguration, the member detaches one of its ends from a node, move the end to attach to the other node. However, a member cannot be moves in the space on its own because it has only 1 DOF. That is why additional manipulator, Master Member-End is necessary to compensate for the insufficient DOF. For this purpose, Master Member-End has 2 DOF in spherical surface and has gripper to grip the member-end [7]. In addition, Master Member-End must maintain the Sphere so as not to interfere function of the node.



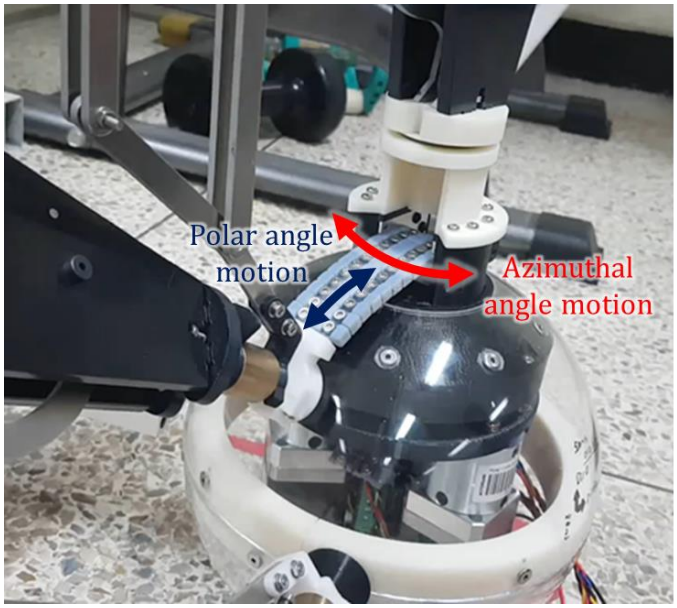
(a) (b)

Fig. 2.2.14. Reconfiguration example from cube topology to tower topology.
(a) Topology reconfiguration. (b) Geometric reconfiguration.

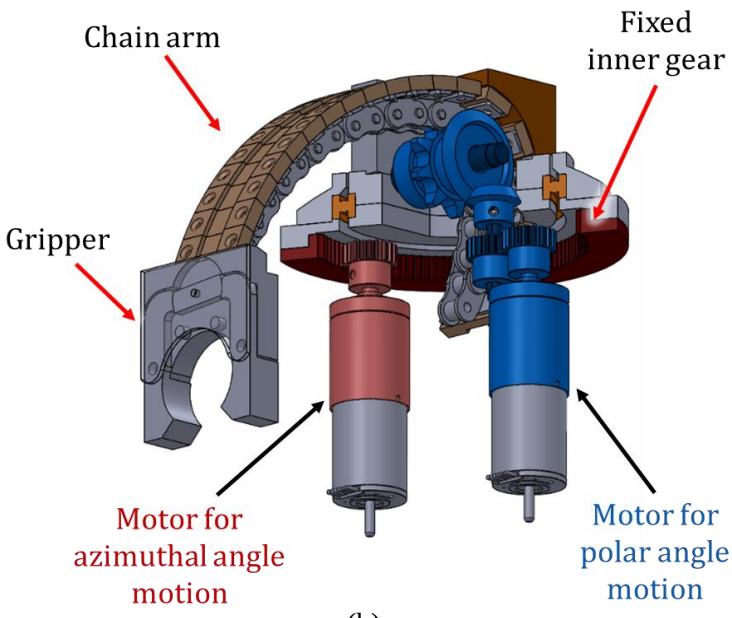
To satisfies all the requirements, Master Member-End was designed as in Fig. 2.2.15. The arm of Master Member-End is composed of one-way chain. Fig. 2.2.16. shows structure of one-way chain. One-way chain is made of attachment chain (Tsubaki Inc., RS25K-1-U.S.) and 3d-printed chain cover. The chain cover was designed in wedge shape to allow one-way chain to be bent only inward as in Fig. 2.2.16.(a) and (b). Thus, the Master Member-End arm can always lie on the Sphere surface as in Fig. 2.2.16.(c).

The Master Member-End arm moves in polar angle and azimuthal angle direction. In Fig.2.2.15.(b), the motor and transmission part of polar angle motion is colored with blue, and those of azimuthal angle motion is colored with red. Polar angle motion is achieved by pulling and push the one-way chain and azimuthal angle motion is achieved using fixed inner gear mechanism. At the end of the arm is a gripper to grip the Sphere Interface of Passive Member-End to move member for topology reconfiguration.

Overall process of moving a member for topology reconfiguration was tested as shown in Fig. 2.2.17. The process of moving a member is as follows: first, gripper approach to member and catch Sphere Interface of member-end, second, move the member to desired position and third, release the member. The test shows that Master Member-End arm can freely moves in spherical surface with 2 DOF while holding the member.



(a)



(b)

Fig. 2.2.15. Master Member-End.
(a) Master Member-End prototype with Passive Member-End.
(b) Master Member-End model

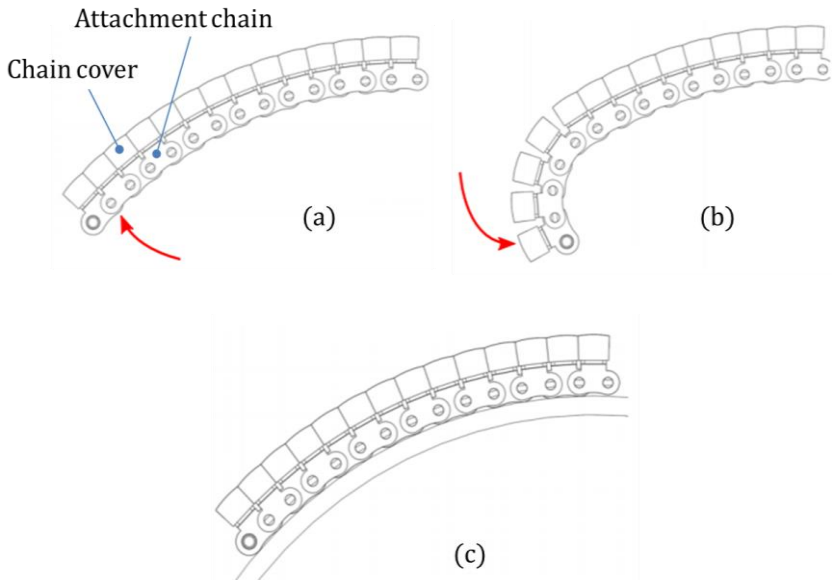


Fig. 2.2.16. Design of one-way chain

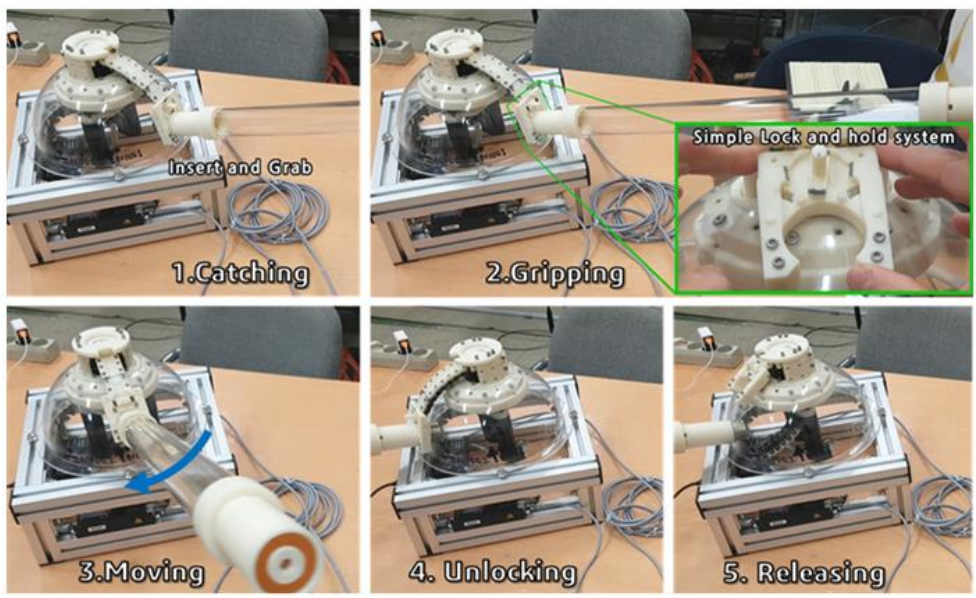


Fig. 2.2.17. Member moving test with Master Member-End

2.3 Control System

VTT is a cluster of unit modular robot, edge module. An edge module does simple 1 DOF motion. However, by combining them into VTT, they can do complex and sophisticated motion to conduct various works such as locomotion, reconfiguration and shoring. For VTT to conduct desired task, each edge module is coordinated by following hierarchy as in Fig. 2.3.1.

When the desired task is given, VTT configuration planner plan the configuration of VTT to conduct the task. Then, VTT configuration planner command the desired edge module length to each edge module controller and each edge module command desired motor position to achieve the desired edge module length. With this control hierarchy, VTT can perform any tasks with same control system regardless of the number of edge modules. This generality and scalability are the advantage of modular robot system:

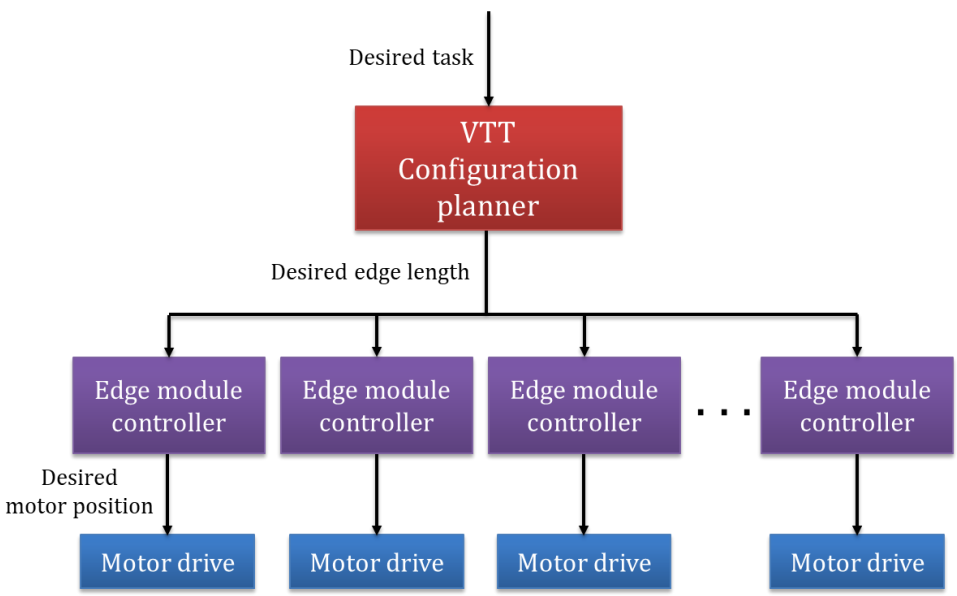


Fig. 2.3.1. Control hierarchy of VTT system

The main contribution of this thesis is the algorithm of VTT configuration planner for locomotion task, which will be covered in Chapter 4. In this section, I will explain how VTT and edge modules are controlled to follow the planned configuration. A more detailed VTT control system is represented by a diagram as shown in Fig. 2.3.2. In the diagram, the blocks that performed by computer, VTT and motion capture system is colored in red, blue and green respectively.

In this study, VTT configuration planner was implemented via MATLAB in computer with Intel i7-4790K CPU. The planner plans position trajectory of VTT's nodes. Because, position of nodes is more intuitive to plan and more easily observed using motion capture system for feedback control than length of edges. Position of node is defined as center of the Sphere and position of all nodes in VTT is represented by position vector x.

Position of the nodes was observed by VICON motion capture system. As in Fig. 2.3.3, the system tracks the position of reflective markers using multiple infrared cameras [26][27]. In VTT test bench, 27 infrared cameras were installed and 6 markers were attached on each Sphere to track the position of nodes.

When desired position of nodes, $x_{desired}$ and observed position of nodes, $x_{observed}$ are given, inverse kinematic solver calculates the length error of edge modules, L_{error} from position error of nodes x_{error}. Then, it determines command velocity of edge modules, $\dot{L}_{command}$ by PID gain and send to edge module controller by Wi-Fi module connected to the computer. When the edge module controller receives $\dot{L}_{command}$, it calculates the desired motor velocity v and command to the motor of Spiral Zipper. v can be easily calculated by dividing the $\dot{L}_{command}$ by pitch of the Spiral Zipper.

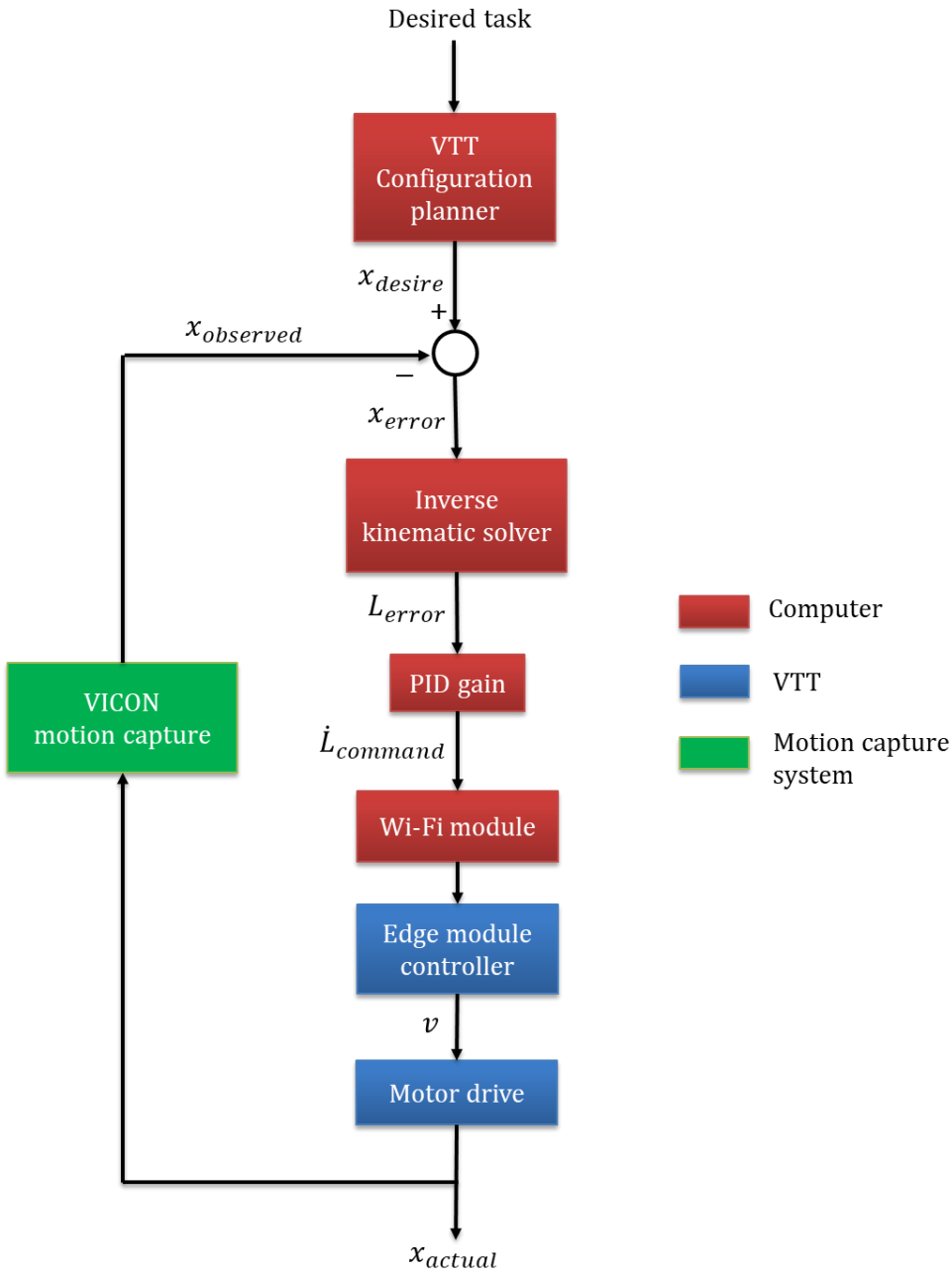


Fig. 2.3.2. Control diagram of VTT

The controller modules used in this control system are shown in Fig. 2.3.4. Sampling rate of sending data via Wi-Fi module was 20 Hz in case of controlling 12 edge modules.

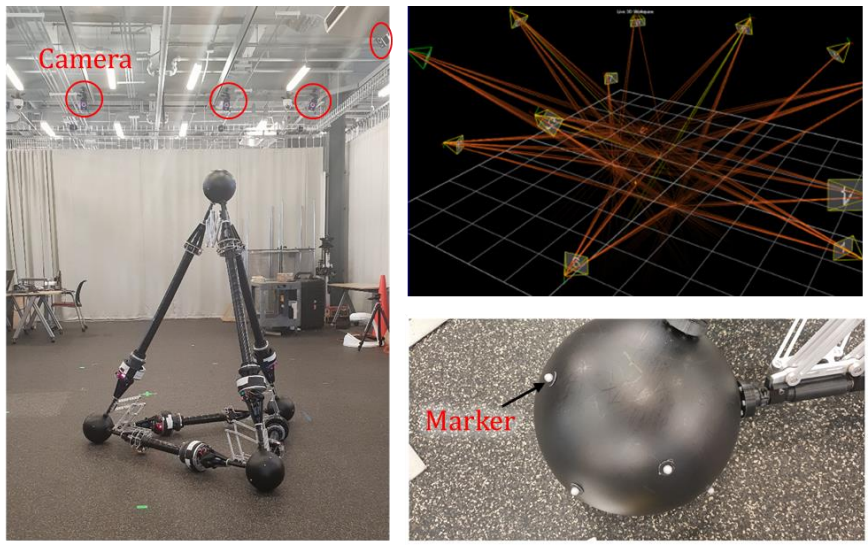


Fig. 2.3.3. VICON motion capture system. VTT in VICON system area (left). Software visualized VICON system (top right). Sphere with markers (bottom right)

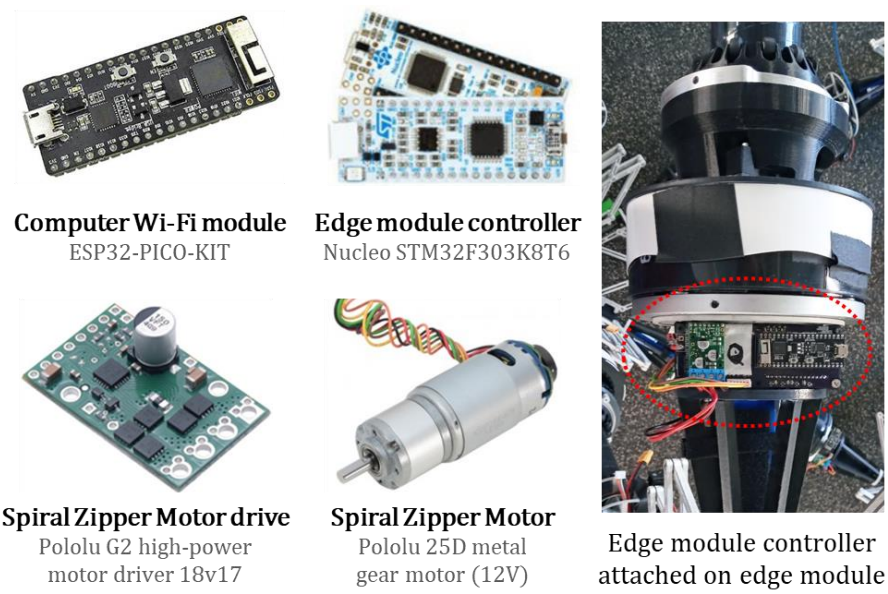


Fig. 2.3.4. Controller module of VTT

2.4 Node Position Control Experiment

By applying the controller system explained in previous section, node position control of VTT was verified by experiment. For the experiment, edge modules were assembled into octahedron VTT with 12 edges and 6 nodes as in Fig. 2.4.1. The initial position of each nodes was set as Table 2.1, where the node number is denoted in Fig. 2.4.1.

Table 2.1. Initial position of the nodes in octahedron VTT

Node no.	X (m)	Y (m)	Z (m)
1	-0.008	-0.428	0.100
2	0.384	0.654	0.100
3	1.124	-0.226	0.100
4	1.008	0.428	1.039
5	0.617	-0.654	1.039
6	-0.124	0.226	1.039

Fig. 2.4.1. Octahedron VTT for node position control experiment

In this experiment, three nodes on the top of the VTT was controlled to move upward and downward. The other three nodes on the bottom was set to be fixed to avoid unpredictable motion by ground friction effect. The desired z-coordinate trajectory of top nodes was set as follows.

$$z = \begin{cases} 1.039 - 0.025t & , \quad 0 \le t < 2 \\ 0.989 & , \quad 2 \le t < 12 \\ 0.989 + 0.025t & , \quad 12 \le t < 22 \\ 1.239 & , \quad 22 \le t < 32 \\ 1.239 - 0.025t & , \quad 32 \le t < 40 \end{cases}$$

x and y coordinate value of the nodes was set to be fixed. The desired trajectory was commanded to the nodes in the form of step function in every 0.5 sec.

Fig. 2.4.2 shows actual and desired position trajectory of the nodes. The average rms error of position of nodes was 0.0065 m. The result shows that VTT can follows desired node trajectory within reasonable error.

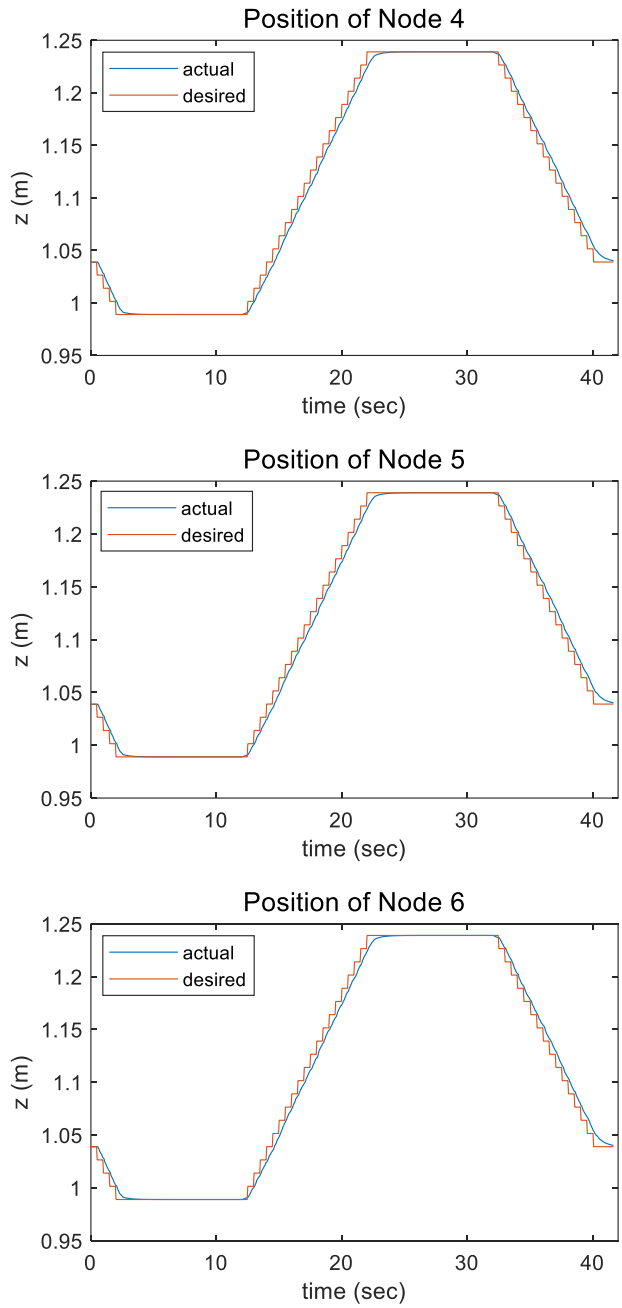


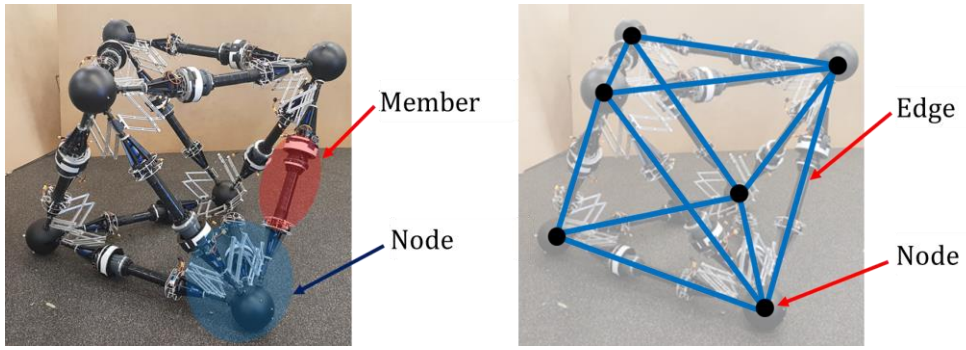
Fig. 2.4.2. Position graph of VTT position control experiment

Chapter 3 Mathematical Model of Variable Topology Truss

In previous chapter, hardware model of VTT was explained. Before discussing about locomotion algorithm, the mathematical model of VTT will be explained in this chapter. First, mathematical expression for geometric and topological configuration of VTT will be defined. Then, inverse kinematic will be explained. Finally, mechanical constraints of VTT will be explained and expressed in mathematical form.

3.1 Configuration and Terminology

Mechanically, VTT is composed of nodes and members as Fig. 3.1.1.(a), where a node consists of member-ends and a Sphere, and members consists of Spiral zipper and Tensioner. Mathematically, VTT is modeled as graph composed of nodes and edges as in Fig. 3.1.1.(b). However, mathematical meaning of node is different from mechanical meaning. Mathematically, node is vertex that represent center of Sphere and edge is line that connect two nodes.



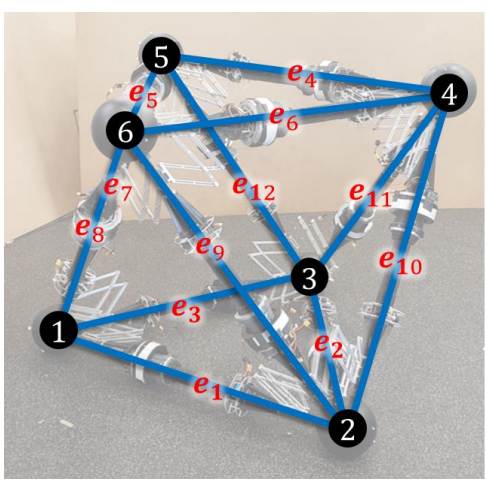
(a) (b)

Fig. 3.1.1. VTT model: (a) Mechanical model. (b) Mathematical model

Topological configuration of VTT is defined by the connectivity between nodes, which is represented as graph form:

$$G = (N, E) \tag{3.1}$$

where, $N = \{1, 2, \cdots, n\}$ is a set of nodes, and $E = \{e_1, e_2, \cdots, e_m\}$ is a set of edges, where, $e_k = \{i, j\}$ represent an edge that connect node i and node j. Fig. 3.1.2. shows an example of topology representation for octahedron VTT.



$$G = (N, E)$$

$$N = \{1, 2, 3, 4, 5, 6\}$$

$$E = \{e_1, e_2, e_3, \cdots, e_{11}, e_{12}\}$$

$e_1 = \{1,2\}$ $e_2 = \{2,3\}$ $e_3 = \{3,1\}$
$e_4 = \{4,5\}$ $e_5 = \{5,6\}$ $e_6 = \{6,4\}$
$e_7 = \{1,5\}$ $e_8 = \{1,6\}$ $e_9 = \{2,6\}$
$e_{10} = \{2,4\}$ $e_{11} = \{3,4\}$ $e_{12} = \{3,5\}$

Fig. 3.1.2. Representation of octahedron topology VTT

Geometric configuration of VTT is defined by position of nodes and connectivity between nodes. Position of node i is represented by vector as follows.

$$p_i = [\, p_{ix}, p_{iy}, p_{iz} \,]^T \tag{3.2}$$

The position vector of all nodes is represented by concatenating the position vectors as follows.

$$x = [\, p_{1x}, \cdots, p_{nx}, p_{1y}, \cdots, p_{ny}, p_{1z}, \cdots, p_{nz}, \,]^T \tag{3.3}$$

Two nodes connected by an edge are called adjacent nodes. $N_{adj(i)}$ is a set of nodes adjacent to node i. Edges that shares a node are called adjacent edges. For example, in Fig. 3.1.2, $N_{adj(2)} = \{1, 3, 4, 6\}$ represent adjacent nodes of node 2 and edges e_1, e_2, e_9, e_{10} are adjacent each other sharing node 2.

3.2 Inverse Kinematics

In VTT, inverse kinematics is the calculation of length of edges when position of nodes is given. There have been several works about general methods of solving forward and inverse kinematics of truss type robot [28]-[30]. However, in this study, all nodes are actively controlled and inverse kinematics can be calculated simply as follows:

$$L_k = \left\| p_i - p_j \right\| \tag{3.4}$$

where $\| \ \|$ denotes 2-norm and L_k denotes the length of edge $e_k = \{i, j\}$. By taking the derivative of Eq. (3.7), the relation between the velocity of the edge length and velocity of node can be linearized as follows:

$$\dot{L}_k = \frac{(p_i - p_j)^T \dot{p}_i + (p_j - p_i)^T \dot{p}_j}{L_k} \tag{3.5}$$

By calculating the Eq. (3.8) for every edge, the relation between the velocity of all edges $\dot{L} = [\dot{L}_1, \dot{L}_2, \cdots, \dot{L}_n]^T$ and nodes can be represented in matrix form as follows:

$$\dot{L} = R\dot{x} \tag{3.6}$$

where R is Jacobian matrix between velocity of edges and velocity of nodes.

3.3 Constraints

In order to do locomotion planning for applying to actual VTT, all the constraints should be carefully considered during planning. In this section, all mechanical constraints will be defined and mathematically modeled based on the hardware design introduced in Chapter 2.

Length of Edges

As explained in Section 3.1, edge is defined as line connecting two nodes. And mathematically node is located in center of Sphere. Thus, length of edge is determined by distance between centers of two nodes on the edge as in Fig. 3.3.1. Length of each edge L_k is confined by minimum length and maximum length of edge module which represented as follows.

$$L_{\min} \le L_k \le L_{\max} \tag{3.7}$$

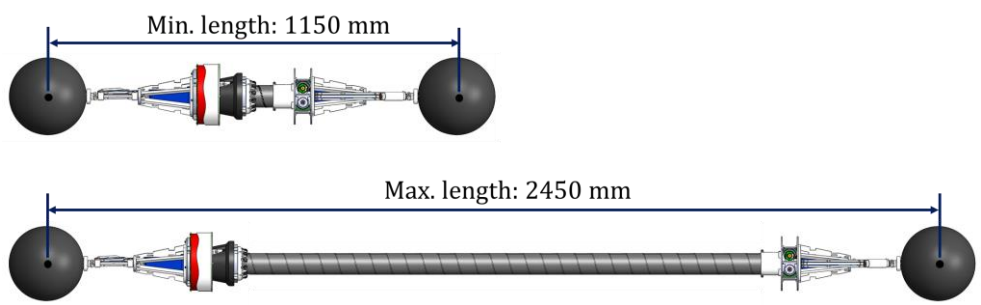


Fig. 3.3.1. Length of edge

Angle between Adjacent Edges

As in Fig. 3.3.2, Passive Member-End has finite angle range which constraints angle between adjacent edges. The angle between adjacent edges at node i is calculated using direction cosine. The angle constraint is represented as follows.

$$\theta_{\min} \le \theta_{i,j,k} = \cos^{-1}\left(\frac{(p_i - p_j)^T (p_i - p_k)}{\|p_i - p_j\| \|p_i - p_k\|}\right) \le \theta_{\max} \tag{3.8}$$

where j and k are adjacent nodes of node i. Because of the structure of Passive Member-End, minimum angle constraint is applied to all adjacent members, but maximum angle constraint is only applied to the adjacent members that connected by a Passive Member-End.

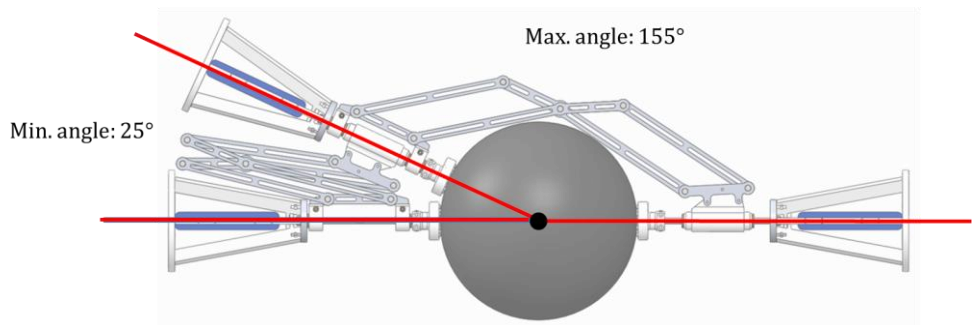


Fig. 3.3.2. Minimum and maximum angle between adjacent edges

Dihedral Angle

When two Passive Member-Ends are folded each other closely, a collision can occur as in Fig. 3.3.3.(a). In order to avoid collision, angle between planes of two passive members, dihedral angle should be larger than certain angle.

Fig. 3.3.3.(b) shows schematic for calculating dihedral angle. Meaning of each variable are listed as follows.

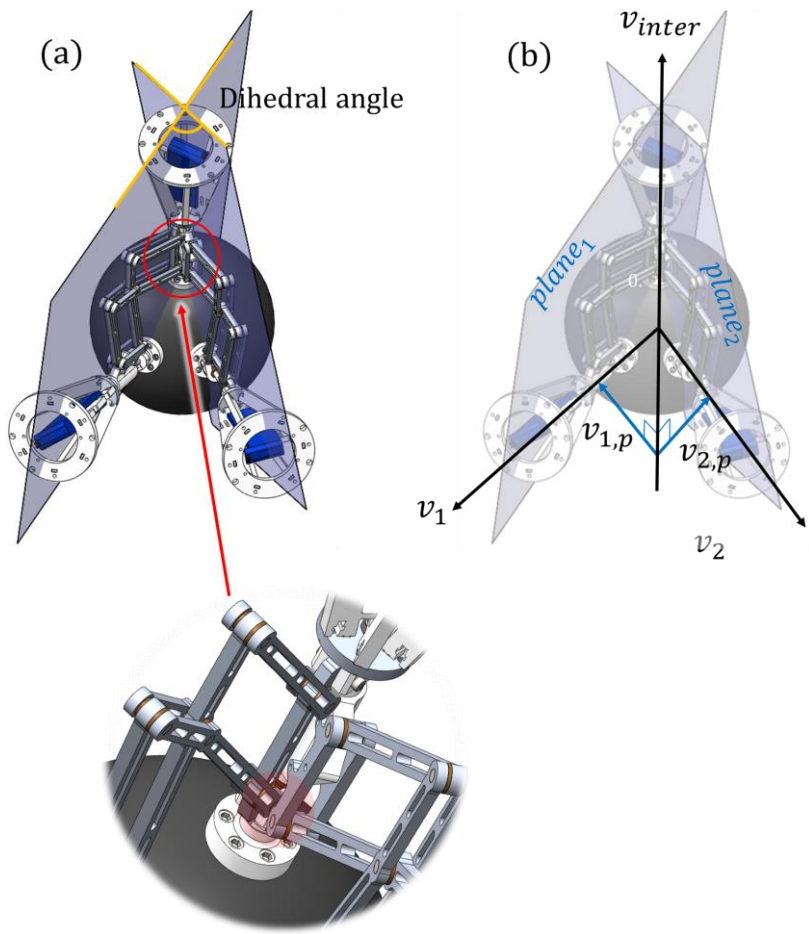


Fig. 3.3.3. Schematic of dihedral angle.
(a) Linkage collision at minimum dihedral angle,
(b) Schematic for dihedral angle calculation

$plane_1$ and $plane_2$: Plane formed by each Passive Member-End

v_{inter} : Vector that lies on the edges connected by two Passive Member-Ends

v_1, v_2 : Vectors that lies on the edges connected by each Passive Member-End

$v_{1,p}$, $v_{2,p}$: Vectors that lies on each plane and perpendicular to v_{inter}

Here, dihedral angle is defined as the angle between $plane_1$ and $plane_2$ which is same as angle between $v_{1,p}$ and $v_{2,p}$.

$v_{1,p}$ and $v_{2,p}$ can be calculated using projection vector as follows.

$$v_{1,p} = v_1 - \frac{v_1 \cdot v_{inter}}{\|v_{inter}\|^2} v_{inter} \tag{3.9}$$

$$v_{2,p} = v_2 - \frac{v_2 \cdot v_{inter}}{\|v_{inter}\|^2} v_{inter} \tag{3.10}$$

Then, dihedral angle θ_{dh} is calculated using direction cosine between $v_{1,p}$ and $v_{2,p}$. Minimum dihedral angle constraint can be represented as follows.

$$\theta_{min,dh} \le \theta_{dh} = \cos^{-1}\left(\frac{v_{1,p} \cdot v_{2,p}}{\|v_{1,p}\|\|v_{2,p}\|}\right) \tag{3.11}$$

Collision between Non-Adjacent Edges

To avoid collision between non-adjacent edges, there should be enough distance between them. Fig. 3.3.4. shows an example of non-adjacent edges. In the figure, dark and light blue edges are a set of non-adjacent edges and red arrow denotes minimum distance between two edges. Thus, the following equation should be satisfied for collision avoidance,

$$d(e_k, e_l) \geq d_{\min} \tag{3.12}$$

where $d(e_k, e_l)$ is the minimum distance between non-adjacent edges, e_k and e_l. $d_{\min}$ is the allowable minimum distance, which is same as maximum diameter of members. For distance calculation, analytical method presented in [32] was used.

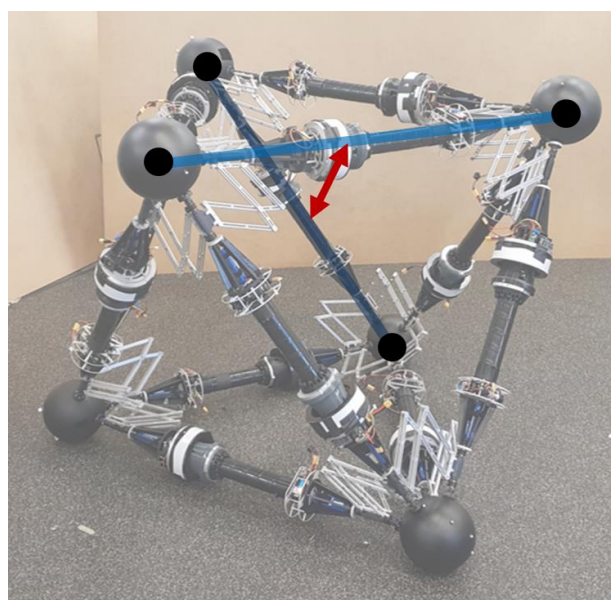


Fig. 3.3.4. Example of minimum distance between non-adjacent members

Ground Contact

In usual case, Sphere is the only component of VTT that interfere with ground, because the diameter of Sphere is larger than that of member. Other components such as member and Passive Member-End can contact to ground if the ground is complex. To avoid complexity, I assumed that the ground is flat enough that only Sphere is contact to ground. I also assumed that the ground is convex. Then, shape of ground can be generally expressed as $z = f(x, y)$. Thus, in order to avoid collision to ground, the following equation should be satisfied.

$$p_{iz} \geq f(p_{ix}, p_{iy}) \tag{3.13}$$

where p_{iz} is z-coordinate of node position

Strength of Member

By weight of VTT's components, each member is subjected to tensile or compressive load. Excessive load makes members cannot be actuated or even be broken. In this study, I defined strength of member as follows.

Strength: Maximum load that induce irreversible damage that make member cannot be actuated again

Actuation strength: Maximum load that make member cannot be actuated, but can be actuated again when the load is removed.

Compressive load and tensile load at VTT members have different characteristic.

In case of compressive load, strength is larger than actuation strength. In other words, member still can maintain its length without damage if it only exceeds actuation strength. This is due to nature of band and Friction Drive of Spiral Zipper. Band of Spiral Zipper has high strength in compression [16]. However, even with the lower load, Friction Drive gets stuck and does not works due to excessive friction generated between the band and Slider in Friction Drive.

In case of tensile load, strength is same as actuation strength. This is due to nature of band and Tensioner of Spiral Zipper. Unlike compression, band of Spiral Zipper has very low tensile strength. Therefore, it needs tensile preload by Tensioner to resist tensile load. Thus, the member is immediately broken when tensile load higher than maximum force of Tensioner is applied.

In order to check if the load at member exceed the strength, VTT load calculation method was developed. In this method, load applied to Friction Drive and Tensioner was calculated respectively. As explained, compressive actuation

strength is determined by compressive force applied to Friction Drive and tensile actuation strength is determined by tensile force applied to Tensioner.

The load was calculated based on ‘method of joint’ in truss structure [31] assuming nodes which contact to ground is fixed. However, unlike typical truss structure, VTT has dominant self-weight: weight of Spiral Zipper, Tensioner and Passive Member-End. Thus, weight of edge module was also considered for truss force calculation. Schematic applied force at edge module is shown in Fig. 3.3.5. Measured mass and center of mass of each component are presented in Fig. 3.3.6. The meaning of variables in the figure are as follows,

F_1, F_2 : Force applied by nodes

P_1, P_2 : Axial force applied by nodes

V_1, V_2 : Shear force applied by nodes

$P_{tensioner}$: Axial force applied to Tensioner

P_{drive} : Axial force applied to Friction Drive

$\omega_{tensioner+ME}$: Weight of Tensioner and member-end

ω_{band} : Weight of Spiral Zipper’s band

$\omega_{drive+ME}$: Weight of Friction Drive and member-end

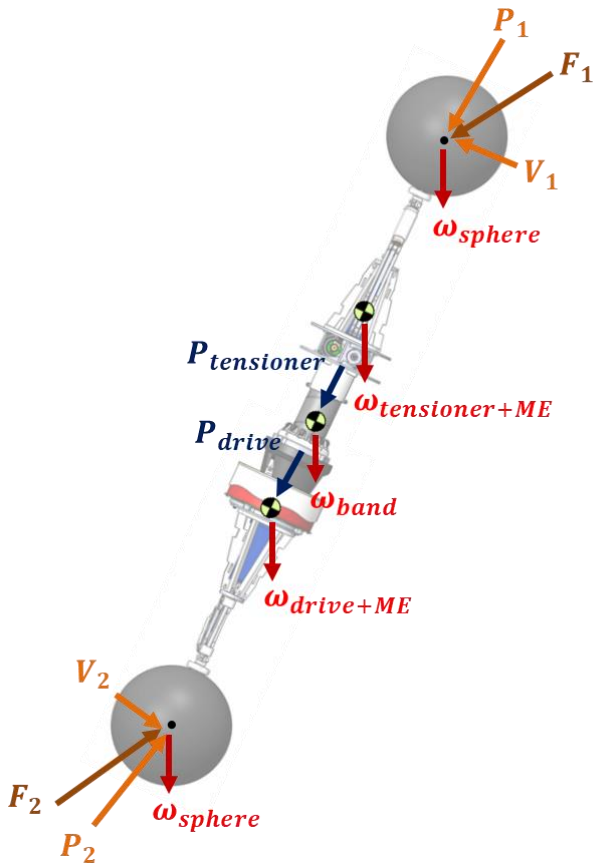


Fig. 3.3.5. Applied forces at edge module

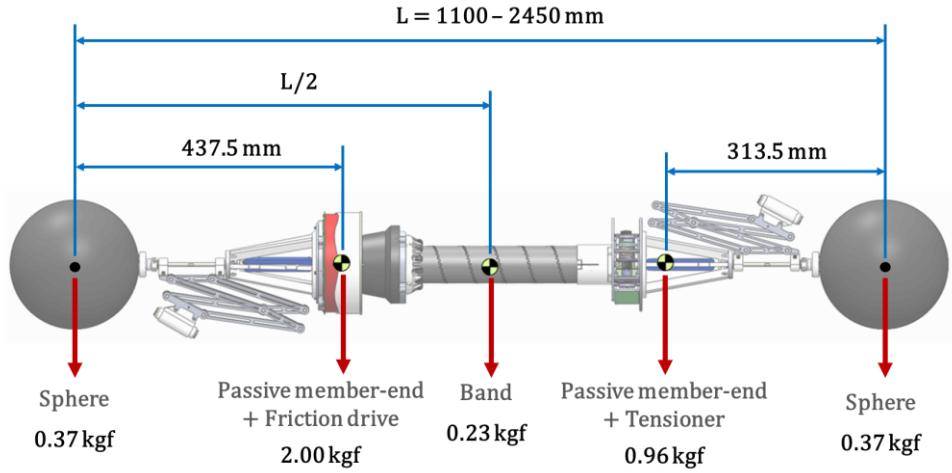


Fig. 3.3.6. Edge module's mass and center of mass

F_1, F_2 can be calculated by solving system of equations comes from 'method of joint' and 'force-moment equilibrium' for every node and member of VTT. P_1, P_2 and V_1, V_2 is calculated by resolving F_1, F_2 in direction of and in perpendicular direction of edge module respectively. Then, $P_{tensioner}$ and P_{drive} is calculated as follows.

$$P_{tensioner} = P_1 + (P_2 - P_1)\frac{\omega_{tensioner+ME}}{\omega_{drive} + \omega_{tensioner+ME} + \omega_{band}} \tag{3.14}$$

$$P_{drive} = P_1 + (P_2 - P_1)\frac{\omega_{tensioner+ME} + \omega_{band}}{\omega_{drive} + \omega_{tensioner+ME} + \omega_{band}} \tag{3.15}$$

The load calculation method was implemented by MATLAB and can be visualized as in Fig. 3.3.7.

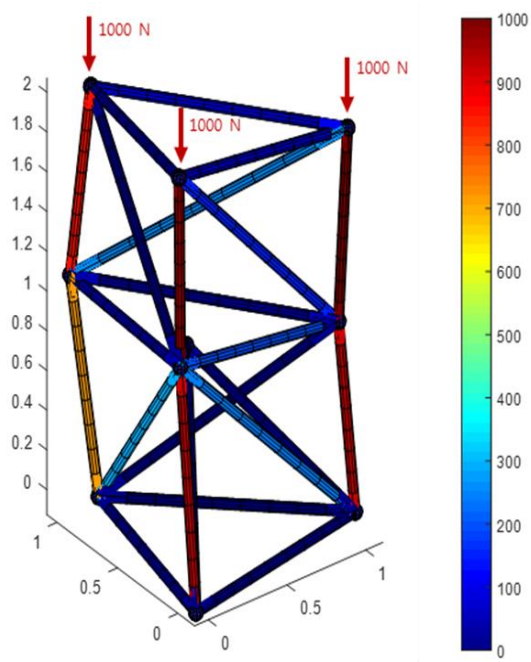
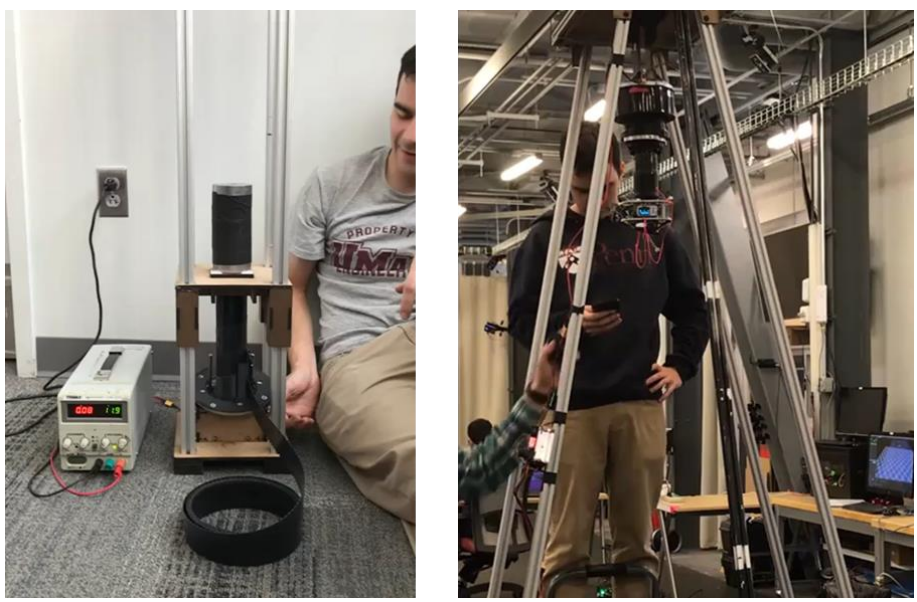


Fig. 3.3.7. VTT load calculation example

Actuation strength of member was measured using test bench shown in Fig. 3.3.8. Compressive actuation strength test bench is shown in Fig. 3.3.8.(a). In the testbench, weights were put on the top of Spiral Zipper and we measured maximum weight that Spiral Zipper can actuate. Tensile actuation strength test bench is shown in Fig. 3.3.8.(b). In the test bench, Spiral Zipper was upside down and weight were put on the bottom. Then, we measured maximum weight that Spiral Zipper can actuate.



(a) (b)

Fig. 3.3.8. Test bench for measuring actuation strength:
(a) Compressive actuation strength test bench
(b) Tensile actuation strength test bench

Maximum Velocity of Member

Maximum velocity that can member actuate varies depends on the compressive load applied on the member. Maximum velocity is lower in extension than in retraction direction, since it needs more motor torque to mesh the band teeth. The maximum velocity was measured in the worst case: velocity in extension when applied compressive actuation strength. It was measured using the same test bench used in compressive strength test as in Fig. 3.3.8.(a). The velocity constraint can be represented as follows,

$$\left|\dot{L}\right| = \left|R\dot{x}\right| \leq \dot{L}_{\max} \tag{3.16}$$

where R is Jacobian matrix between velocity of edges and nodes, and $\dot{L}_{\max}$ is maximum velocity of member measured in the worst case.

Fixed Nodes

In this study, ground friction was assumed to be large enough to fix the nodes on the ground. In addition, in order to avoid unpredictable motion of the nodes caused by ground friction effect, nodes on the ground will not actively moves. Thus, position of the nodes on the ground nor length of the members connecting those nodes was set to be fixed. The fixed nodes constraint can be represented as follows,

$$C\dot{x} = 0 \tag{3.17}$$

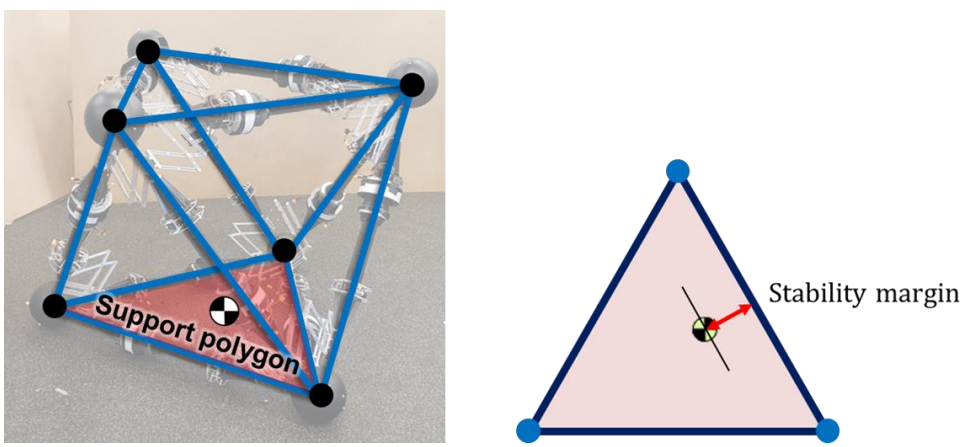
where, $\dot{x}$ is velocity vector of all nodes and C is constraint Jacobian matrix to fix the motion of fixed nodes. C is composed of 0 and 1. For example, if the fixed nodes are 1, 2 and 3, C can be written as $9 \times 3n$ matrix as follows.

$$C = \begin{bmatrix} \overbrace{1 & 0 & 0 & \cdots}^{x\text{ coordinate}} & \overbrace{0 & 0 & 0 & \cdots}^{y\text{ coordinate}} & \overbrace{0 & 0 & 0 & \cdots}^{\text{z coordinate}} \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots \end{bmatrix}$$

3.4 Stability Criteria

VTT must maintain stability to be controllable and avoid tumbling to the ground. VTT is said to be stable if it maintains its configuration without control input. In this study, VTT is assumed to moves quasi-statically, because maximum velocity of member is so low (less than 45 mm/s) that dynamic effect is negligible. In this system VTT is stable when it satisfies static stability criteria.

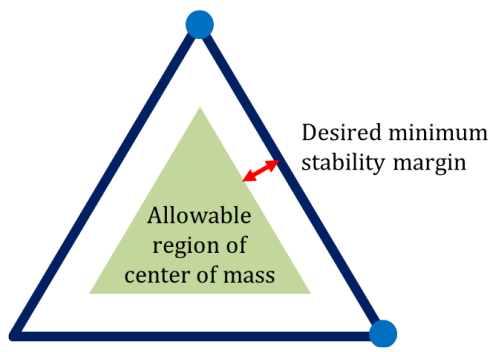
In static stability criteria, the robot is stable if the center of mass projected on the ground is inside support polygon. Support polygon of VTT is the polygon or convex hull which vertices are nodes that contact to ground as in Fig. 3.4.1. The center of mass can be mathematically calculated based on the mass information presented in Fig. 3.3.6.



(a) (b)

Fig. 3.4.1. (a) Support polygon of VTT. (b) Schematic of stability margin

However, actual center of mass can have errors which comes from control error and deflection of hardware components. To compensate these errors, minimum stability margin was given. Stability margin is minimum distance from the center of mass to the edges of support polygon as in Fig. 3.4.1.(b). Stability margin of VTT must be larger than desired minimum stability margin. To satisfy minimum stability margin constraint, center of mass must be inside allowable region of center of mass denoted by green shade as in Fig. 3.4.2. Desired minimum stability margin is decided considering the accuracy of mechanical and control system.



3.4.2. Schematic of allowable region of the center of mass

Chapter 4

Locomotion Algorithm

In this chapter, stable rolling locomotion algorithm will be suggested based on the kinematics and constraints explained in previous chapter. Here I revisit the objectives of the locomotion algorithm addressed in Section 1.3 as follows.

1. Plan optimal locomotion trajectory with high planning success rate.

2. Avoid obstacles

3. Prevent VTT from being damaged from the ground

As inputs, initial configuration and goal position of VTT will be given, and position and shape of obstacle is assumed to be known by vision sensor. The concept of stable rolling locomotion algorithm to achieve the objectives will be explained in next section.

4.1 Concept of Locomotion Algorithm

4.1.1 Method for Successful Planning and Obstacle Avoidance

In the previous research, Usevitch et al. suggested useful optimization based rolling locomotion algorithm that can be applied to any topology and locomotion path [23]. Usevitch's algorithm optimize the velocity of nodes at every time step to follow desired center of mass trajectory.

However, Usevitch's algorithm easily fails planning, because the shape of the robot tends to be distorted and approaches constraints and as it moves forward as in Fig. 4.1.1. The cause of this problem is that this algorithm is greedy algorithm which makes the locally optimal choice in that moment [33]. In Usevitch's algorithm, although the configuration is optimal at a time step, it may not be globally optimal and may leads to shape distortion. Therefore, higher-level planning is needed to guide the robot to maintain good-conditioned shape and makes it far from the constraints.

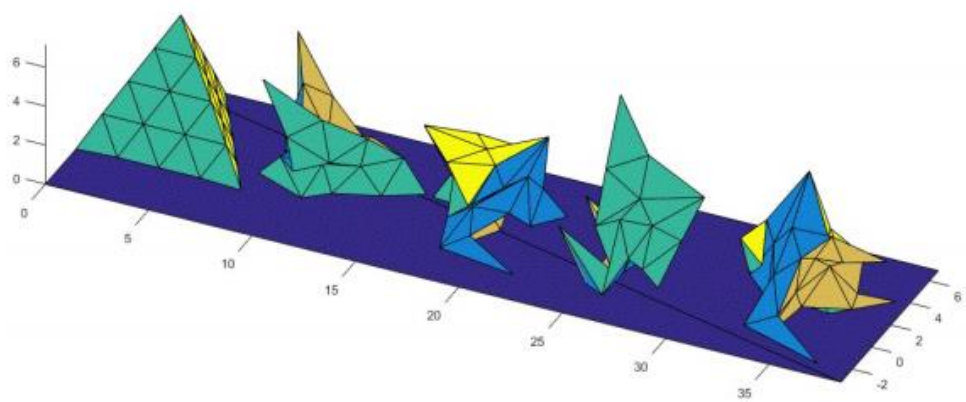


Fig. 4.1.1. Locomotion simulation with optimization method by Usevitch [23]

To analyze the reason of shape distortion, I performed locomotion simulation for octahedron topology by applying Usevitch's algorithm [9]. Fig. 4.1.2 shows motion of the robot in the simulation and Fig. 4.1.3 shows minimum and maximum length of the robot during the locomotion. Initial length of edges was set to be 1 and minimum and maximum length constraint was set to be 0.5 and 2 respectively. Initially, the shape of robot is regular octahedron and farthest from the constraint. However, as it moves, the shape get distorted and approaches the constraint which is worst at 470 second and 530 second. In those time, it can be seen that support polygon is distorted a lot from regular triangle and the desired center of mass is located near the edge of support polygon. This shows that improper support polygon and center of mass trajectory distort the shape of robot.

As a comparison, I performed locomotion simulation by giving desired support polygon which is close to regular triangle. The other conditions were set as same as previous simulation. Fig. 4.1.4 shows motion of the robot in the simulation and the green triangles in the figure denotes desired support polygon. Fig. 4.1.5 minimum and maximum length of the robot during the locomotion. Unlike the previous simulation, the robot maintained a shape close to regular octahedron and lengths was far from constraints. Being far from constraints implies that it has more space to search feasible configuration and more likely to succeed the planning.

In conclusion, proper support polygon planning and center of mass trajectory planning makes the robot to maintain good-conditioned shape. This makes the planning more likely to success even in complex and large environment.

In addition, support polygon is a good criterion to check obstacle collision, since it is the area where the robot stands. Thus, obstacle can be avoided by properly planning support polygon.

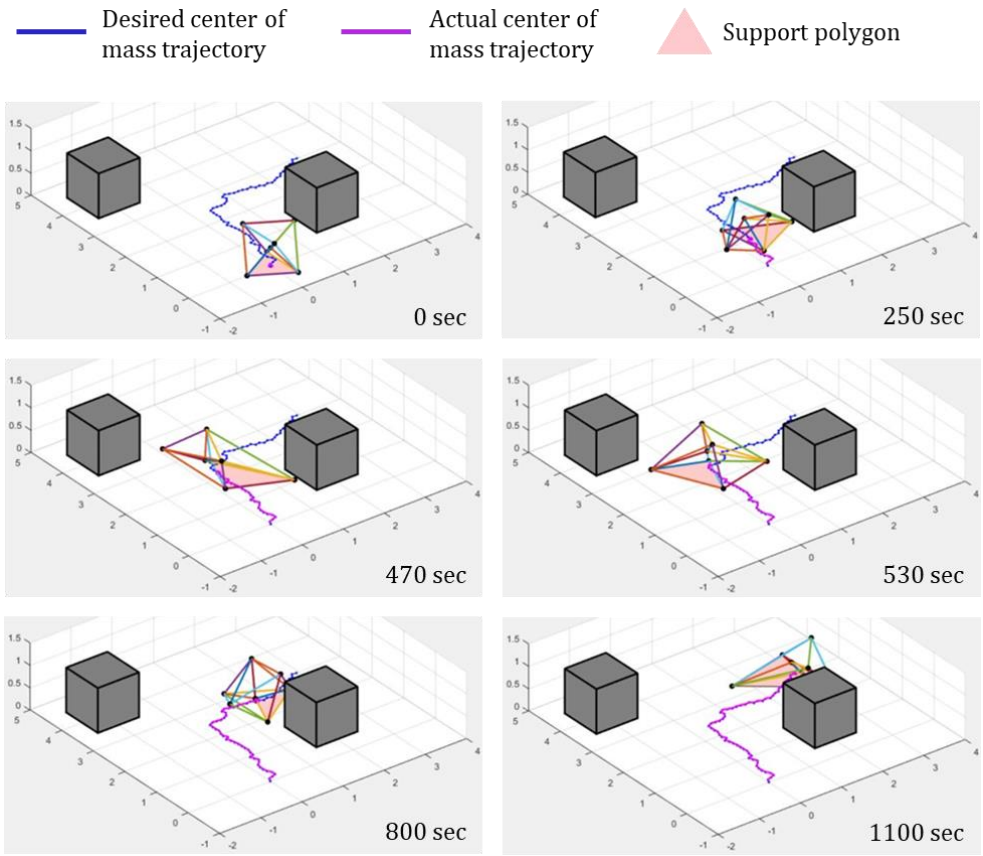


Fig. 4.1.2. Center of mass trajectory following locomotion simulation with octahedron topology

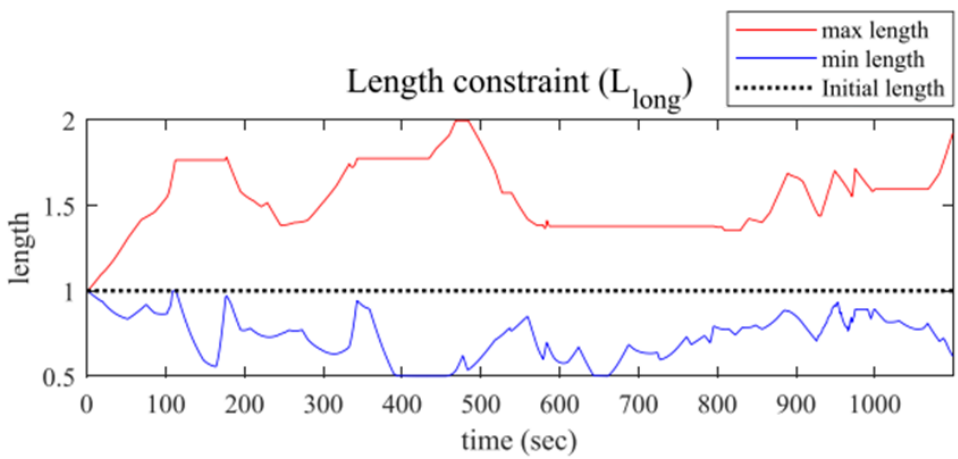


Fig. 4.1.3. Maximum and minimum member lengths in locomotion simulation

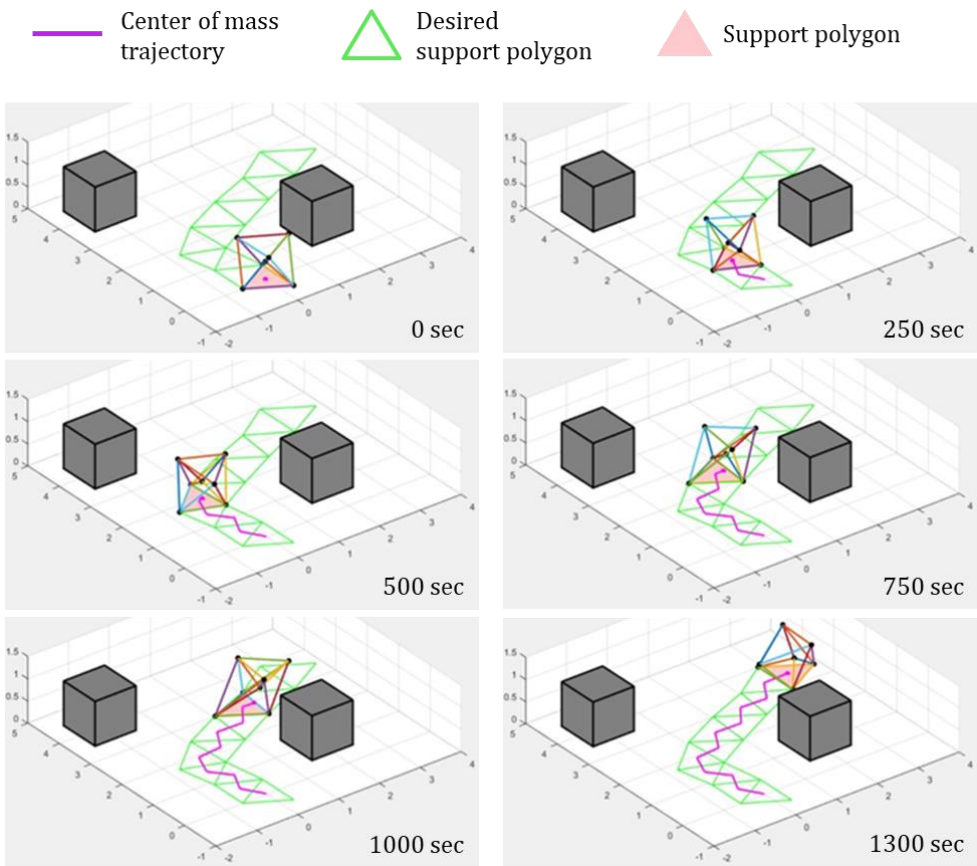


Fig. 4.1.4. Center of mass trajectory following locomotion simulation with octahedron topology

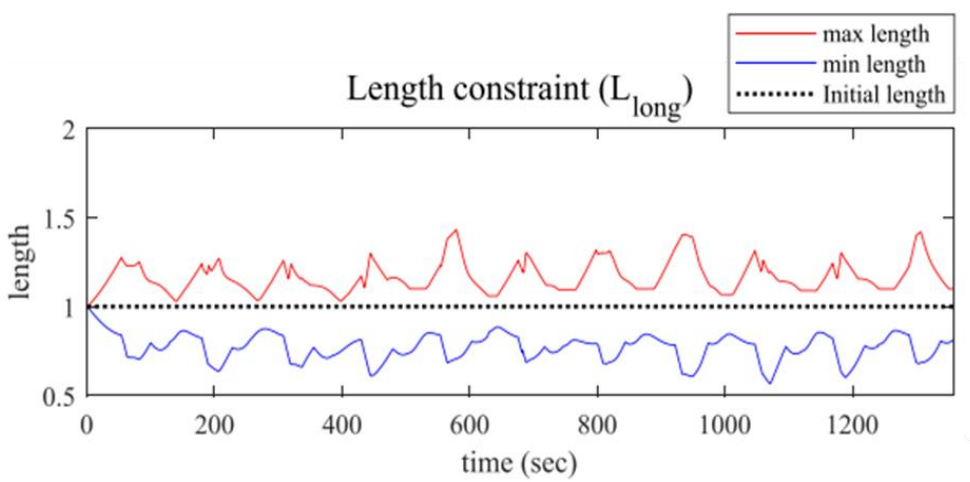


Fig. 4.1.5. Maximum and minimum member lengths in locomotion simulation

4.1.2 Method to Prevent Damage from the Ground

VTT is composed of many sophisticated components such as Spiral Zippers, Tensioners and Passive Member-Ends. Thus, a sudden collision or impact can cause significant damage to VTT. Especially, during rolling locomotion, VTT must roll stably without tumbling to prevent impact and damage from the ground. This is achieved by maintaining static stability using Non-Impact Rolling locomotion.

Depending on whether the robot is subjected to impact from the ground, I divided rolling locomotion into two types, Impact Rolling and Non-Impact Rolling [8]. Fig. 4.1.6. presents schematic of the two rolling. As in Fig. 4.1.6. (a) the motion of Impact and Non-Impact Rolling is same before frontal node start landing. After that time, as in Fig. 4.1.6. (b) Impact Rolling loses stability and tumble to the ground which exert impact to VTT. On the contrary, as in Fig. 4.1.7. (b) Non-Impact Rolling maintain stability and smoothly land on the ground without tumbling and impact from the ground.

Therefore, to prevent damage from the ground, proposed algorithm will plan the trajectory of nodes to moves in Non-Impact Rolling.

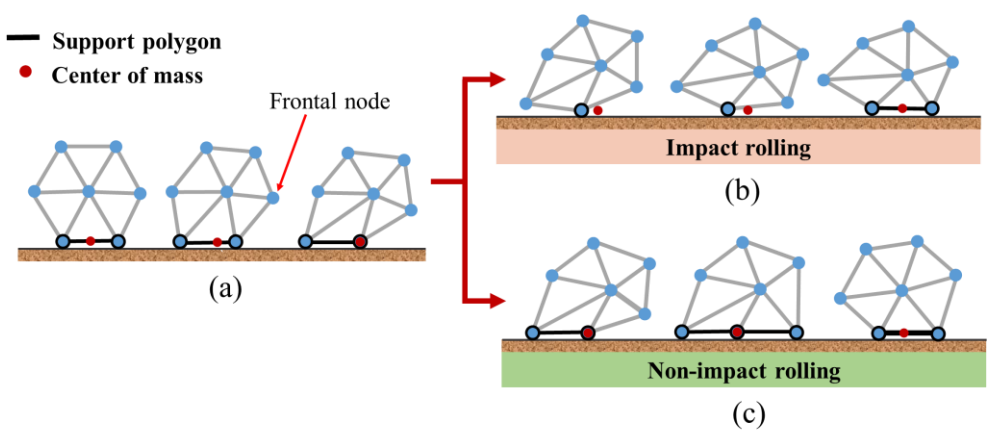


Fig. 4.1.6. Schematic of Impact Rolling and Non-Impact Rolling.
(a) Configuration before frontal node landing.
(b) Impact Rolling motion. (c) Non-Impact Rolling motion

4.1.3 Steps of Locomotion Algorithm

To accomplish objectives of the locomotion, proposed algorithm is divided into three steps as follows.

Step 1: Support polygon planning

Step 2: Center of mass planning

Step 3: Node position planning

In Step 1, support polygon planning, path of support polygon is planned for VTT to follow as it rolls. The example of the planned path is shown Fig. 4.1.7 Green triangles in the figure denote the support polygon to follow. VTT can avoid obstacle by support polygon planning.

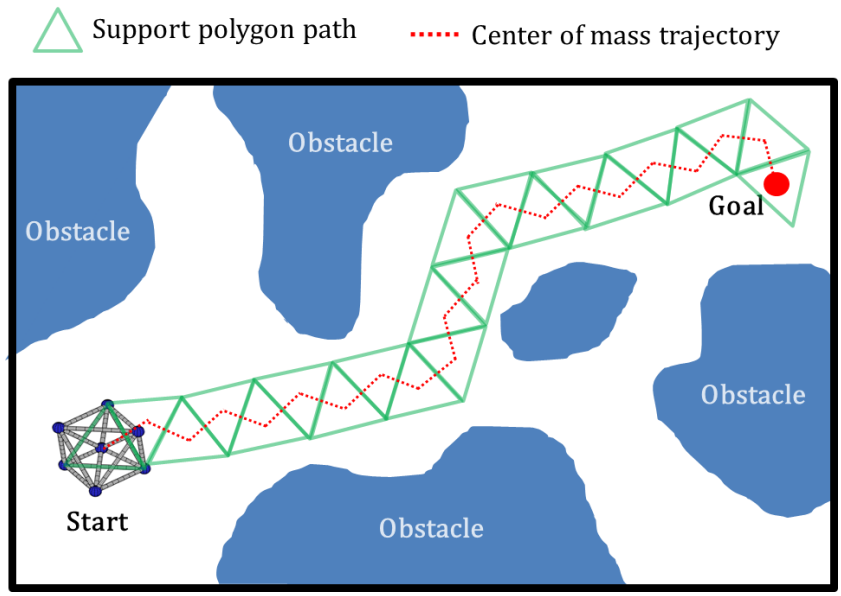


Fig. 4.1.7. Schematic of support polygon path and center of mass trajectory

In Step 2, center of mass planning, trajectory of VTT's center of mass is planned. Red dotted line in Fig. 4.1.7 shows an example of center of mass trajectory. By support polygon planning and center of mass planning, proposed algorithm can successfully plan locomotion trajectory that travel complex and large environment.

In Step 3, node position planning, position of VTT's node are planned to follow support polygon path and center of mass trajectory. The node position planning is based on optimization and Non-Impact Rolling algorithm is applied to prevent tumbling to the ground.

4.2 Support Polygon Planning

Objectives of support polygon planning is generating path of support polygon when initial support polygon, goal position and obstacles is given. Since this planning is non-convex and complex problem, it is effective to solve by using random search algorithm. For support polygon planning, I will propose new random search algorithm, Polygon-Based Random Tree (PRT).

4.2.1 Polygon-based Random Tree (PRT) Algorithm

Polygon-Based Random Tree (PRT) is motivated from Rapidly-Exploring Random Tree (RRT). As in Fig. 4.2.1, RRT tree is composed of vertices and edges. RRT start with initial vertex x_{init} and grws its tree by randomly pick a vertex x and find x_{near} to generate new vertex and edge toward x to the length of step size ε. RRT is an efficient sampling scheme to quickly search high-dimensional spaces that have constraints such as obstacles [34][35].

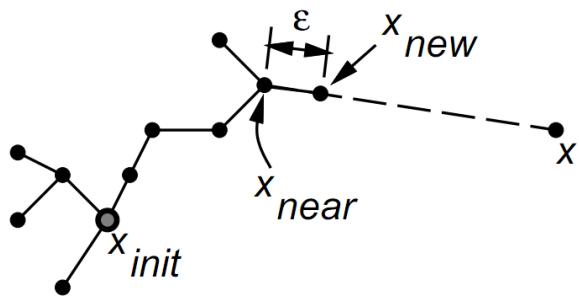


Fig. 4.2.1. Schematic of RRT structure [35]

PRT adopted the tree concept in RRT. However, PRT tree is composed of polygon and edge. The tree structure of PRT is represented as follows.

$$T = (Polygon, E) \tag{4.1}$$

$$Polygon = \{polygon_1, polygon_2, \cdots, polygon_n, \} \tag{4.2}$$

$$polygon_i = \{foot_{i1}, foot_{i2}, foot_{i3}\} \tag{4.3}$$

$$E = \{e_1, e_2, \cdots, e_m\} \tag{4.4}$$

$$e_k = \{polygon_i, polygon_j\} \tag{4.5}$$

In the tree T, $Polygon$ is a set of polygons, and E denotes an edge of the tree structure, which implies the relationship between parent and child. $polygon_i$ contains a set of foots which is vertices of $polygon_i$.

Algorithm 1 is a pseudo code of PRT algorithm. For each cycle, an objective point of the algorithm is randomly selected within the operational space. To ensure the algorithm converges well, the objective point is sometimes set as the goal position. The probability to select the goal position is $MixingFactor$ in line 3 of Algorithm 1.

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 $Foot\_Alter = GenerateFootAlter(T)$
9 $Temp\_Foot = FindNearestFoot(Obj\_Position, Foot\_Alter)$
// Making alternative polygon with new foot
10 $Polygon\_new = MakePolygon(Temp\_Foot)$
11 **If** $ObstacleCollision(Polygon\_new) == false$ **then**
// Add new polygon to the tree
12 $T.AddPolygon(Polygon\_new)$
13 $T.AddEdge(Polygon\_pre, Polygon\_new)$
14 **end**
15 **if** $Goal\_Position \in Polygon\_new$ **then**
16 $Reach\_Goal == true$
17 **end**
18 **end**
19 **return** T

After the objective position is selected, the new *Polygon* is expanded from the previous *Polygon*s sharing two *Foot* . Fig. 4.2.2 shows the schematic of the polygon tree expansion.

First, the alternatives of the next *Foot* are generated around the previous *Polygon* (line 8 in Algorithm 1). The distances between a foot alternative and two *Foot*s of the previous *Polygon* are set as the same distance, L_{nom} . L_{nom} is nominal length of VTT's edge which represents desirable length of an edge for good-conditioned shape of VTT.

Second, the nearest foot alternatives from the objective point is selected for setting the next *Foot* of the next polygon (line 9 in Algorithm 1). The PRT provides flexibility for generating next *Polygon* by permitting a slight distortion within the distortion margin, which is denoted as a circle in Fig. 4.2.2. The next *Foot* , which is drawn as a red dot in Fig. 4.2.2, is on the closest position within the distortion margin circle.

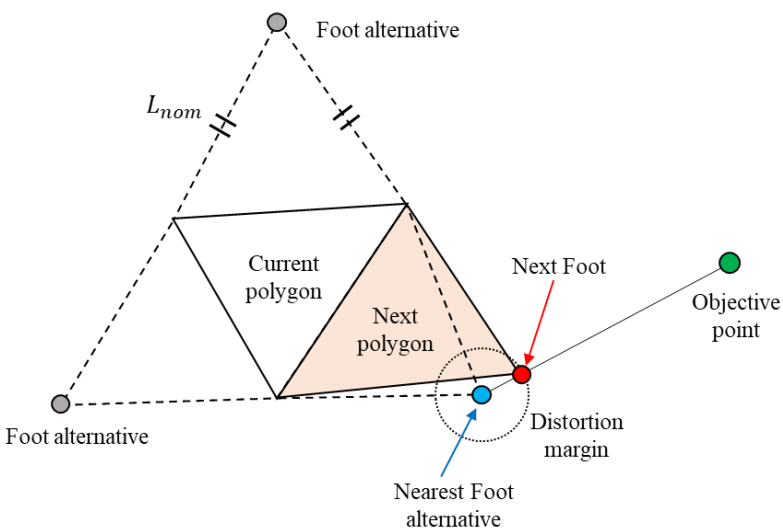


Fig. 4.2.2. Schematic of finding next support polygon

Third, next polygon is generated and add to PRT tree if it does not collide with obstacles collision (line 10-14 in Algorithm 1). Method for checking obstacle collision may vary depending on the topology and constraints of the robot. In this section, I will assume that obstacle collision occurs when the edge of next polygon and edge of obstacles intersect.

PRT tree grows as above process are repeated, and the algorithm ends when the last *Polygon* includes the goal position. After obtaining the tree from the PRT algorithm, the path of support polygon can be found by tracking the parent *Polygon* from the last *Polygon*.

Fig. 4.2.3 shows an example of support polygon planning in various distortion margin. In the plots, start point is marked with red dot in lower left corner and goal point is marked with red dot in upper right corner. Blue dot denotes foot alternatives and blue triangles denotes tree of polygons generated during PRT. Green triangle denotes planned path of support polygon. Objective point generated until the support polygon including goal point is generated. The algorithm stops when the objective point generated 400 times. In case of distortion margin 0 and 0.15, path of support polygon was generated to reach nearest polygon to goal point.

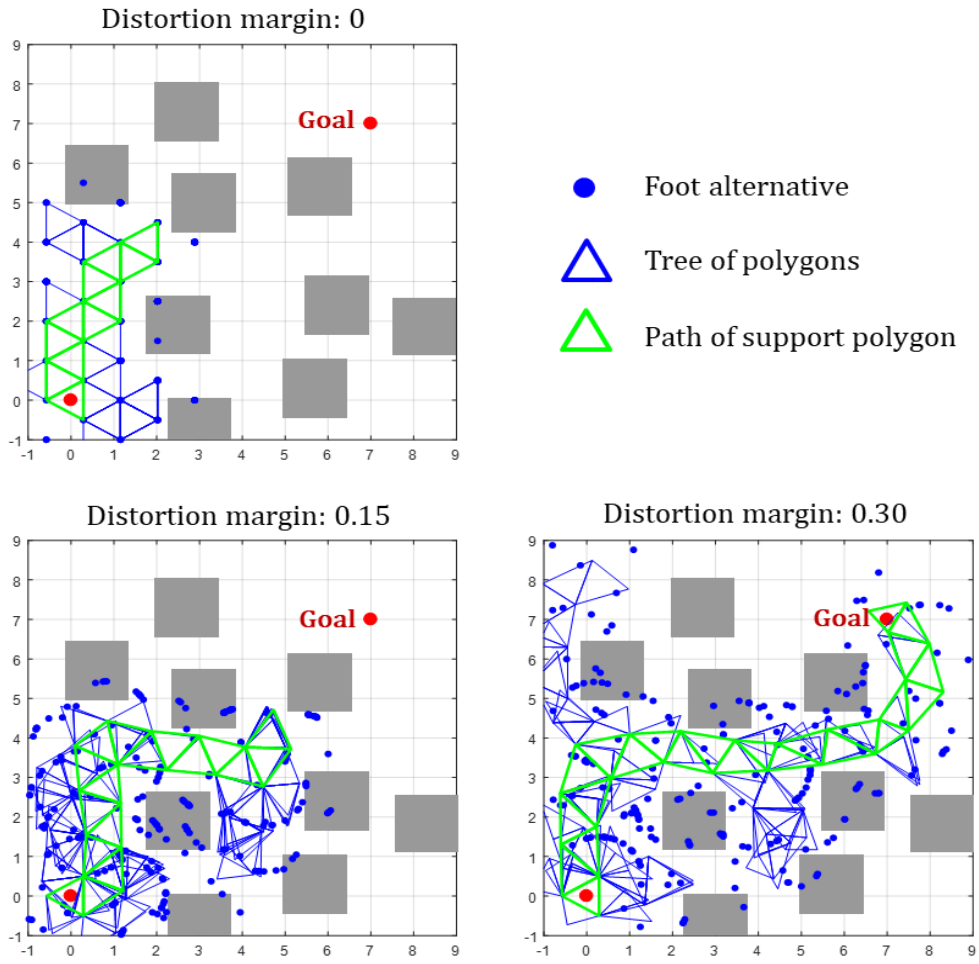


Fig. 4.2.3. Example of support polygon planning

As in the figure, the larger distortion margin it has, the more probably it reaches the goal. This is because high distortion margin provides more flexibility of polygon shape to pass through narrow space. However, large distortion margin deforms support polygon too much, which makes VTT difficult to follow support polygon satisfying constraints. Thus, distortion margin should be selected properly to succeed both in high-level and low-level planning.

4.2.2 Probabilistic Completeness of PRT algorithm

RRT has probabilistic completeness property that that makes it find a path to any goal position in operation space [34][35]. Probabilistic completeness is the property that, probability of the planner finding a path to goal position, if one exists, converges to 1 as the number of iterations increases [36]. This property is important in that it guarantees the planner finds existing path to the goal position.

To check if PRT has probabilistic completeness property, simulations was performed to observe how PRT tree grows in operational space. In the simulations, the nominal length L_{nom} and operational space X was set as follows.

$$L_{nom} = 1 \tag{4.6}$$

$$X = \{(x, y) \in R^2 \,|\, -10 \le x \le 10, -10 \le y \le 10\} \tag{4.7}$$

In addition, nine cases were set with different initial position of polygon, obstacles and distortion margin as in Table 4.1. For the obstacles, 15 squares with 1.5 size was randomly selected. The simulation was performed 5 times for each case.

Table 4.1. Condition of cases for probabilistic completeness simulation

	Initial position	**Existence of obstacles**	**Distortion margin**
Case 1	(0,0)	X	0.1
Case 2	(0,0)	X	0.2
Case 3	(0,0)	X	0.4
Case 4	(-9,-9)	X	0.1
Case 5	(-9,-9)	X	0.2
Case 6	(-9,-9)	X	0.4
Case 7	(0,0)	O	0.1
Case 8	(0,0)	O	0.2
Case 9	(0,0)	O	0.4

Fig. 4.2.4 – 4.2.6. shows PRT trees in different cases and iteration. This visually shows that, regardless of conditions, polygon of PRT tree covers almost all operational space as iteration increase. In addition, as RRT did, PRT tree tends to grow toward unexplored space. This is because Voronoi region [37] of foot alternatives in sparse space is larger than that of in dense space, and more likely to be chosen as nearest point by randomly picked objective point. Therefore, PRT can efficiently search unexplored space.

To calculate the probability of finding a path to the goal position, 1000 points was randomly sampled as goal positions and checked if the goal position is inside any polygon generated by PRT. Then, probability was calculated by dividing the number of sampled goal positions inside polygon by number of samples.

The average probability of finding a path for each PRT iterations is presented in Fig. 4.2.7. The result shows that the probability asymptotically approaches 1 as the iteration increases, regardless of initial position, distortion margin and obstacles. Although, the rate of convergence was higher for the cases with larger distortion margin, the difference was not significant. Although mathematical proof still remains, the simulation result shows that PRT probably has probabilistically completeness property. This property ensures that PRT find support polygon path, if one exists, regardless of initial position and obstacles.

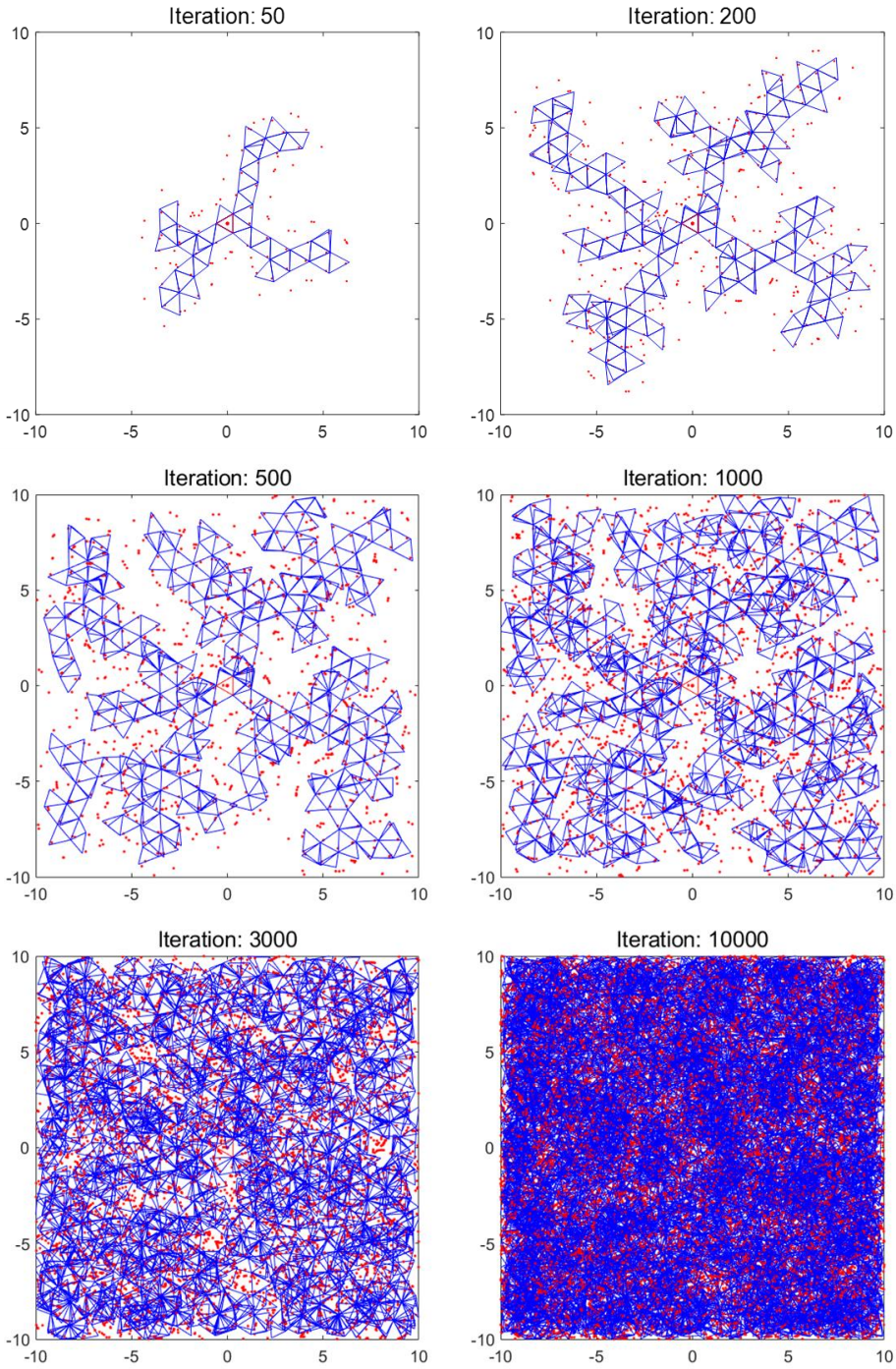


Fig. 4.2.4. PRT trees of (0,0) initial position and 0.2 distortion margin

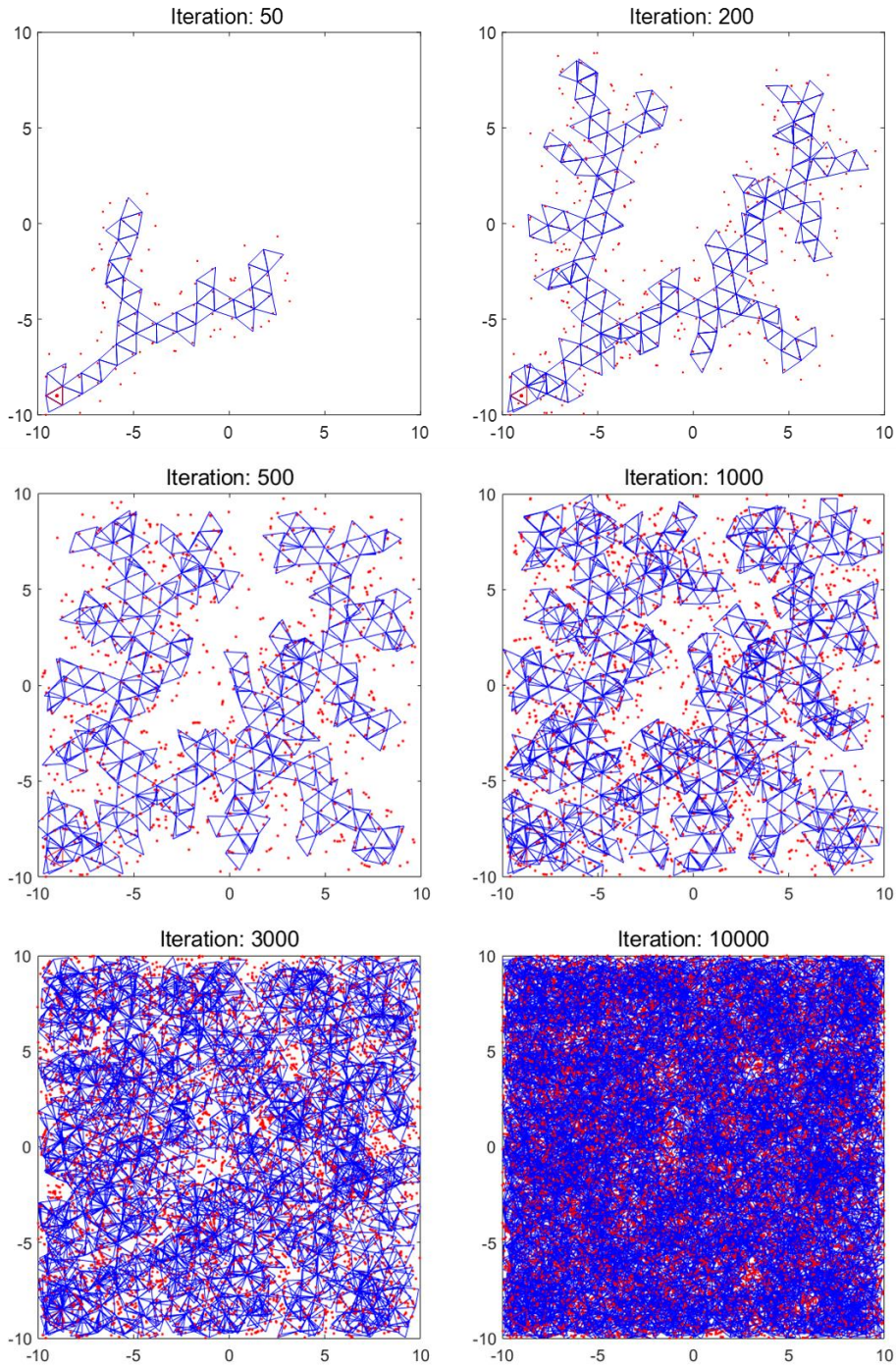


Fig. 4.2.5. PRT trees of (-9,-9) initial position and 0.2 distortion margin

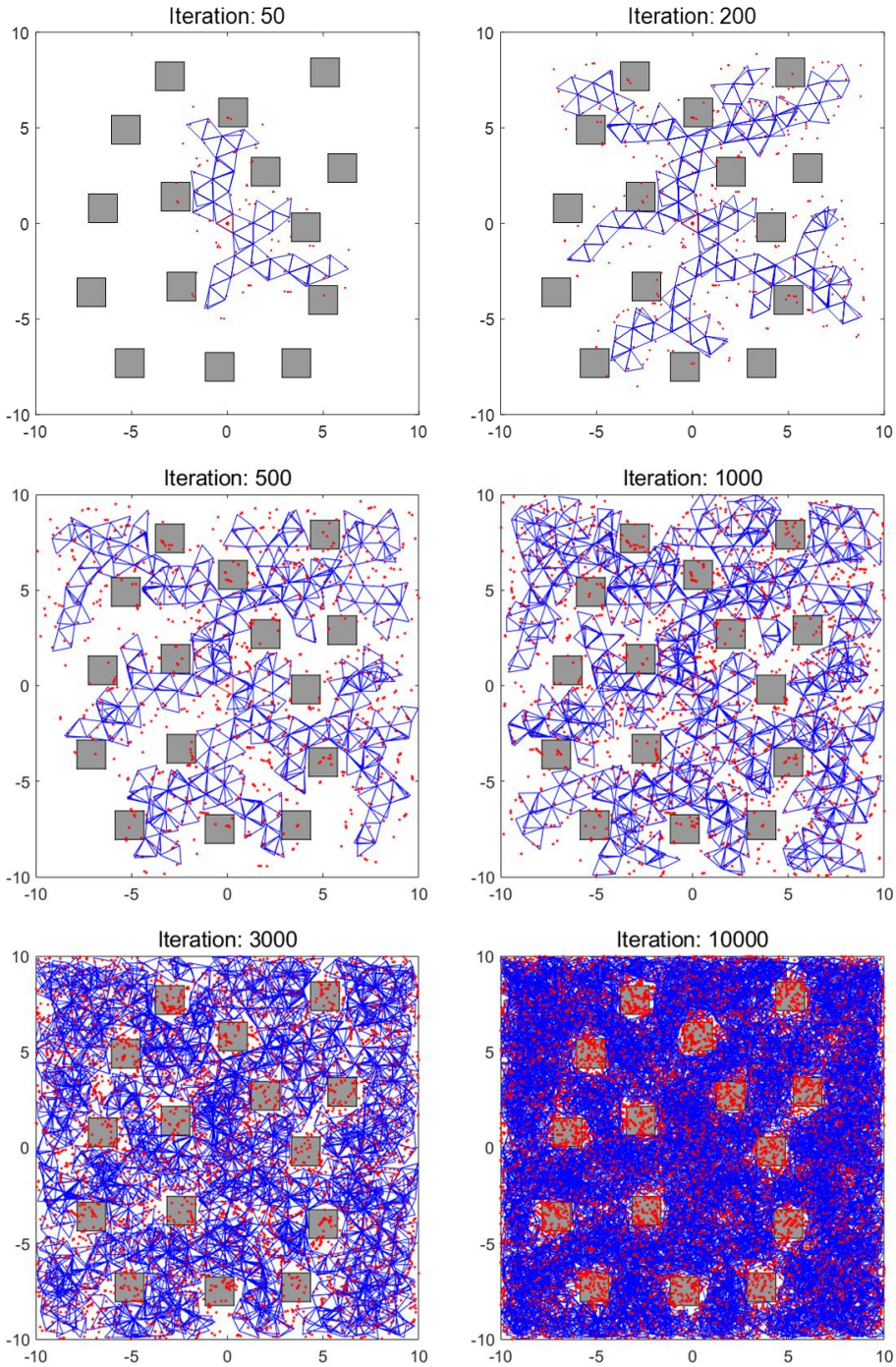
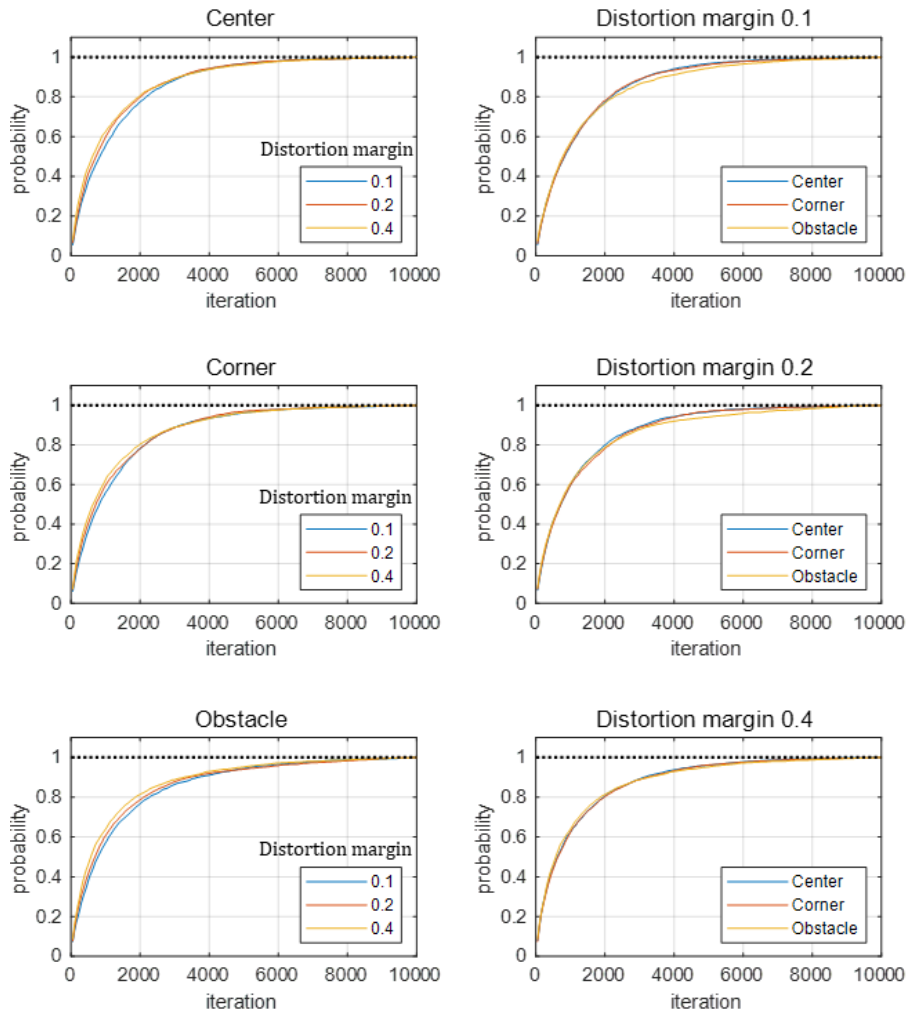


Fig. 4.2.6. PRT trees of (0,0) initial position and 0.2 distortion margin with 15 obstacles



(a) (b)

Fig. 4.2.7. Probability graph of PRT for each case.
(a) Probability graph for different distortion margin.
(b) Probability graph for different condition:
'Center' denotes cases with (0,0) initial position,
'Corner' denotes cases with (-9,-9) initial position
, and 'Obstacle' denotes cases with (0,0) initial position with obstacles

4.3 Center of Mass Planning

Objectives of center of mass planning is generating trajectory of center of mass projected on the ground for VTT to follow. Fig. 4.3.1 shows planned center of mass trajectory. The path of center of mass should be as stable as possible to make VTT easily follow and maintain good-conditioned shape. To maximize stability margin, the path generated by connecting centroid of polygon and midpoint of polygon edges. The speed of center of mass was set to be constant.

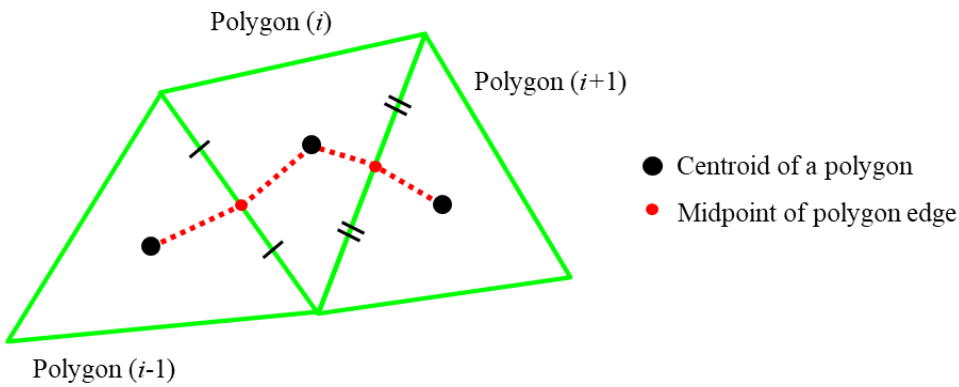


Fig. 4.3.1. Schematic of center of mass trajectory

4.4 Node Position Planning

Objectives of node position planning are as follows.

Generate trajectory of nodes that,

1) follows support polygon path and center of mass trajectory
2) maintain stability to prevent tumbling and damage from the ground

To achieves objectives, Non-Impact Rolling algorithm was proposed for node position planning.

4.4.1 Concept of Non-Impact Rolling Locomotion

Non-Impact Rolling locomotion algorithm plans trajectory of nodes that follows support polygon path and center of mass trajectory while maintaining stability of VTT. As explained in Section 3.3, VTT is stable when the projected center of mass is inside allowable region which generated by offsetting support polygon with minimum stability margin as in Fig. 4.4.1

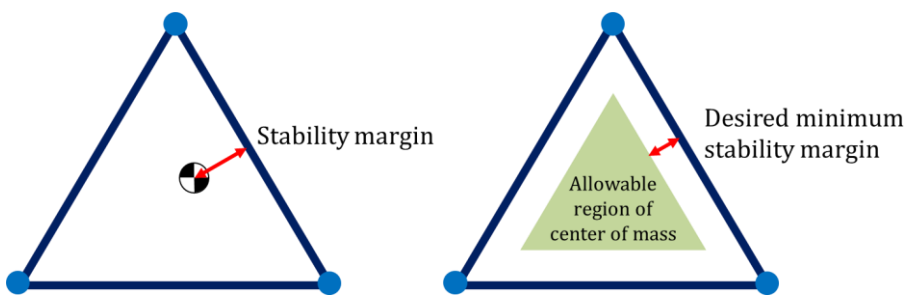


Fig. 4.4.1. Support polygon and stability margin

Considering the stability criteria, the motion of Non-Impact Rolling is divided into three phases: moving phase, landing phase, and transient phase. Non-Impact Rolling locomotion process from one support polygon to another is depicted in Fig. 4.4.2. The figures are arranged in chronological order from top to bottom. In the figure, green triangles denote desired support polygon planned in Step 1 and red dotted lines denote desired center of mass trajectory planned in Step 2.

The locomotion process begins with moving phase. In moving phase, the center of mass is in allowable region and VTT moves to the follow planned center of mass trajectory while maintaining the support polygon.

When the center of mass reaches the border of allowable region, the motion shifts to landing phase. In landing phase, VTT lands the frontal node on the vertex or foot of next desired support polygon while fixing the center of mass.

When the frontal node lands on the ground, the motion shifts to transient phase. As the frontal node lands, the frontal node and previous support polygon forms connected support polygon. Then, the allowable region of connected support polygon covers that region that was originally not allowable in previous support polygon. In transient phase, VTT moves to follow the planned center of mass trajectory in originally non-allowable region while maintaining the connected support polygon. After the center of reaches the border of allowable region in next support polygon, the motion shifts to moving phase again.

By repeating the process in Fig. 4.4.2 for every desired support polygon in the path, VTT can reach the goal position in the end.

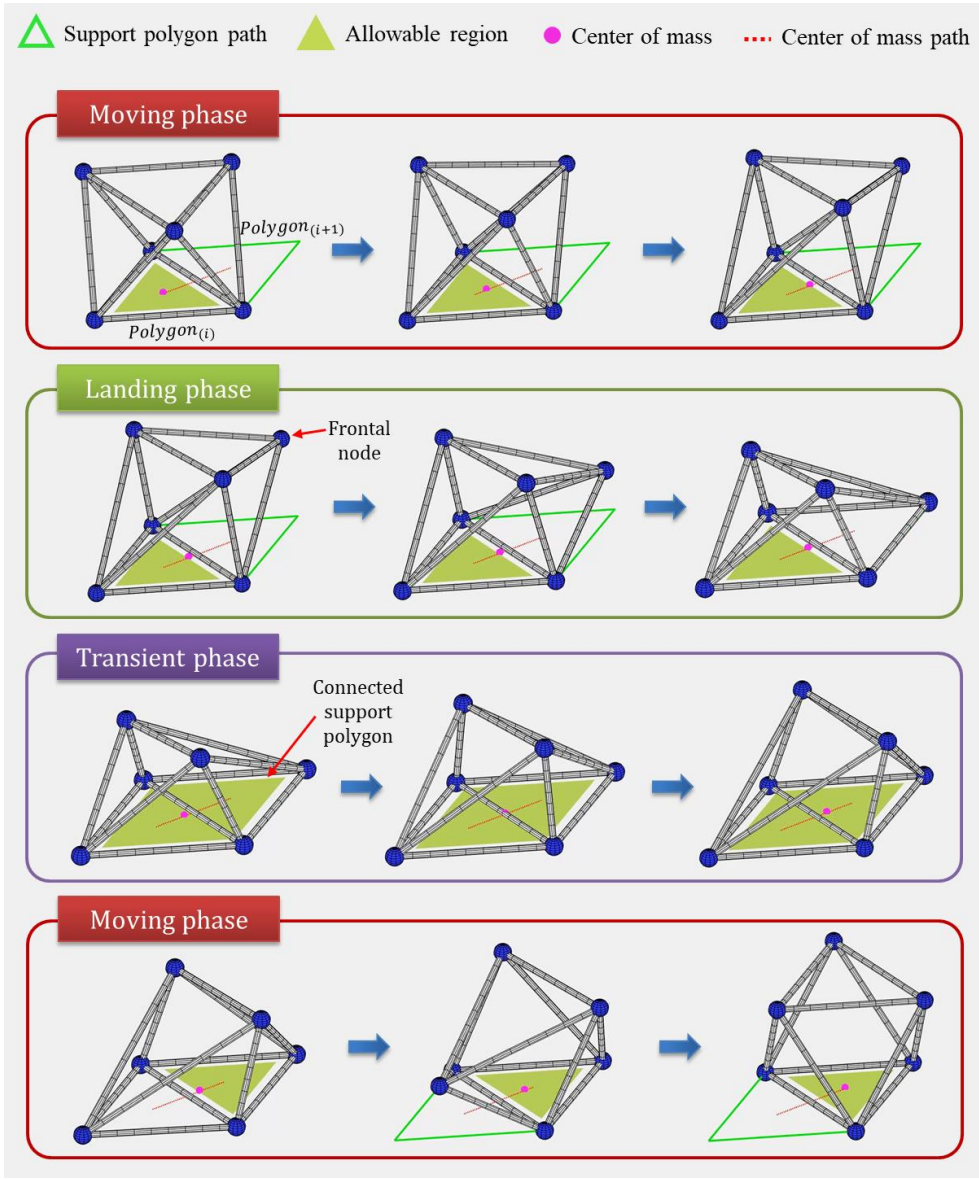


Fig. 4.4.2. Non-Impact Rolling locomotion process between two support polygons

4.4.2 Planning Algorithm for Non-Impact Rolling Locomotion

Based on the concept of Non-Impact Rolling locomotion, node position planning algorithm will be proposed in this section. Basically, proposed algorithm optimizes the velocity of each node at every time step. Finding the optimal velocity is easier and efficient than finding optimal position directly, because velocity relation can be represented in linear equation in inverse kinematics as follows.

$$\dot{L} = R\dot{x} \tag{4.8}$$

where $\dot{L}$ is velocity vector of all edges, $\dot{x}$ is velocity vector of all nodes, and R is Jacobian matrix between velocity of edges and velocity of nodes. After optimal velocity of nodes is found, optimal position of nodes be calculated by simple integration.

The overall algorithm of node position planning between two support polygons is provided in Algorithm 2.

Algorithm 2: Locomotion between two support polygons

Input: $Polygon_{(i)}, Polygon_{(i+1)}, \dot{x}_{CM,desire}, x_{init}$

$x_{CM} = Center\_of\_Mass(x_{init})$
$Sup\_Polygon = Polygon_{(i)}$
while $Reach\_Goal == false$ **do**
  if $Stb\_Margin(Sup\_Polygon, x_{CM}) > Min\_Stb\_Margin$
    // Moving phase
    $Phase = 'Moving'$
  else if $Landing\_Finish == false$
    // Landing phase
    $Phase = 'Landing'$
    if $Foot\_Landing(x_{new}) == true$ **then**
      // Change support polygon to the next one
      $Sup\_Polygon = Polygon_{(i+1)}$
      $Co\_Sup\_Polygon = ConvexHull(Polygon_{(i)}, Polygon_{(i+1)})$
      $Landing\_Finish = true$
    end if
  else if $Landing\_Finish == true$
    // Transient phase
    $Phase = 'Transient'$
  end if
  $x_{new} = NodeOptimization(x, \dot{x}_{CM,desire}, Phase)$
  $x_{CM} = Center\_of\_Mass(x_{new})$
  if $x_{CM} == Center\_of\_Mass(Polygon_{(i+1)})$ **then**
    $Reach\_Goal = true$
  end if
end while

Meaning of each variable in Algorithm 2 is as follows.

Table 4.2. Meaning of variables in Algorithm 2

x_{init}	Initial node position vector
x	Node position vector at current time step
x_{new}	Node position vector at next time step
x_{CM}	VTT's center of mass projected on the ground
$\dot{x}_{CM,desired}$	Desired velocity of the center of mass at current time step Derived from planned trajectory of the center of mass in Step 2
$Polygon_{(i)}$	Initial support polygon
$Polygon_{(i+1)}$	Desired next support polygon
$Sup\_Polygon$	Desired current support polygon
$Co\_Sup\_Polygon$	Connected support polygon composed of $Polygon_{(i)}$ and $Polygon_{(i+1)}$
$Stb\_M\arg in$	Current stability margin
$Min\_Stb\_M\arg in$	Desired minimum stability margin

Depends on the type of phase, position of node at next time step is optimized by function *NodeOptimization* in line 20 of Algorithm 2. The algorithm of *NodeOptimization* is presented in Algorithm 3.

Algorithm 3: Node position optimization

1 **function** $NodeOptimization(x, \dot{x}_{CM,desire}, Phase)$
2 $D = [\,]$
3 $Feasible = false$
4 **while** $Feasible == false$ **do**
5 **switch** $Phase$
6 **case** $'Moving'$
7 $\dot{x}_{temp} = MovingOptimization(x, \dot{x}_{CM,desire}, D)$
8 **break**
9 **case** $'Landing'$
10 $\dot{x}_{temp} = LandingOptimization(x, \dot{x}_{CM,desire}, D)$
11 **break**
12 **case** $'Transient'$
13 $\dot{x}_{temp} = TransientOptimization(x, \dot{x}_{CM,desire}, D)$
14 **break**
15 **end switch**
16 $x_{new} = x + \dot{x}_{temp}\Delta t$
17 $Active = 0$
18 **for** $i = 1$ **to** $N_{constraints}$ **do**
19 **if** $f_i(x_{new}) < 0$ **do**
20 $D = [D; \frac{\partial f_i(x)}{\partial x}]$
21 $Active = Active + 1$
22 **end if**
23 **end for**
24 **if** $Active == 0$ then
25 $Feasible = true$
26 **end if**
27 **end while**
28 **return** x_{new}
29 **end function**

Meaning of the variables in Algorithm 3 are as follows.

Table 4.3. Meaning of variables in Algorithm 3

MovingOptimization	Function to optimize velocity of nodes in moving phase
LandingOptimization	Function to optimize velocity of nodes in landing phase
TransientOptimization	Function to optimize velocity of nodes in transient phase
Δt	Time step size
Active	The number of active constraints
$N_{constraints}$	The number of all constraint equations
f_i	i-th constraint equation
D	Active constraint Jacobian

In line 5 to 15, it solves optimization problem depends on the type of phase. *MovingOptimization*, *LandingOptimization*, *TransientOptimization* is function to optimize velocity of nodes in moving phase, landing phase and transient phase respectively. Optimization method of this functions will be explained in next sections.

In line 16, it calculates temporary optimal node position in next time step, x_{new}. Here, Δt is time step size.

In line 17 to 26, it checks if x_{new} satisfies the constraints, and it generates constraint Jacobian D. This constraint handling strategy was suggested by Usevitch et al. [23]. The constraint equation can be generally written as follows,

$$f(x) = [\, f_1(x) \;\; f_2(x) \;\; \cdots \;\; f_n(x)]^T \leq 0 \tag{4.9}$$

where, $f_i(x)$ is a function that represent a constraint and $f(x)$ is a vector function that represent all constraints. Then, the equation can be linearized by taking time derivatives as follows.

$$\frac{df(x)}{dt} = \frac{\partial f(x)}{\partial x}\dot{x} \le 0 \tag{4.10}$$

Since it is differential form, above equation isn't applied all the time, but, it is applied only when configuration is on boundary of certain constraint. In line 18 to 23, it checks if the constraints are satisfied one by one, and if violated, the constraint become active and generate active constraint Jacobian D by adding a row, $\frac{\partial f_i(x)}{\partial x}$.

4.4.3 Optimization Problem of Moving Phase

In moving phase, VTT moves to follow planned center of mass trajectory while maintaining the support polygon as in Fig. 4.4.3.

For this motion, optimization problem of *MovingOptimization* is set as follows.

$$\begin{aligned} \min_{\dot{x}} \quad & \left\| \dot{x} - \dot{x}_{desire} \right\| \\ \text{Subject to} \quad & C\dot{x} = 0\,, \qquad \left| R\dot{x} \right| \le \dot{L}_{\max}\,, \qquad D\dot{x} \le 0\,, \\ & Center\_of\_Mass(\dot{x}) = \dot{x}_{CM,desired} \end{aligned} \tag{4.11}$$

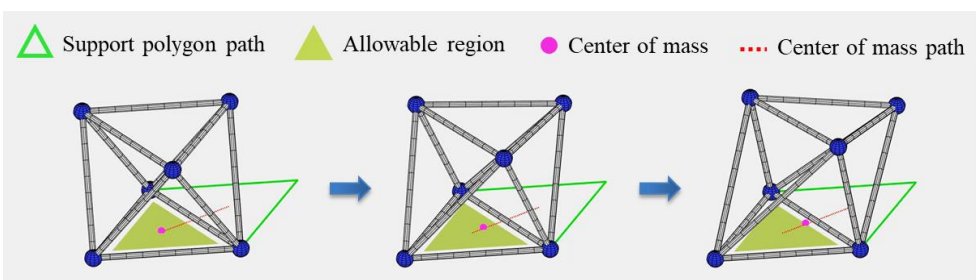
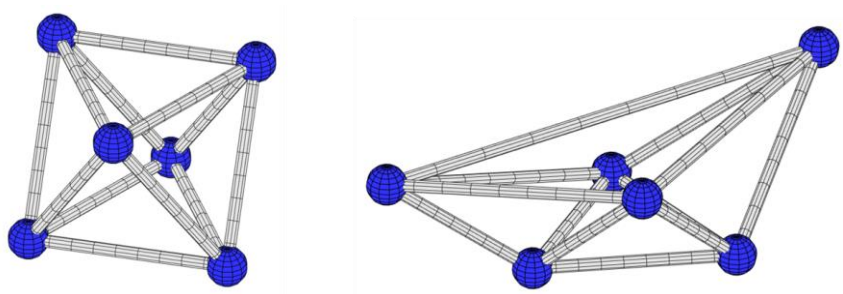


Fig. 4.4.3. Schematic of moving phase

The optimization variable is node velocity vector $\dot{x}$. Objective function is set to make $\dot{x}$ as near as possible to desired node velocity vector $\dot{x}_{desire}$. $C\dot{x} = 0$ is fixed node constraint and $\left|R\dot{x}\right| \le \dot{L}_{\max}$ is maximum member velocity constraint. The other constraints are applied by $D\dot{x} \le 0$.

In this algorithm, $\dot{x}_{desire}$ was defined as the node velocity that make VTT's configuration as close as possible to nominal configuration. The reason of pursuing nominal configuration is illustrated in. Fig. 4.4.4. Fig. 4.4.4.(a) shows the nominal configuration and Fig. 4.4.4.(b) shows ill-conditioned configuration with same topology, octahedron with 6 nodes and 12 members. Nominal configuration, having edges with nominal length, is most good-conditioned and far from the constraints. On the other hand, Ill-conditioned configuration which edge length is far from nominal length, is unstable and easily violate constraints.



(a) (b)

Fig. 4.4.4. Configuration of octahedron topology with 6 nodes and 12 edges.
(a) Nominal configuration (desirable). (b) Ill-conditioned configuration

From this idea, desired velocity of node i, $\dot{x}_{i,desire}$ can be represented as follows [38],

$$\dot{x}_{i,desire} = \sum_{j \in N_i} (\left\| x_j - x_i \right\| - L_{nom}) \frac{(x_j - x_i)}{\left\| x_j - x_i \right\|} \tag{4.12}$$

where, x_i is position vector of node i and N_i is set of nodes adjacent to node i. Then, $\left\| x_j - x_i \right\| - L_{nom}$ is length difference between actual length of edge $\{i, j\}$ and nominal length. $\frac{(x_j - x_i)}{\left\| x_j - x_i \right\|}$ is unit direction vector of edge $\{i, j\}$. Schematic of $\dot{x}_{i,desire}$ is depicted in Fig. 4.4.5. In Fig. 4.4.5.(a), red arrow denotes vectors that proportional to difference between nominal length and actual length of edges, $\{i, j_1\}, \{i, j_2\}$ and $\{i, j_3\}$. In Fig. 4.4.5.(b), thick red arrow denotes $\dot{x}_{i,desire}$ which is sum of the vectors. Consequently, it can be seen that length of edges become similar to nominal length after node i moves with $\dot{x}_{i,desire}$.

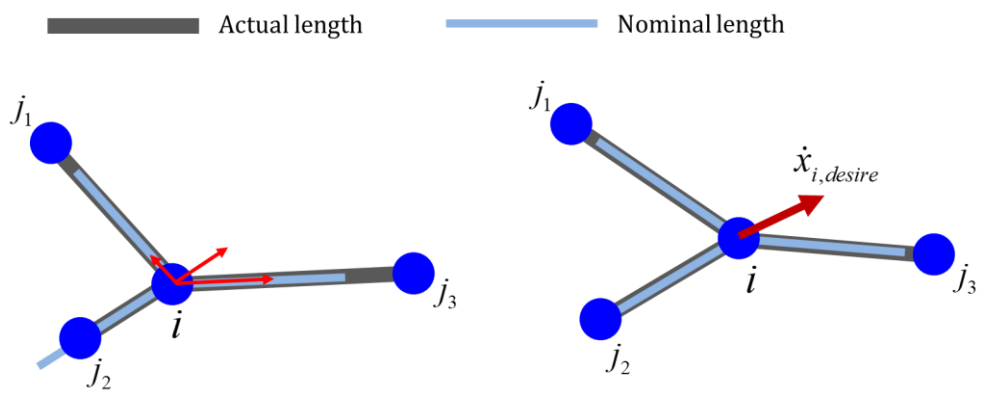


Fig. 4.4.5. Schematic of desired velocity of a node.
(a) Initial configuration, (b) Configuration after moves with desired velocity

By generalizing the equation, desired velocity of all nodes, $\dot{x}_{desire}$ can be written as follows,

$$\dot{x}_{desire} = -Lx + d \tag{4.13}$$

where, L is graph Laplacian [39] and d is vector which element is represented as follows.

$$d_i = \sum_{j \in N_i} L_{nom} \frac{(x_j - x_i)}{\left\| x_j - x_i \right\|} \tag{4.14}$$

To solve the optimization problem, interior-point method [40] was used as optimizer which is suitable for the problem with high dimensional variables. Having similar structure, optimization problem of moving phase and transient phase also uses interior-point method as optimizer.

4.4.4 Optimization Problem of Landing Phase

In landing phase, VTT lands the frontal node on the vertex or foot of next desired support polygon while fixing the center of mass as in Fig. 4.4.6. For this motion, optimization problem of $LandingOptimization$ is as follows.

$$\begin{aligned} \min_{\dot{x}} \quad & \left\| P_{front}(\dot{x}) - F_{next} \right\| \\ \text{Subject to} \quad & C\dot{x} = 0\,, \qquad \left| R\dot{x} \right| \le \dot{L}_{\max}\,, \qquad D\dot{x} \le 0\,, \\ & Center\_of\_Mass(\dot{x}) = 0 \end{aligned} \tag{4.15}$$

Here, $P_{front}(\dot{x})$ denotes the position vector of the frontal node and F_{next} is the position vector of objective foot for the next support polygon. The other constraints are same as $MovingOptimization$ except that the center of mass is set to be fixed.

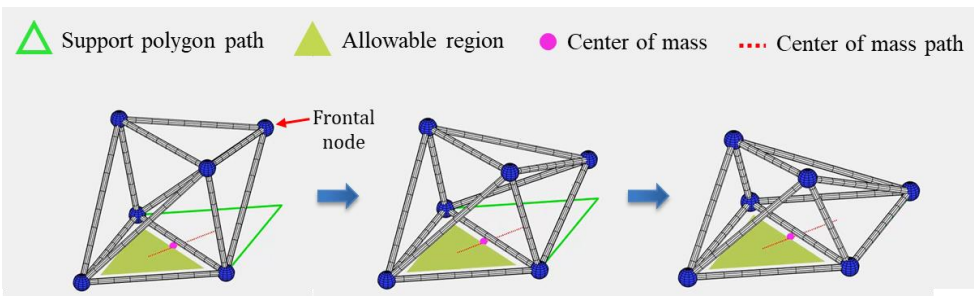


Fig. 4.4.6. Schematic of landing phase

4.4.5 Optimization Problem of Transient Phase

In transient phase, VTT moves to follow the planned center of mass trajectory in originally non-allowable region while maintaining the connected support polygon as in Fig. 4.4.7. The motion of transient phase is basically same as that of moving phase except the support polygon is connected support polygon. Thus, optimization problem of *TransientOptimization* was set as same as *MovingOptimization* as follows,

$$\min_{\dot{x}} \quad \left\| \dot{x} - \dot{x}_{desire} \right\|$$

$$\text{Subject to} \quad C\dot{x} = 0\,, \qquad \left| R\dot{x} \right| \le \dot{L}_{\max}\,, \qquad D\dot{x} \le 0\,,$$

$$Center\_of\_Mass(\dot{x}) = \dot{x}_{CM,desired} \tag{4.16}$$

where, fixed nodes are set to be connected support polygon.

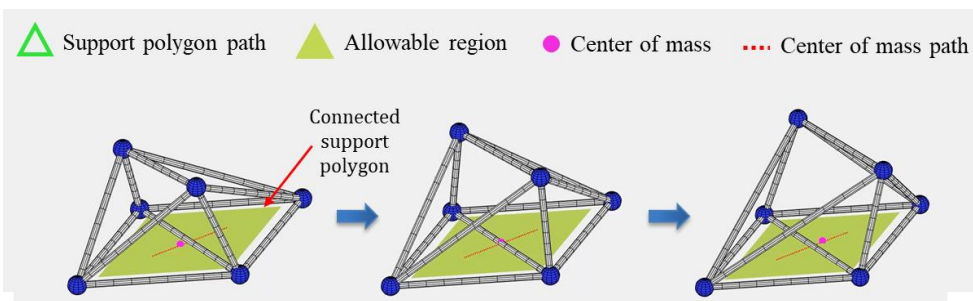


Fig. 4.4.7. Schematic of transient phase

Chapter 5

Experimental Verification

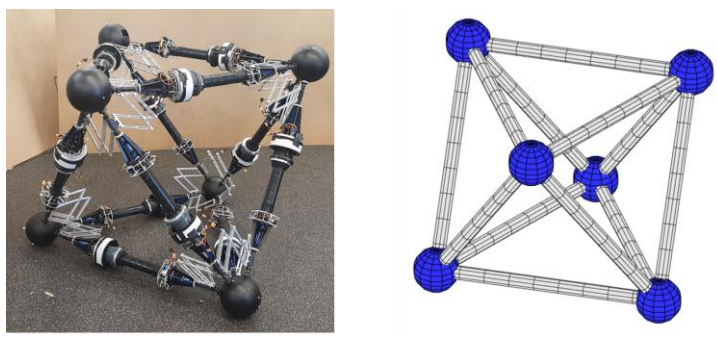
Proposed locomotion algorithm was verified by simulation experiment. In the simulation, trajectory of nodes position was planned for locomotion when initial configuration, goal position and obstacles were given. Here, all kinds of constraints explained in Chapter 3 was considered.

The locomotion planning was performed in two case studies. In case study 1, I will verify the algorithm is applicable to actual VTT. For the verification, the specification of actual VTT was applied as the constraints, so the planned trajectory can be used for actual VTT prototype. In case study 2, I will verify that if the constraints are relaxed, the algorithm is applicable to more complex environment with narrow passage. In the simulation, constraints were relaxed by giving wider range of member length, and higher member strength. Then, I will show VTT is possible to travel complex environment by changing its size more dramatically.

5.1 Case Study 1: Actual VTT Prototype

5.1.1 Simulation Condition

In this case study, actual constraint of VTT was applied for locomotion planning to verify the proposed algorithm is applicable to actual VTT prototype. Locomotion topology of VTT was set as octahedron with 12 edges and 6 nodes as Fig. 5.1.1. Mass and center of mass of an edge module is as Fig. 5.1.2. Constraint of VTT are listed in Table 5.1. Note that the ground contact function was set to be $f(x, y) = 0$ to represent flat ground.



(a) (b)

Fig. 5.1.1. Octahedron topology (12 edges, 6 nodes)
(a) VTT prototype, (b) Schematic of the topology

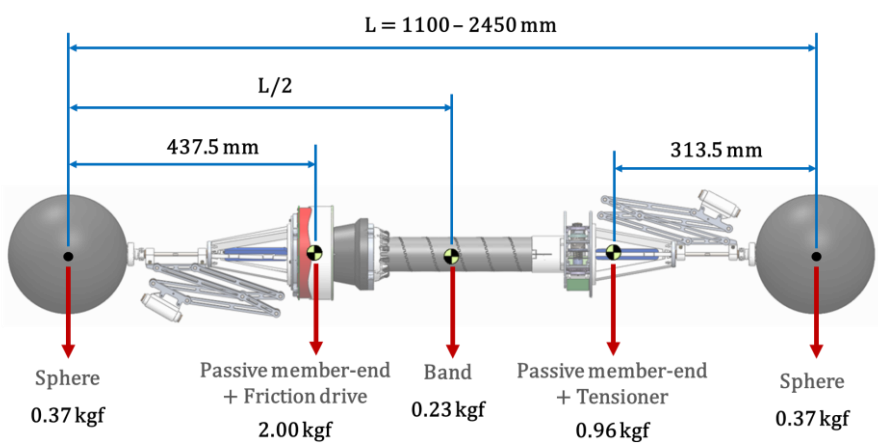


Fig. 5.1.2. Edge module's mass and center of mass

Table 5.1. Constraint parameter of VTT in Case study 1

Nominal length of edges	1.3 m
Length of edges	1.15 m – 2.4 m
Angle between adjacent edges	25° – 155°
Minimum dihedral angle	45°
Minimum distance between non-adjacent edges	0.16 m
Ground contact condition	$f(x, y) = 0$
Compressive strength	1000 N
Compressive actuation strength	80 N
Tensile strength	75 N
Tensile actuation strength	75 N
Maximum velocity of members	0.02 m/s
Stability margin	0.05 m

The locomotion test bench was set inside VICON motion capture system which covers 5 m × 20 m area. Start, goal position and obstacles were set to be in the VICON area as Fig. 5.1.3. Initial configuration was set to be regular octahedron which length of edge is nominal length. Initial support polygon is regular triangle and its center of mass is (0 , 0).

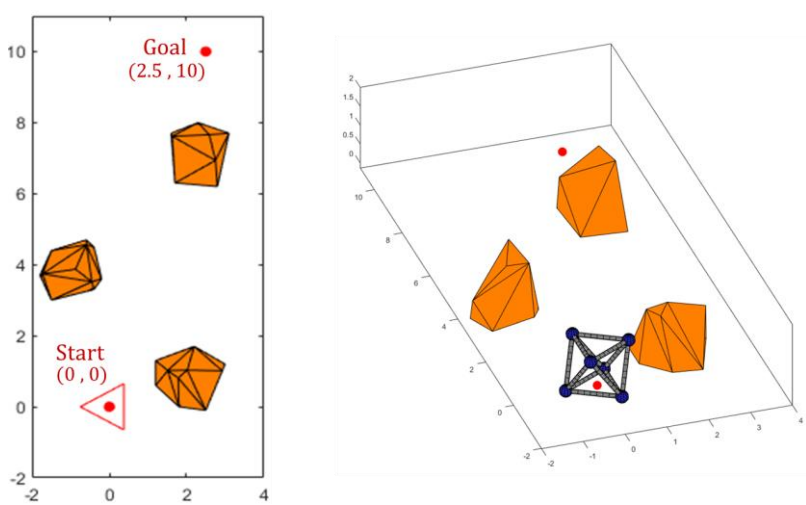


Fig. 5.1.3. Start position, goal position, obstacles and initial configuration

5.1.2 Obstacle Avoidance Method

In support polygon planning in Chapter 4, I assumed that the robot avoid obstacle when the edge of support polygon and edge of obstacles do not intersect. However, in the actual case, this does not guarantee obstacle avoidance since the robot has complex 3D shape. Especially, as actual VTT has tight constraint, it needs collision check method that keeps sufficient distance from obstacles.

For obstacle collision check of the VTT in support polygon planning, I defined safety space as in Fig. 5.1.4. Safety space is cylindrical space that enclose the good-conditioned octahedron configuration. Top and bottom face of the cylinder was set to be a circle with a center at the circumcenter of support polygon. Radius of the circle, safety radius was set as the sum of radius of Sphere and radius of circumcircle of support polygon.

Considering safety space in support polygon planning, I assumed that a polygon collides with obstacle when the minimum distance from circumcenter of the polygon to obstacles is less than safety radius. This collision check criteria gives sufficient space for VTT to avoid obstacle in node position planning.

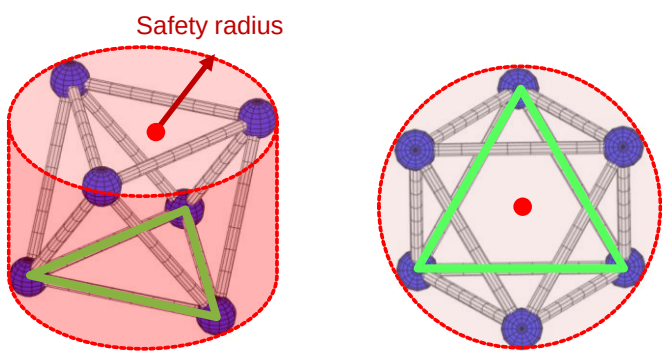


Fig. 5.1.4. Safety space of octahedron VTT: safety space is modeled as a cylinder

In node position planning, the obstacle collision was checked more precisely. The nodes of VTT were modeled as Spheres with a radius of 100 mm and the members was modeled as cylinders with a radius of 80 mm which is same as the actual radii. The obstacles were modeled as polygon mesh. During node position planning algorithm, the collision was checked by calculating minimum distance between components (nodes, members) of VTT and obstacles mesh using checkCollision function in MATLAB Robotics System Toolbox [41]. The minimum distance was represented as constraint equation and handled using constraint Jacobian D as explained in Section 4.3.

5.1.3 Simulation Result

Using the locomotion algorithm and the obstacle avoidance method, support polygon planning, center of mass planning and node position planning was performed sequentially. In Step 1, support polygon planning, the radius of distortion margin was set to be 0.1 m and mixing factor was set to be 0.2. Support polygon planning result are shown in Fig. 5.1.5. It can be seen that, as the PRT iteration increases, *Polygon* trees are expands to near goal position. And once a *Polygon* includes a goal position, path of support polygon generated by tracking by tracking the parent *Polygon* iteratively. Note that safety space denoted by red dotted circle does not collide with obstacles.

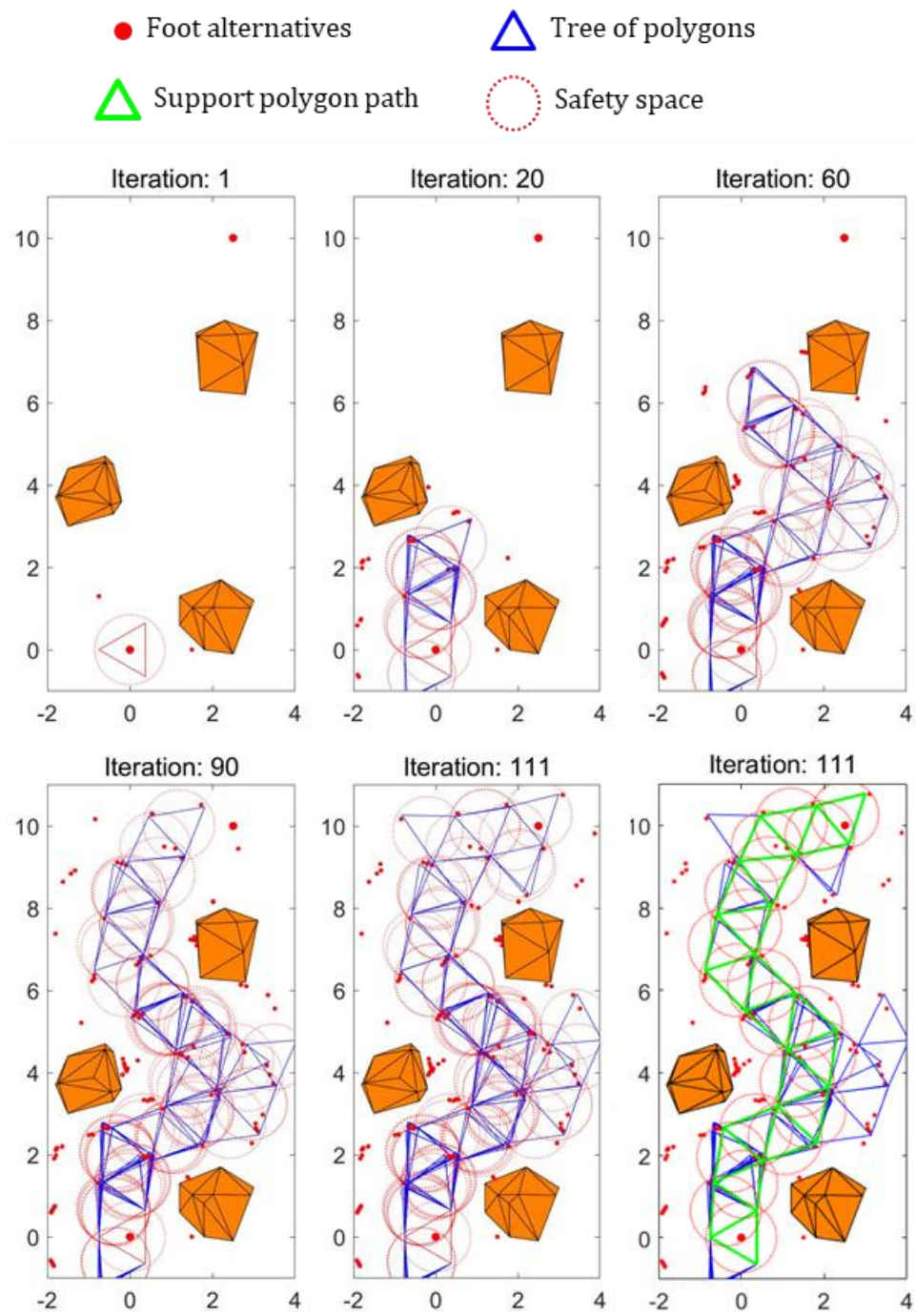


Fig. 5.1.5. Result of support polygon planning

In Step 2, center of mass planning, path of the desired center of mass was generated by connecting centroid of polygon and midpoint of each polygon edges as in Fig. 5.1.6. Speed of desired center of mass was set to be 6.67 mm/s which is one third of the maximum velocity of members. The desired center of ass was set to be stops at the centroid of support polygon for 10 seconds, in order to provide time to adjust VTT's configuration to nominal configuration.

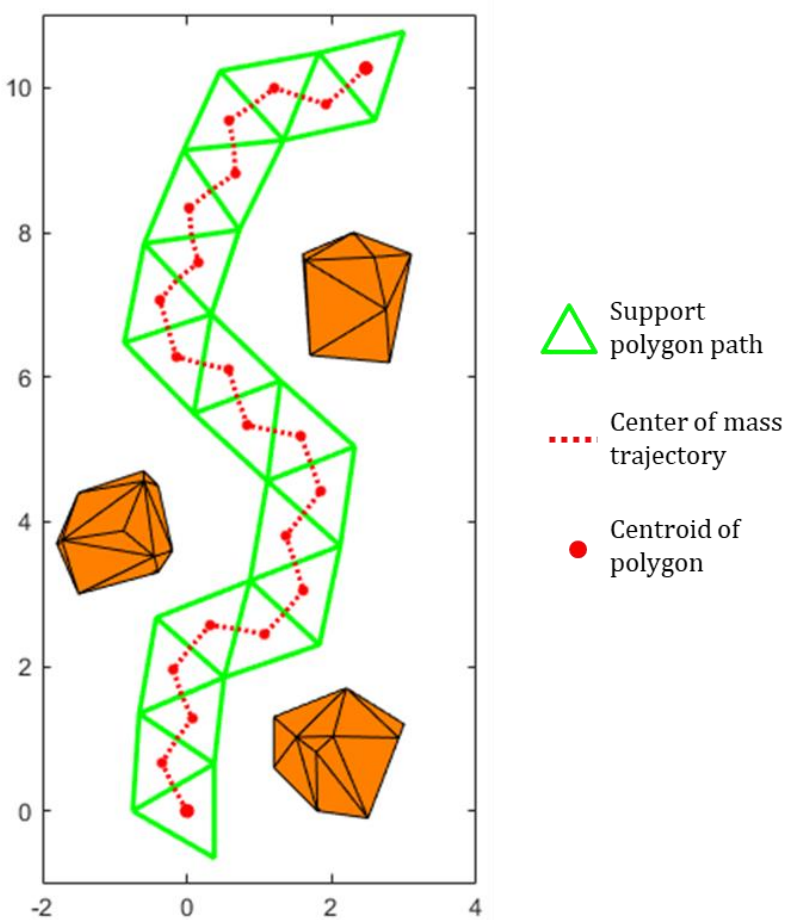


Fig. 5.1.6. Center of mass trajectory

In Step 3, node position planning, time step size Δt was set to be 1 second. The resultant motion of VTT is shown in Fig. 5.1.7 and Fig. 5.1.8. Constrains graphs are shown in Fig. 5.1.9. In constraint graph, the maximum and minimum values among all edges or combinations are presented. The black dotted line denotes the constraint boundary.

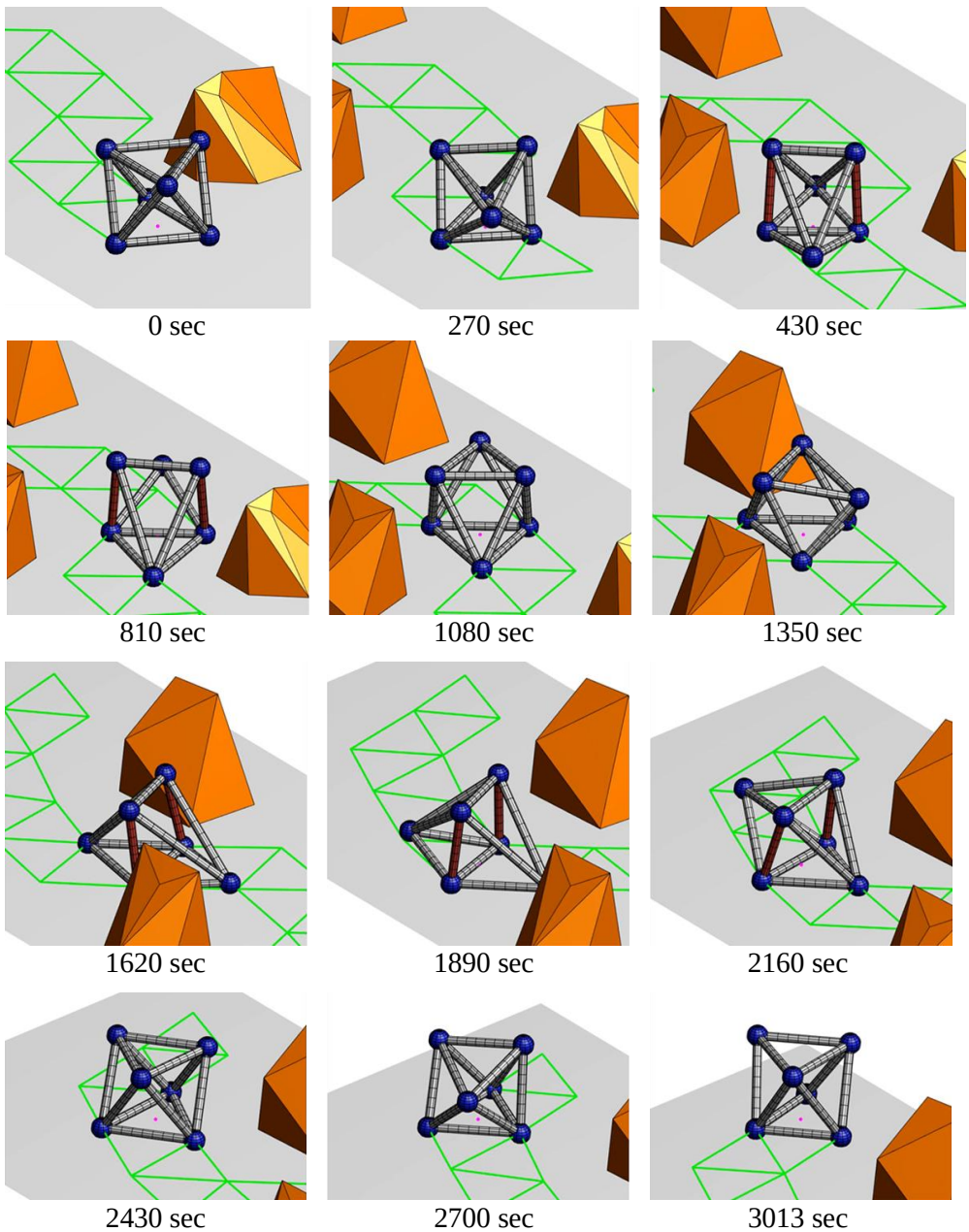


Fig. 5.1.7. Simulation of VTT locomotion (isometric view)

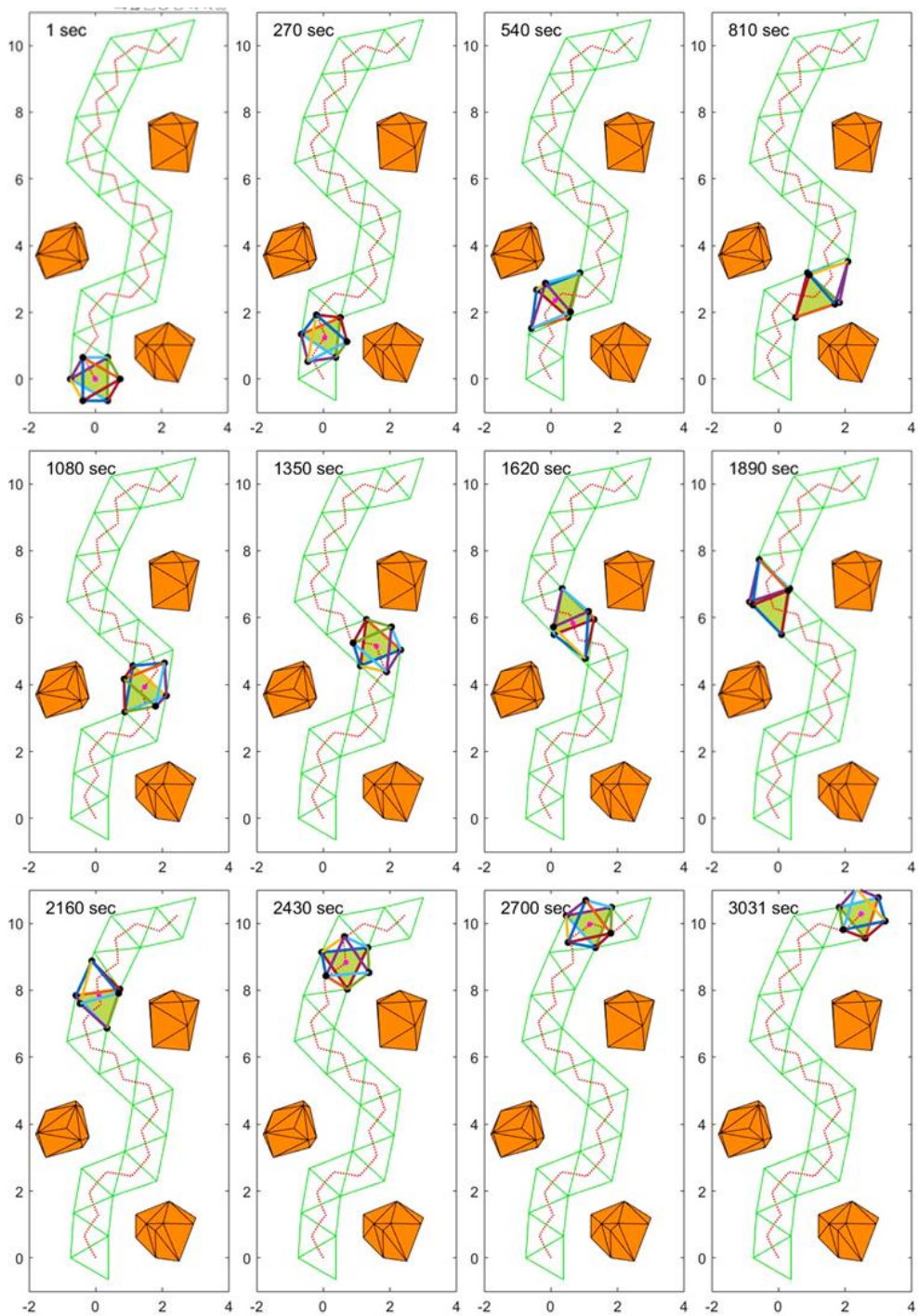


Fig. 5.1.8. Simulation of VTT locomotion (top view)

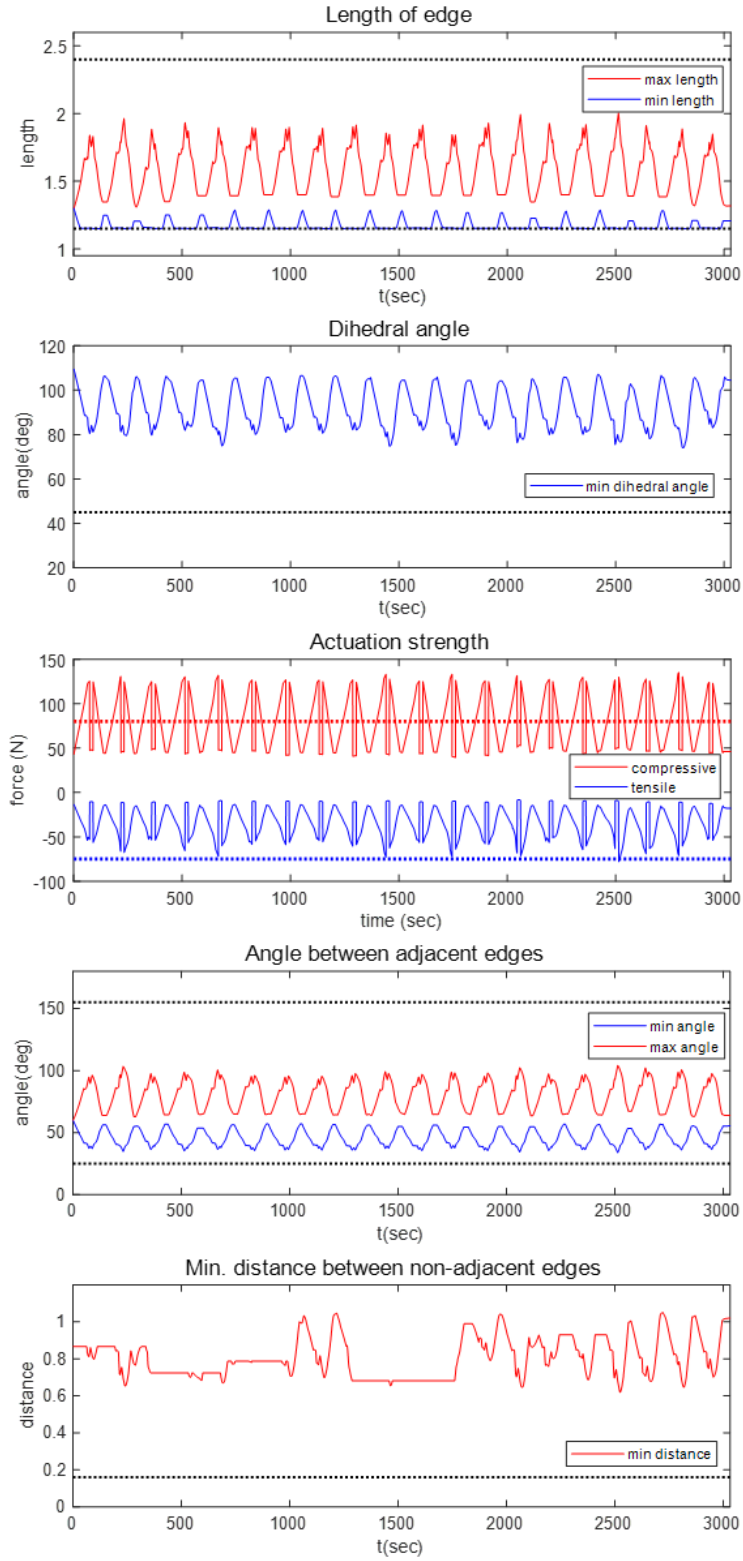


Fig. 5.1.9. VTT constraints graph

In actuation strength graph in Fig. 5.1.9, it can be seen that compressive force exceeds compressive actuation strength time to time. Although this excessive compressive force does not fail or damage the member, the member be actuated and changes the length. Thus, the member exceeding compressive actuation strength was set to be fixed during the planning. The fixed members are colored with red in Fig. 5.1.7.

In the simulation, VTT followed both support polygon path and center of mass trajectory while satisfying all constraints. By following near nominal support polygon and center of mass trajectory that centered in support polygon path, VTT could maintain good conditioned shape which makes the planning successful. The constraint graph in Fig. 5.1.9 shows that the constraint values fluctuate periodically within certain bound. These represent that VTT moves in moving phase, landing phase, and transient phase repetitively for Non-Impact Rolling locomotion, while not deviating from near good-conditioned shape. From this tendency, it can be expected that proposed algorithm can plan the motion of VTT that can travel even longer distance without planning failure.

5.2 Case Study 2:

Environment with Narrow Passage

5.2.1 Simulation Condition

In case study 1, I verified that proposed algorithm is applicable to actual VTT prototype. In this case study, I will verify that if VTT has relaxed constraints. The algorithm is also applicable to more extreme environment with narrow passage. Topology of VTT was set as octahedron with 12 edges and 6 nodes as Fig. 5.2.1. In this case study, constraint of VTT was arbitrarily set as Table 5.2, and mass and center of mass of an edge module was set as Fig. 5.2.2.

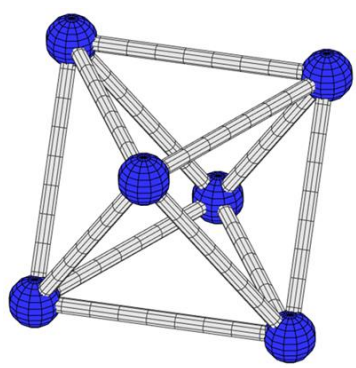


Fig. 5.2.1. Octahedron topology (12 edges, 6 nodes)

Table 5.2. Constraint parameter of VTT in Case study 2

Length of edges	0.3 m – 6 m
Angle between adjacent edges	15° – 180°
Minimum dihedral angle	35°
Minimum distance between non-adjacent edges	0.1 m
Ground contact condition	$f(x, y) = 0$
Compressive strength	1000 N
Compressive actuation strength	200 N
Tensile strength	200 N
Tensile actuation strength	200 N
Maximum velocity of members	0.04 m/s
Stability margin	0.02 m
Radius of nodes	0.3 m
Radius of members	0.1 m

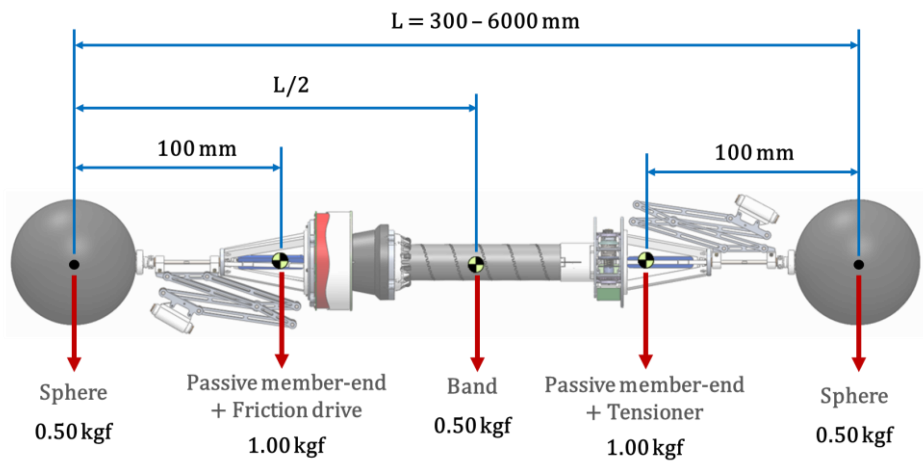


Fig. 5.2.2. Edge module's mass and center of mass

Start, goal position and obstacles were as Fig. 5.2.3. Initial configuration was set to be regular octahedron which length of edge is nominal length. Initial support polygon is regular triangle and its center of mass is (0 , 0).

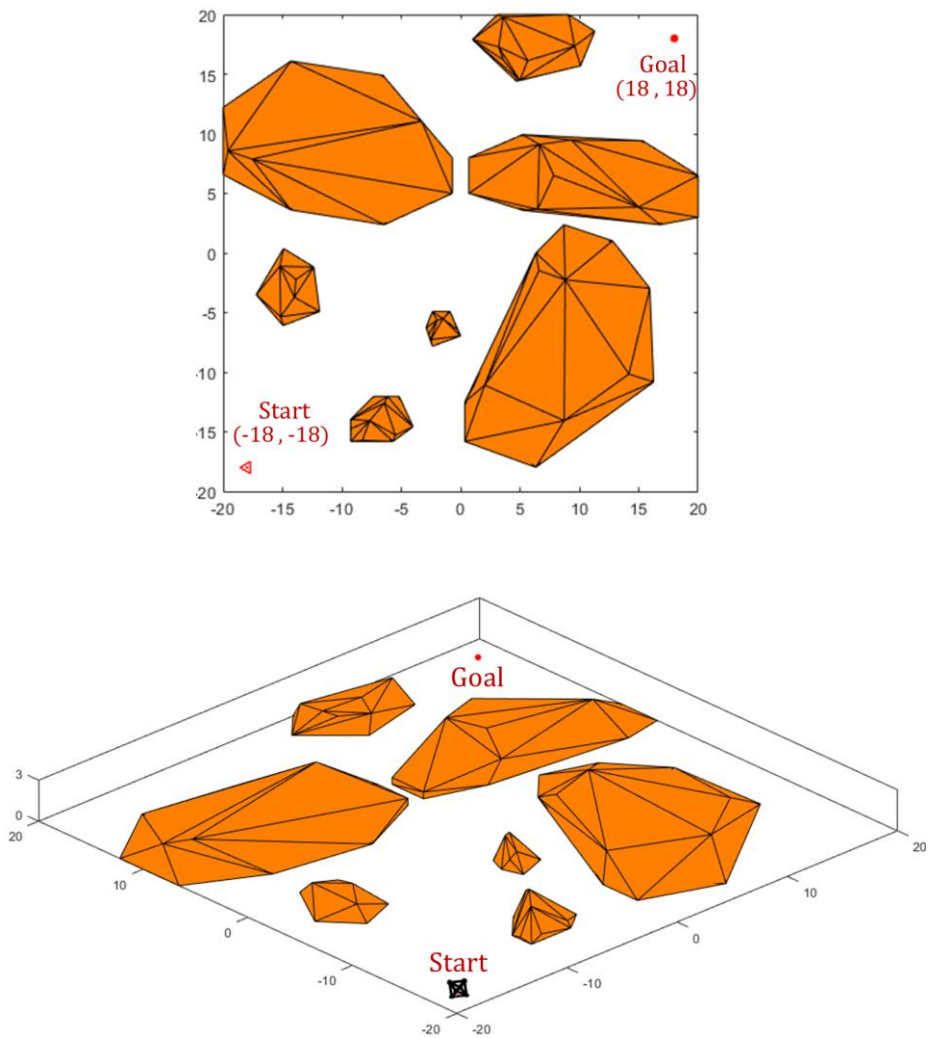


Fig. 5.2.3. Start position, goal position, obstacles and initial configuration

5.2.2 Support Polygon Planning with Varying Nominal Length

In the proposed algorithm, nominal length determines the overall shape and size of VTT during locomotion. The nominal length was fixed in Case study 1, since the given environment has simple and regular terrain. However, the environment in Case study 2 is irregular having both wide free space and narrow passage. In this case, VTT can moves more efficiently by varying the nominal length: moves rapidly by applying long nominal length, and go through narrow passage by applying short nominal length. In this section, I will suggest support polygon planning method for applying different nominal length depending on the terrain.

To divide the operational space and apply different motion strategy, waypoints was assigned and three zones were defined based on the waypoints as in Fig. 5.2.4. Zone 1 and 3 has wide free space, so the long nominal length will be applied to move rapidly. Whereas, Zone 2 has narrow passage, so the short nominal length will be applied to go through the narrow passage.

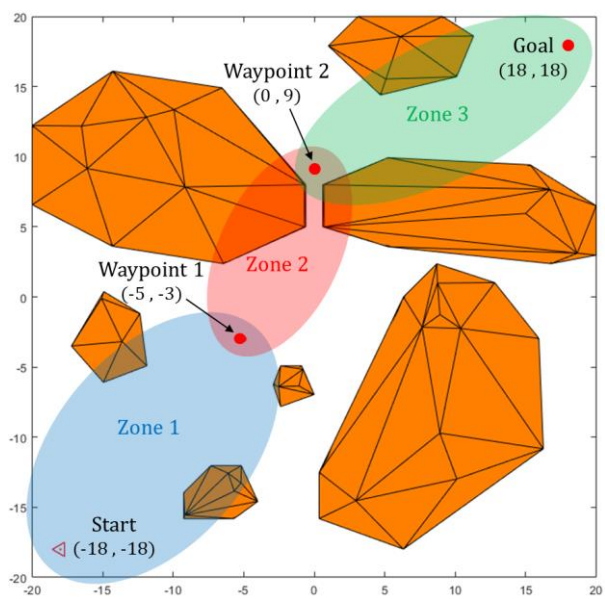


Fig. 5.2.4. Waypoints and zones in locomotion environment

Nominal length determines the size of support polygon, and drastic changes of nominal length generates too distorted shape of support polygon. Thus, rather than changing the nominal length directly to desired value, the nominal length should be changes gradually as the polygons expand by PRT. In this study, nominal length for each zone was set as follows.

$$L_{nom} = \begin{cases} L_{nom\_ini} \times \left(\dfrac{L_{nom\_des}}{L_{nom\_ini}} \right)^{\frac{G}{N_{step}}} , & G \le N_{step} \\ L_{nom\_des} , & G > N_{step} \end{cases} \quad (5.1)$$

Where, L_{nom_ini} is initial nominal length, L_{nom_des} is desired nominal length, N_{step} is the number of generation to reach L_{nom_des}, and G is generation. The generation of PRT polygon is depicted in Fig. 5.2.5. As in the figure, the generation of the initial polygon is 1, and the generation of expanded polygon became 1 larger than that of previous polygon.

As in equation (5.1), L_{nom} increases exponentially by generation, to keep constant ratio of L_{nom} of a polygon to that of the next generation polygon.

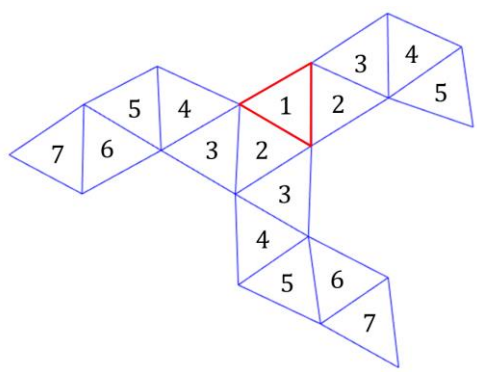


Fig. 5.2.5. PRT trees with generations: initial polygon is colored with red

L_{nom_ini}, L_{nom_des}, N_{step} was set for each zone as Table 5.3, radius of distortion margin was set to be $0.15 \times L_{nom}$.

Table 5.3. Nominal lengths for each zone

	L_{nom_ini}	L_{nom_des}	N_{step}
Zone 1	1	3	6
Zone 2	3	0.5	10
Zone 3	0.5	2	8

While, changing the nominal length as described, support polygon planning is performed as follows.

1) **Support polygon planning in Zone 1:**
   - Set goal position to waypoint 1.
   - Generate support polygon path between start position and waypoint 1.
2) **Support polygon planning in Zone 2:**
   - Set initial polygon to the final polygon in the path generated in 1)
   - Set goal position to waypoint 2
   - Generate support polygon path between the initial polygon and goal Position
3) **Support polygon planning in Zone 3**
   - Set initial polygon to the final polygon in the path generated in 2)
   - Generate support polygon path between the initial polygon and final goal position
4) **Generate the final support polygon path by connecting the path generated in 1) to 3)**

5.2.3 Simulation Result

The locomotion planning was performed by changing nominal length according to the type of terrain. For reference, Fig. 5.2.6. shows the waypoints, zones, and desired nominal length defined in this case study. Obstacle avoidance method described in Section 5.1.2 was also applied for the planning. In Step 1, support polygon planning, radius of distortion margin was set to be $0.15 \times L_{nom}$ and mixing factor was set to be 0.3. The result of support polygon planning is shown in Fig. 5.2.7. Fig. 5.2.7.(a),(b), and (c) is the planned support polygon path in Zone1, Zone2, and Zone 3 respectively, and Fig. 5.2.7. (d) is final support polygon from start position to goal position made by connecting the support polygon path in each zone. Note that the size of support polygon increases or decreases by generation to reach desired nominal length in each zone.

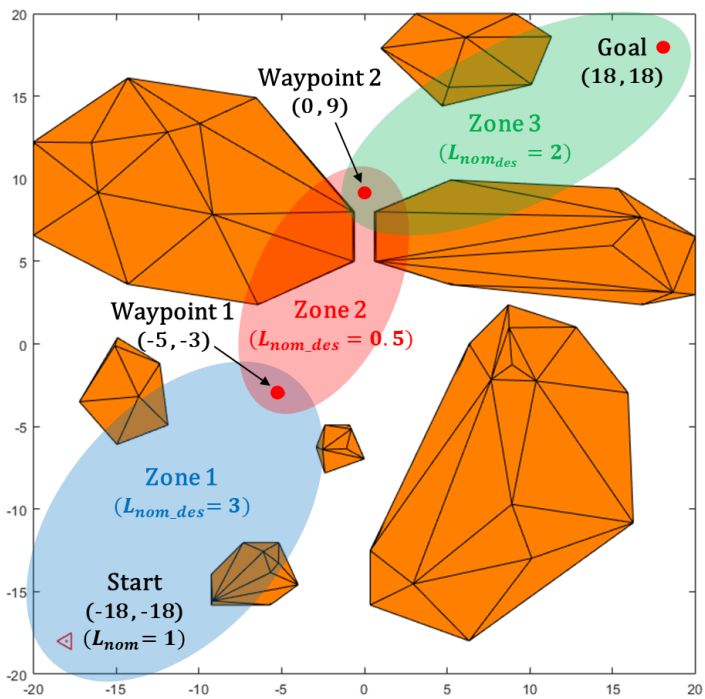


Fig. 5.2.6. Waypoints and zones in locomotion environment

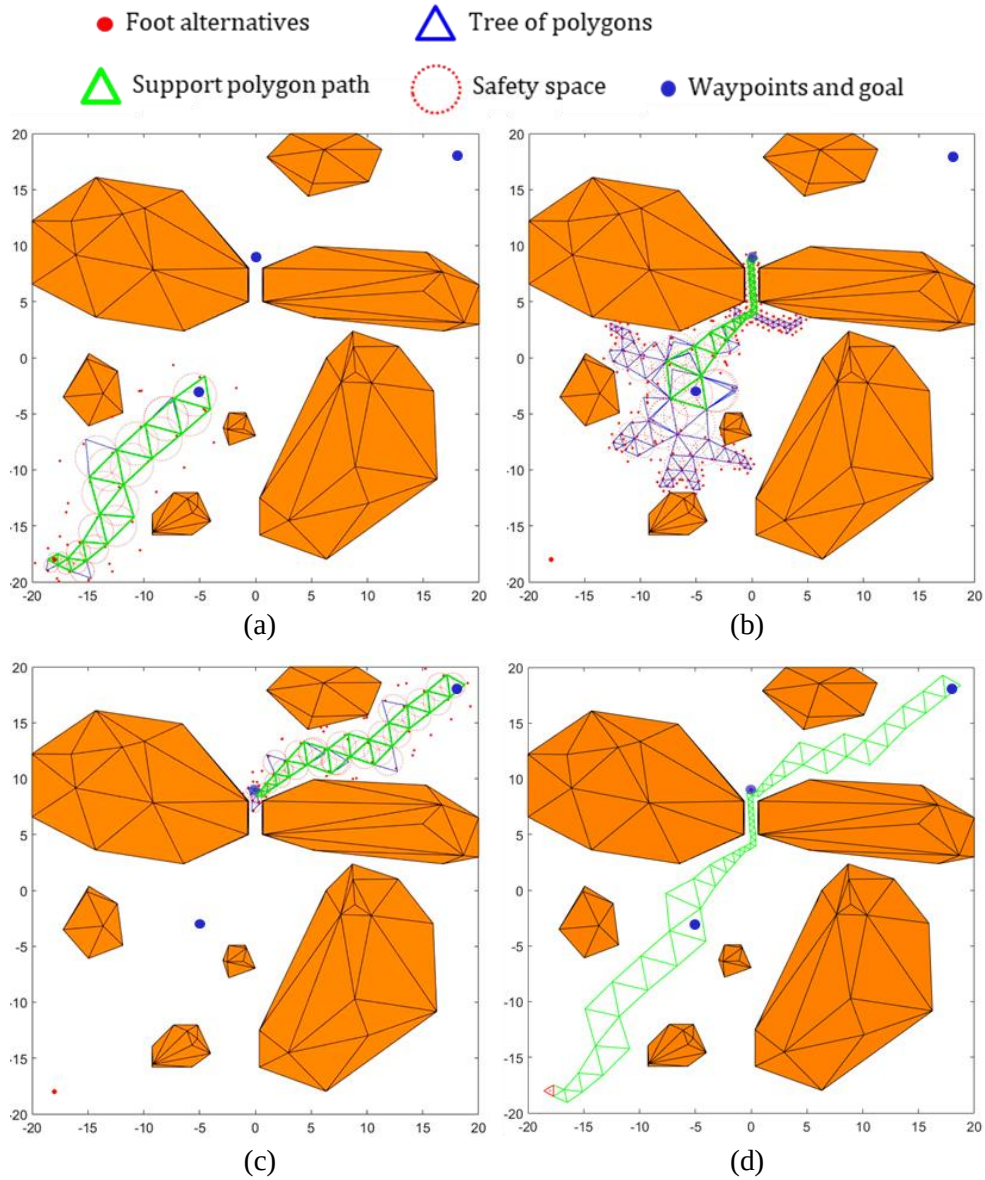


Fig. 5.2.7. Result of support polygon planning.
Support polygon path in (a) Zone 1, (b) Zone 2, (c) Zone 3.
(d) Support polygon path from start position to goal position

In Step 2, center of mass planning, path of the desired center of mass was generated by connecting centroid of polygon and midpoint of each polygon edges as in Fig. 5.2.8. Speed of desired center of mass was set to be 13.33 mm/s which is one third of the maximum velocity of members. The desired center of ass was set to be stops at the centroid of support polygon for 10 seconds, in order to provide time to adjust VTT's configuration to nominal configuration.

After node position planning, the resultant motion of VTT is shown in Fig. 5.2.9 to Fig. 5.2.12. Constrains graphs are shown in Fig. 5.1.13. In constraint graph, the maximum and minimum values among all edges or combinations are presented.

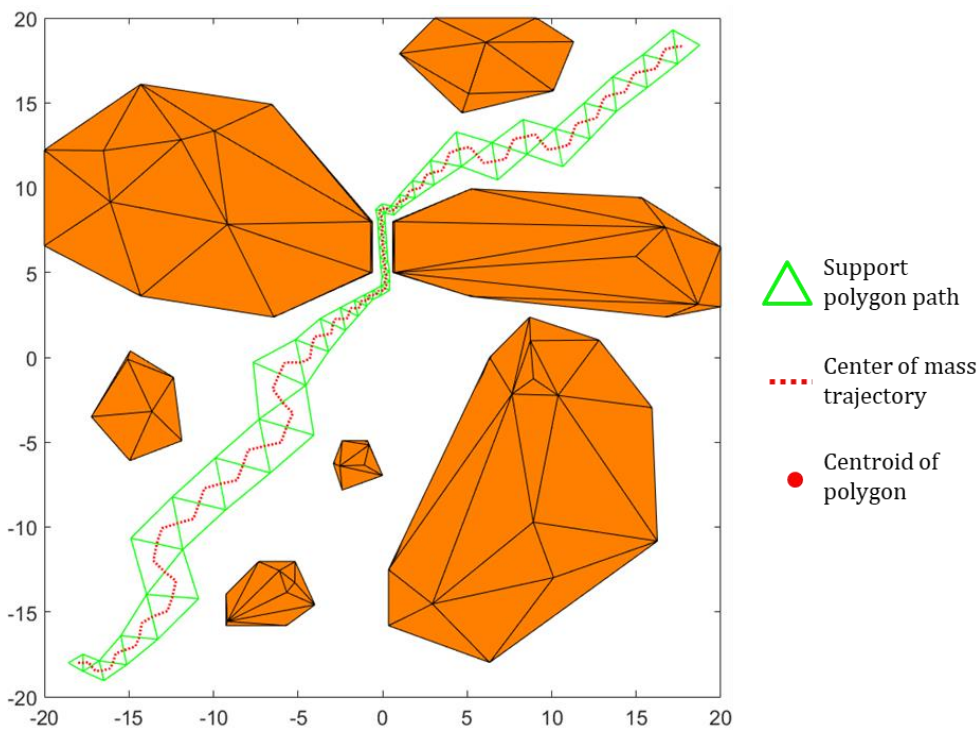
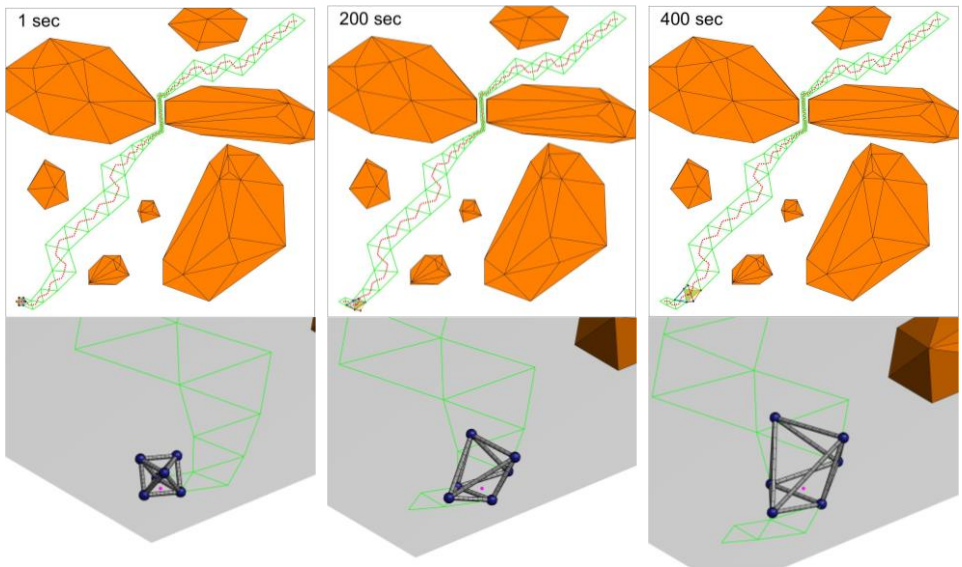
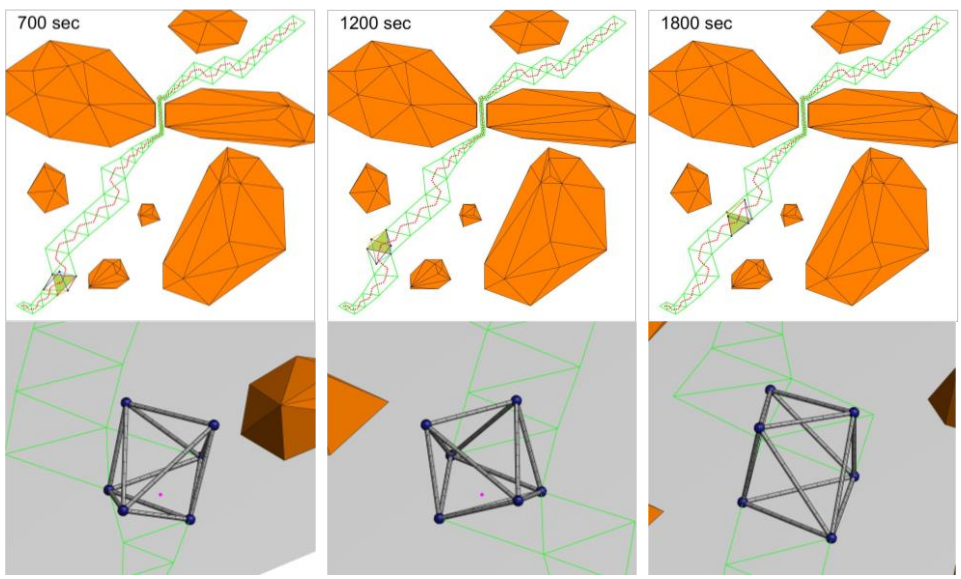


Fig. 5.2.8. Center of mass trajectory

In Step 3, node position planning, nominal length was set the same as that of support polygon. Time step size Δt was set to be 1 second. The resultant motion of VTT is shown in Fig. 5.2.9 to Fig. 5.2.12. Constrains graphs are shown in Fig. 5.2.13. In constraint graph, the maximum and minimum values among all edges or combinations are presented. The black dotted line denotes the constraint boundary.

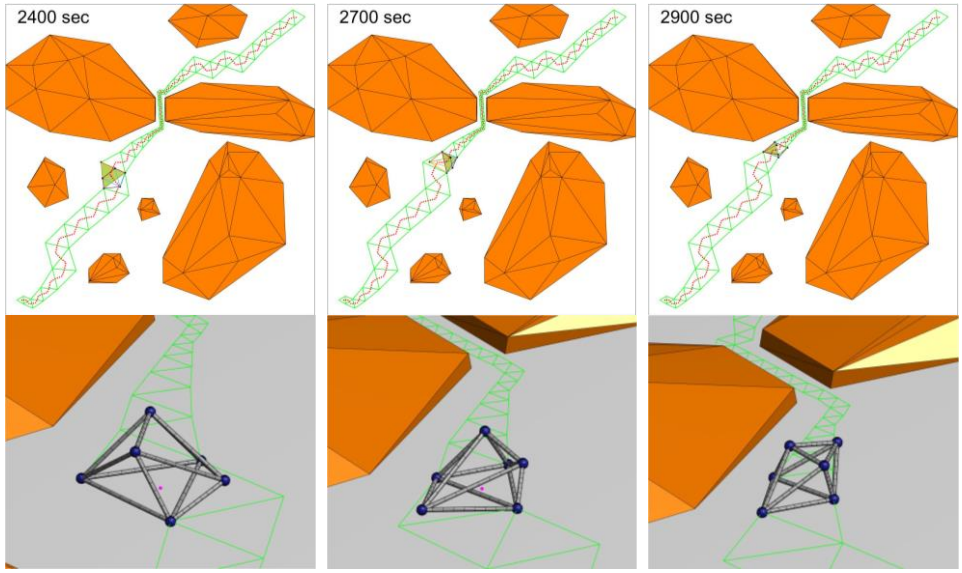


1 sec 200 sec 400 sec

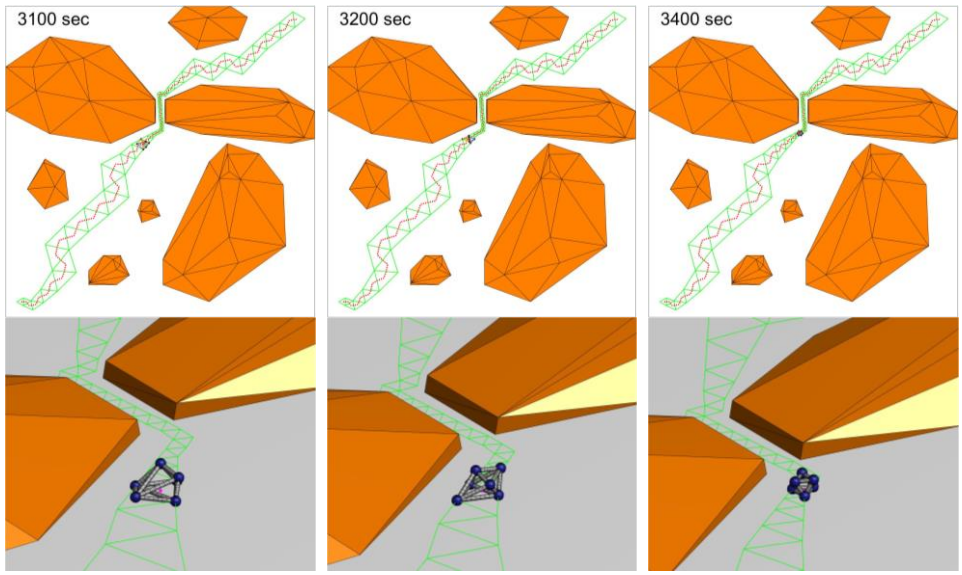


700 sec 1200 sec 1800 sec

Fig. 5.2.9. Simulation of VTT locomotion (1 sec to 1800 sec)

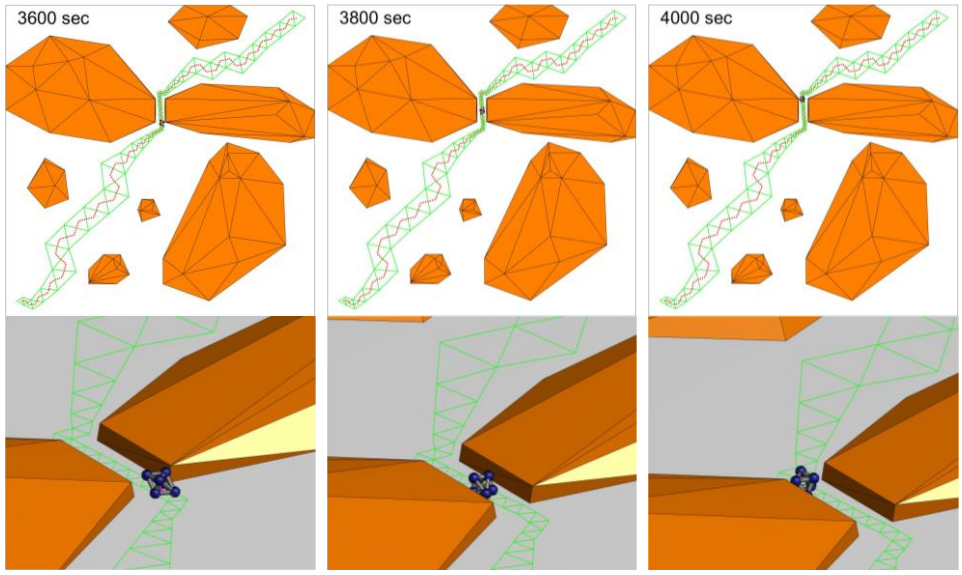


2400 sec 2700 sec 2900 sec

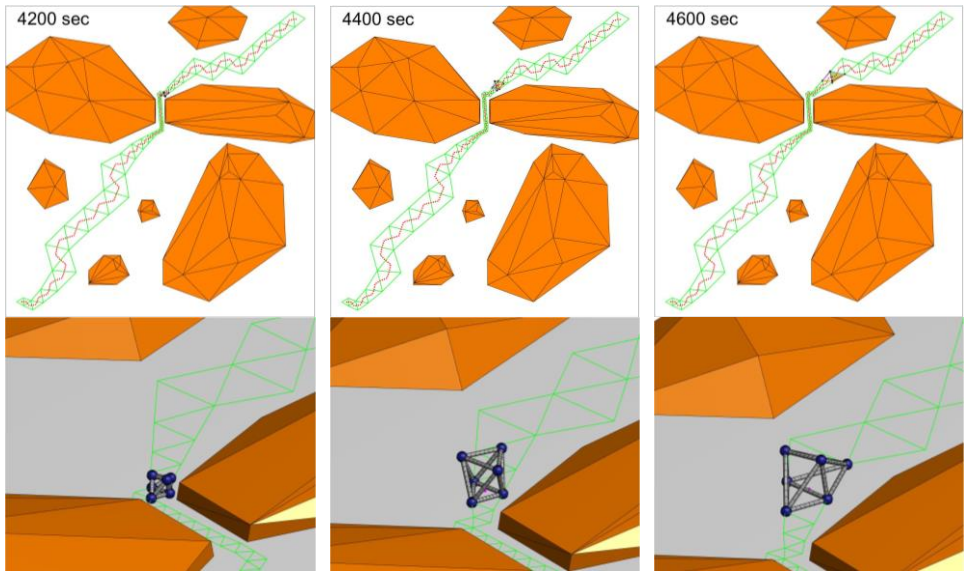


3100 sec 3200 sec 3400 sec

Fig. 5.2.10. Simulation of VTT locomotion (2400 sec to 3400 sec)

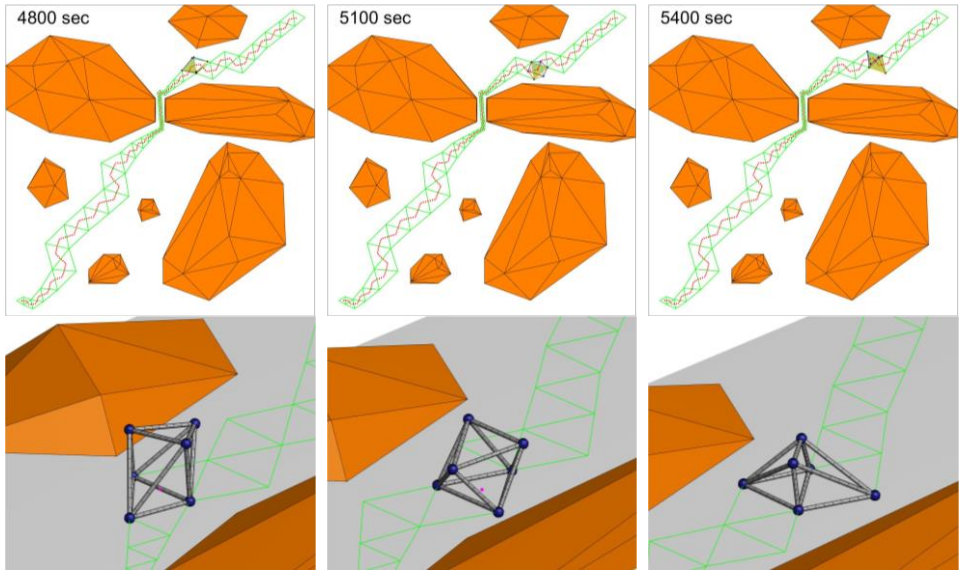


3600 sec 3800 sec 4000 sec

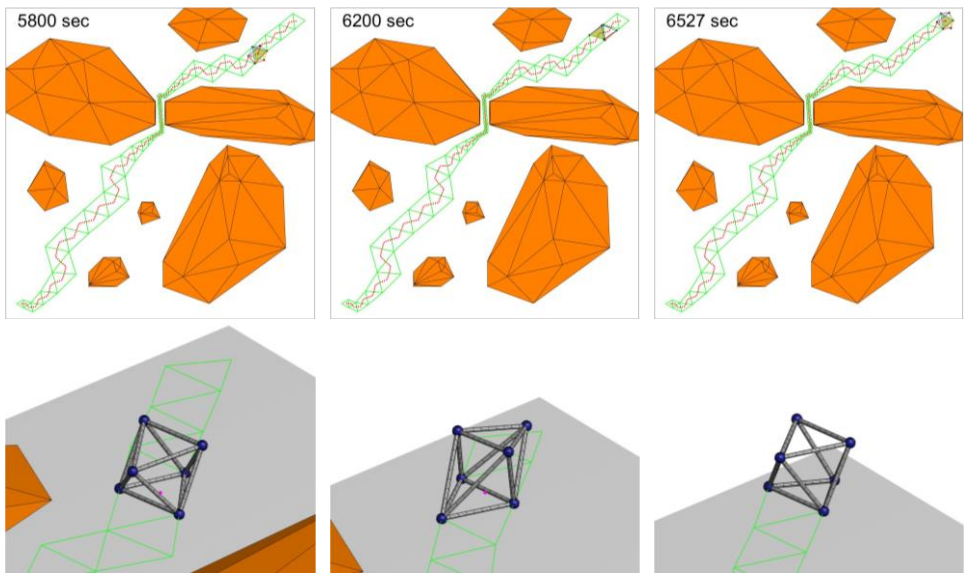


4200 sec 4400 sec 4600 sec

Fig. 5.2.11. Simulation of VTT locomotion (3600 sec to 4600 sec)



4800 sec 5100 sec 5400 sec



5800 sec 6200 sec 6527 sec

Fig. 5.2.12. Simulation of VTT locomotion (4800 sec to 6527 sec)

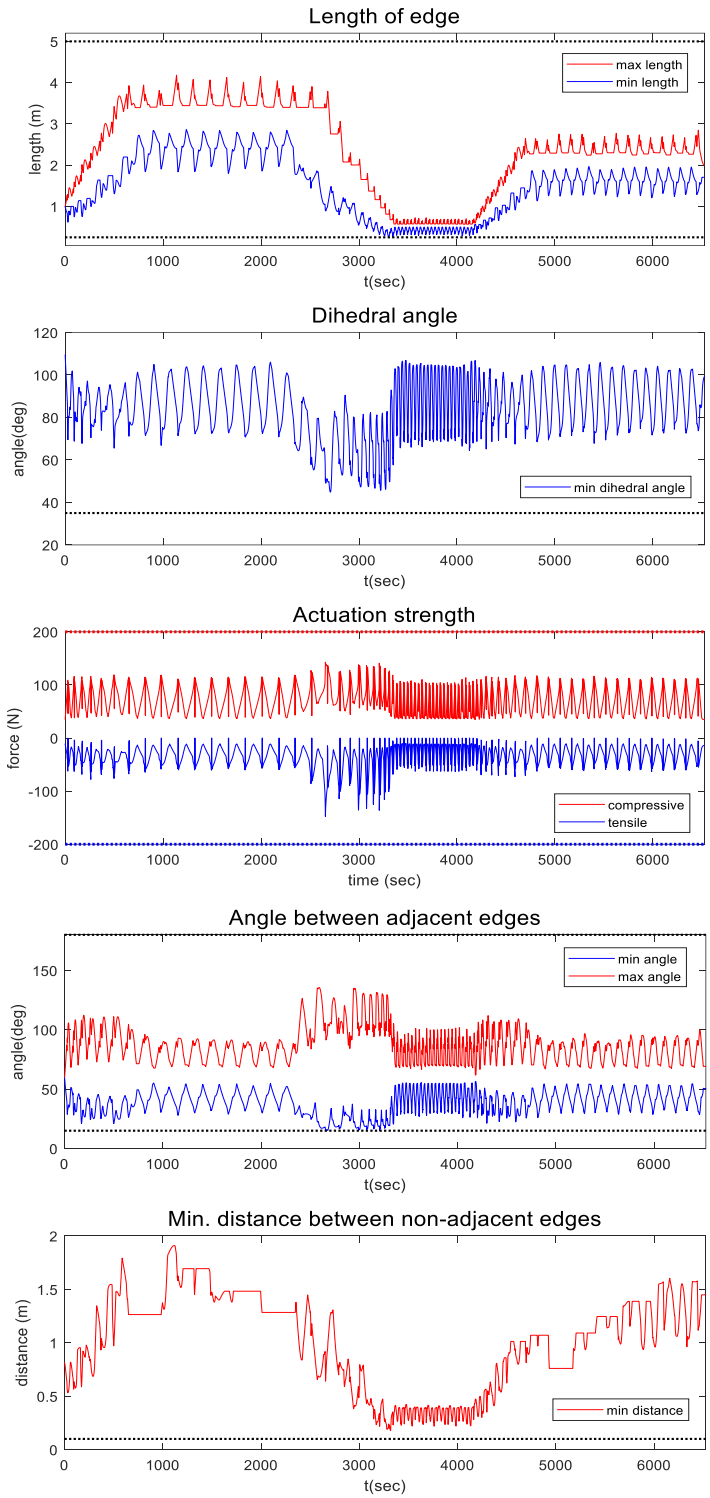


Fig. 5.2.13. VTT constraint graph

The simulation shows that VTT successfully change its nominal length to move in wide spaces and pass through narrow passage. Fig. 5.2.14 is length of edges graph that divides the section according to the time that VTT stays at each zone. The dotted line in the middle of the graph denotes the desired nominal length in each zone. As in the graph, the length changes gradually to desired nominal length at start of the zone. Once the nominal length reaches desired nominal length, the lengths fluctuates stably near the desired nominal length. This shows VTT can successfully adjust its nominal length to desired value after several steps of rolling.

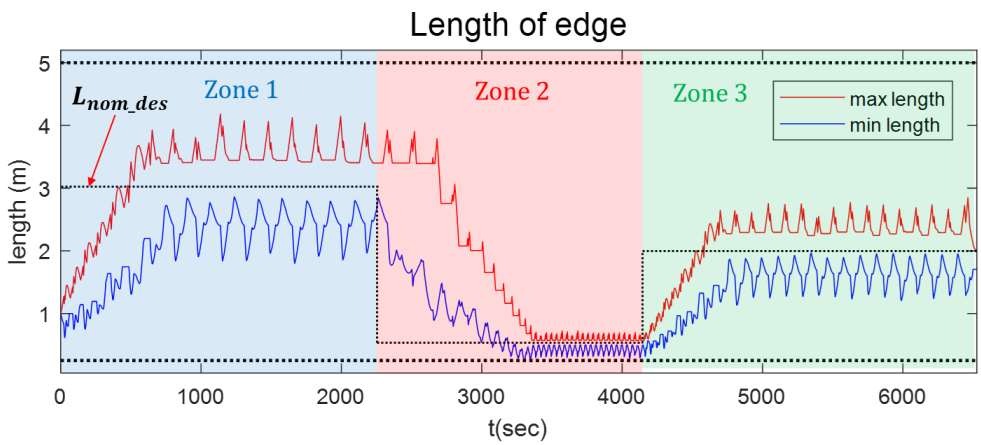


Fig. 5.2.14. Length of edge graph

Chapter 6

Conclusion

In this thesis, I introduced design of Variable Topology Truss (VTT) and proposed stable rolling locomotion algorithm for VTT that can applicable to actual situation and various environment.

VTT is first truss type modular robot that can self-reconfigure its topology. Self-reconfiguration is realized by specially designed members and nodes. A member of VTT is composed of a Spiral Zipper. Spiral Zipper is novel linear actuator that change its length by winding and unwinding its zipper shaped band. Spiral Zipper has high extension ratio and has high compressive strength [16]. However, Spiral Zipper alone has comparably low tensile and shear strength. To compensate these, Support Ring and Tensioner is attached to Spiral Zipper. Support Ring, with rubber rollers on Lazy Susan, protect the weak meshing point of Spiral Zipper band to increase shear strength. Tensioner, with Series Elastic Actuator (SEA) system, apply compressive preload to the Spiral Zipper band to increase tensile strength. Using Spiral Zipper with high extension ratio and high strength, VTT can stably and flexibly change its geometric shape to facilitate topology reconfiguration.

A node of VTT is composed of Passive Member-Ends, a Sphere, and a Master Member-End. Passive Member-End is a linkage type spherical joint based on Concentric Multilink Spherical (CMS) joint [10][25]. With its structure, Passive Member-End has large angle range, accommodate many members, and makes

member axes meet at one point which makes the members be subjected to pure axial force. Sphere is made of steel shell and protect Passive Member-Ends and members from ground. Master Member-End is spherical manipulator that is built in Sphere. The arm of Master Member-End is composed of one-way chain and moves in polar angle and azimuthal angle direction. With its structure, Master Member-Ends can move members for topology reconfiguration

Considering the structure of VTT, stable rolling locomotion algorithm was proposed. The proposed algorithm is stable in three aspect. First, proposed algorithm is based on optimization with high planning success rate. Second, proposed algorithm can avoid obstacles. Third, proposed algorithm can prevent VTT from being damaged from ground. To achieve this aspects, proposed locomotion algorithm is composed of 3 steps; support polygon planning, center of mass planning and node position planning.

In support polygon planning, support polygon path is planned by newly proposed random search algorithm, Polygon-Based Random Tree (PRT). Motivated from Rapidly-Exploring Random Tree (RRT), PRT algorithm randomly generate polygon tree to reach goal position. Edge length of generated polygon is near nominal length, but has flexibility with distortion margin, so it can effectively find a support polygon path that can avoid obstacle. The test simulation results show that, PRT probably has probabilistic completeness property which ensure that PRT finds support polygon path, if one exists, regardless of initial position and obstacles.

In center of mass planning, trajectory of desired projected center of mass is plan. To maximize the stability margin, the path generated by connecting centroid of support polygon and midpoint of support polygon edges. Planned support polygon path and center of mass trajectory guide VTT to have good-conditioned shape which configuration is far from constraints and makes locomotion planning

more likely to success even in complex and large environment.

In node position planning, position of VTT's nodes is planned by optimization to follow planned support polygon path and center of mass trajectory. In addition, node position is planned for Non-Impact Rolling to prevent damage from the ground. Non-Impact Rolling is composed of three motion phases: moving phase, landing phase and transient phase. By alternating three phases, VTT can always maintain stability and never be subjected to impact from ground when rolling.

The algorithm was verified by two case study. In case study 1, locomotion planning and simulation was performed considering actual constraints of VTT. To avoid collision between VTT and obstacle, safety space was defined and considered in support polygon planning. The result shows that VTT successfully reaches the goal while avoiding obstacles and satisfying constraints.

In case study 2, locomotion planning and simulation was performed in the environment having wide space and narrow passage. The operational space was divided according to the type of terrain. Then, Nominal length of VTT was set to be large in wide space to move efficiently, and set to be small in narrow passage to pass through it. The result shows that VTT successfully reaches the goal while changing its nominal length in different terrain.

Bibliography

[1] *Shoring operations guide*, 4th ed. U.S. Army Corps of Engineers Urban Search and Rescue Program, 2015.

[2] A. Spinos, D. Carroll, T. Kientz, and M. Yim, "Variable topology truss: Design and analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2717-2722.

[3] S. Jeong, B. Kim, S. Park, E. Park, A. Spinos, D. Carroll, T. Tsabedze, Y. Weng, T. Seo, M. Yim, F. C. Park, and J. Kim, "Variable topology truss: Hardware overview, reconfiguration planning and locomotion," in *15th International Conference on Ubiquitous Robots (UR)*, 2018, pp. 610-615.

[4] C. Liu, S. Yu, and M. Yim, "Shape morphing for variable topology truss," in *16th International Conference on Ubiquitous Robots (UR)*, 2019

[5] C. Liu, and M. Yim, "Reconfiguration Motion Planning for Variable Topology Truss," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1941-1948.

[6] C. Liu, S. Yu, and M. Yim, "A Fast Configuration Space Algorithm for Variable Topology Truss Modular Robots," in *International Conference on Robotics and Automation (ICRA)*, 2020.

[7] E. Park, J. Bae, S. Park, and T. Seo, "Reconfiguration Solution of a Variable Topology Truss: Design and Experiment," *IEEE Robotics and Automation Letters*, Jan. 2020.

[8] S. Park, E. Park, M. Yim, J. Kim, and T. Seo, "Optimization-Based Non-impact Rolling Locomotion of a Variable Geometry Truss," *IEEE Robotics and*

Automation Letters, vol. 4, no. 2, pp. 747-752, Jan. 2019.

[9] S. Park, J. Bae, S. Lee, M. Yim, J. Kim, and T. Seo, "Polygon-Based Random Tree Search Planning for Variable Geometry Truss Robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 813-819, Jan. 2020.

[10] G. J. Hamlin, and A. C. Sanderson, "Tetrobot: A modular approach to parallel robotics," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 42-50, Mar. 1997.

[11] K. Miura, H. Furuya, and K. Suzuki, "Variable geometry truss and its application to deployable truss and space crane arm," *Acta Astronautica*, vol. 12, no. 7-8, pp. 599-607, July. 1985.

[12] P. C. Hughes, W. G. Sincarsin, and K. A. Carroll, (1991). "Trussarm—a variable-geometry-truss manipulator," *Journal of Intelligent Material Systems and Structures*, 1991, vol. 2, no. 2, pp. 148-160.

[13] A. Spinos, and M. Yim, "Towards a variable topology truss for shoring," in *14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2017, pp. 244-249.

[14] S. Curtis, M. Brandt, G. Bowers, G. Brown, C Cheung, ... , and J. Vranish, "Tetrahedral robotics for space exploration," in 2007 IEEE Aerospace Conference, 2007, pp. 1-9.

[15] A. Lyder, R. F. M. Garcia, and K. Stoy, "Mechanical design of odin, an extendable heterogeneous deformable modular robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 883-888.

[16] F. Collins, and M. Yim, "Design of a spherical robot arm with the spiral zipper prismatic joint," in *IEEE international conference on robotics and automation (ICRA)*, 2016, pp. 2137-2143.

[17] W. H. Lee, and A. C. Sanderson, "Dynamic rolling, locomotion planning, and control of an icosahedral modular robot," in *IEEE/RSJ International Conference on*

Intelligent Robots and Systems (IROS), 2000, vol. 3, pp. 2178-2183.

[18] M. Abrahante, A. Silver, and L. Wendt, “Gait design and modeling of a 12-tetrahedron walker robot,” in *Thirty-Ninth Southeastern Symposium on System Theory*, 2007, pp. 21-25.

[19] M. Abrahantes, L. Nelson, and P. Doorn, ”Modeling and gait design of a 6-tetrahedron walker robot,” in *42nd Southeastern Symposium on System Theory (SSST)*, 2010, pp. 248-252.

[20] S. M. Motahari-Bidgoli, M. J. Mahjoob, and S. Davaria, “Simulation and analysis of a TET-walker robot motion,” in *Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, 2014, pp. 914-919.

[21] K. Stoy, A. Lyder, R. F. M. Garcia, and D. J. Christensen, “Hierarchical robots,” in *IROS Workshop on Self-Reconfigurable Modular Robot*, 2007

[22] M. Yim, “Locomotion with a unit-modular reconfigurable robot,” Ph.D. Dissertation, Stanford University, 1994.

[23] N. Usevitch, Z. Hammond, S. Follmer, and M. Schwager, “Linear actuator robots: Differential kinematics, controllability, and algorithms for locomotion and shape morphing,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5361-5367.

[24] K. Kong, J. Bae, and M. Tomizuka, “Control of rotary series elastic actuator for ideal force-mode actuation in human–robot interaction applications,” *IEEE/ASME transactions on mechatronics*, vol. 14, no. 1, pp. 105-118, Feb. 2009.

[25] G. J. Hamlin, and A. C. Sanderson, “A novel concentric multilink spherical joint with parallel robotics applications,” *IEEE international conference on robotics and automation (ICRA)*, 1994, pp. 1267-1272.

[26] Vicon Industries, Inc. (2019, June 10). *What is motion capture* [Online]. Available: https://www.vicon.com/about-us/what-is-motion-capture/

[27] *Vicon Tracker User Guide*, Vicon Industries, Inc. 2018.

[28] S. Jain, and S. N. Kramer, “Forward and inverse kinematic solution of the variable geometry truss robot based on an N-celled tetrahedron-tetrahedron truss,” *Journal of Mechanical Design*, vol. 112, no. 1, pp. 16-22, Mar. 1990.

[29] F. Naccarato, and P. Hughes, “Inverse kinematics of variable-geometry truss manipulators,” *Journal of Robotic Systems*, vol. 8, no. 2, pp. 249-266, Apr. 1991.

[30] P. H. Tidwell, C. F. Reinholtz, H. H. Robertshaw, and C. G. Horner, “Kinematic analysis of generalized adaptive trusses,” in *Joint U.S./Japan Conference on Adaptive Structures*, 1991, pp. 772-791.

[31] R. R. Archer, S. H. Crandall, N. C. Dahl, T. J. Lardner, and M. S. Sivakumar, “An Introduction to Mechanics of Solids,” 3rd ed. New Delhi, India: Tata McGraw-Hill Education, 2012.

[32] V. J. Lumelsky, “On fast computation of distance between line segments,” *Information Processing Letters*, vol. 21, no. 2, pp. 55-61, Aug. 1985

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to algorithms,” 3rd ed. Cambridge: The MIT press, 2009.

[34] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” 1998.

[35] S. M. LaValle, and J. J. Kuffner, “Rapidly-exploring random trees: Progress and prospects,” *Algorithmic and computational robotics: new directions*, no. 5, pp. 293-308, Apr. 2000.

[36] J. C. Latombe, “Robot motion planning,” 1st ed. Berlin, Germany: Springer Science & Business Media, 1991.

[37] M. Van Kreveld, O. Schwarzkopf, M. de Berg, and M. Overmars, “Computational geometry algorithms and applications,” 3rd ed. Berlin, Germany: Springer, 2008.

[38] L. Krick, M. E. Broucke, and B. A. Francis, “Stabilisation of infinitesimally rigid formations of multi-robot networks,” *International Journal of control*, vol. 82,

no. 3, pp. 423-439, Mar. 2009.

[39] D. Babić, D. J. Klein, I. Lukovits, S. Nikolić, and N. Trinajstić, "Resistance-distance matrix: A computational algorithm and its application," *International Journal of Quantum Chemistry*, vol. 90, no. 1, pp. 166-176, Dec. 2001.

[40] Y. Nesterov, and A. Nemirovskii, "Interior-point polynomial algorithms in convex programming," vol. 13, Philadelphia: SIAM, 1994.

[41] Mathworks, Inc. (2020). *checkCollision* [Online]. Available: https://www.math-works.com/help/robotics/ref/checkcollision.html

Abstract in Korean

가변 토폴로지 트러스 (Variable Topology Truss, VTT)는 토폴로지와 기하학적 형상의 재구성이 가능한 트러스 구조의 모듈 로봇이다. 본 논문에서는 VTT의 설계 구조를 소개하고 VTT의 안정적인 주행을 알고리즘을 제안한다.

VTT는 토폴로지와 기하학적 형상의 재구성을 위해 특수한 구조의 멤버와 노드를 가진다. VTT의 멤버는 높은 압축비, 가벼운 중량, 높은 강도를 가진 신개념 선형 구동기인 스파이럴 지퍼로 구성되어 있다. VTT의 노드는 패시브 멤버 엔드와 마스터 엔드로 구성되어 있다. 패시브 멤버는 링키지 구조의 3 자유도 관절로, 넓은 각도 구동 범위를 가지고 있고 많은 수의 멤버를 연결할 수 있다. 마스터 멤버 엔드는 노드 부의 내장된 구형 매니퓰레이터로, 토폴로지 재구성 시 멤버를 이동시키는데 사용된다.

VTT는 기하학적 형상을 변화하여 구르는 움직임을 통해 주행한다. VTT의 주행 알고리즘은 서포트 폴리곤 계획 단계, 무게 중심 계획 단계, 노드 위치 계획 단계로 이루어진다. 서포트 폴리곤 계획 단계에서는 새롭게 제안된 무작위 탐색 (random search) 알고리즘인 Polygon-Based Random Tree (PRT)을 적용해 서포트 폴리곤의 경로를 계획한다. 무게 중심 계획 단계에서는 안정성을 최대화하는 VTT의 무게 중심 궤적을 계획한다. 계획된 서포트 폴리곤 경로와 무게 중심 궤적을 VTT가 제한 조건으로부터 먼 좋은 상태의 형상을 유지하게 하여 복잡한 환경에 대해서도 경로 계획이 실패하지 않고 안정적으로 이루어질 수 있도록 한다.

노드 위치 계획 단계에서는 서포트 폴리곤 경로와 노드 위치의 궤적을 추종하는 노드 위치 궤적을 계획한다. 이 과정에서 비충격 롤링 이동 알고리즘 (Non-Impact Rolling locomotion algorithm)을 적용하여 지면과의 충돌로 인한 충격이 일어나지 않는 궤적을 계획한다.

실제 VTT의 제한 조건을 반영한 모델에 본 알고리즘을 적용하여 시뮬레이션을 수행한 결과, VTT가 모든 제한 조건을 만족하고 장애물을 회피하면서 목표 지점에 도달할 수 있음을 확인하였다.

주요어 : 모듈 로봇, 가변 토폴로지 트러스, 주행 경로 계획,
장애물 회피, 무작위 탐색 알고리즘

학번 : 2013-20675