Tree Climbing Robot: Design, Kinematics and Control

LAM, Tin Lun

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Automation and Computer-Aided Engineering

The Chinese University of Hong Kong
September 2010

Thesis /Assessment Committee

Professor Yun-hui Liu (Chair)

Professor Yangsheng Xu (Thesis Supervisor)

Professor Changling Wang (Committee Member)

Professor Lilong Cai (External Examiner)

Abstract

Climbing robots have become a hot research topic in recent decades. Most research in this area focuses on climbing manmade structures, such as vertical walls, glass windows, and structural frames. Little research has been conducted specifically on climbing natural structures such as trees. The nature of trees and manmade structures is very different. For example, trees have an irregular shape and their surface is not smooth. Some types of trees have soft bark that peels off easily. Hence, most of the climbing methods for manmade structures are not applicable to tree climbing.

Preventing trees from failing is important to protect human life and property in urban areas. Most trees in urban areas require regular maintenance. To reach the upper parts of a tree to perform such maintenance, workers need to climb the tree. However, tree climbing is dangerous, the development of a tree climbing robot is important to assist or replace humans works.

Several robots have been designed to climb trees such as WOODY and RiSE. However, these robots are limited to climbing straight tree trunks, and cannot climb trees that are curved or have branches. As branches and curvature are present in almost all trees, the application of these robots is strongly restricted.

As a result, this dissertation proposes a novel type of tree climbing robot, named Treebot, which has high maneuverability on trees. The design of Treebot was inspired by arboreal animals such as squirrels and inchworms. The applied extendable continuum maneuvering mechanism has large workspace and high degrees of freedom. It allows Treebot to perform various actions, such as moving between trunk and branches. Treebot is able to grip the surface of trees tightly with a wide range of gripping curvature. It enables Treebot to grip from a big tree trunk to small branches. The special gripping mechanism allows zero energy consumption in static gripping. Although Treebot has high maneuverability, it is compact, lightweight, and only five actuators are used in total. By installing proper equipments, Treebot can assist workers to perform forestry tasks such as inspection and maintenance. It can also be used as a mobile surveillance system to observe behaviors of both ground and arboreal animals.

In addition to presenting the mechanical design of Treebot, this dissertation also proposes several autonomous tree climbing algorithms. Making a robot climb a tree autonomously is a challenging task, as trees are complex and irregular in shape. However, a certain level of autonomous climbing ability is needed to simplify the operational use of Treebot. The proposed works include autonomous climbing on unknown environment and global path planning on known environment.

論文摘要

攀爬機器人是近年的一個熱門的研究課題。大部分攀爬機器人的研究都集中於攀爬城市建設，例如牆、玻璃窗和支架。然而，攀爬樹的研究卻不多。樹的特性與城市建設有很大的差異，樹有不規則的外形，而且表面粗糙，有些樹木的樹皮更會容易脫落。因此，大多數攀爬城市建設的機器人並不適用於攀爬樹木。

城市綠化是現今的大趨勢，城市四周都會種了各式各樣的樹。因此，防止樹木倒塌以保障途人的安全及財產顯得極為重要。為了令樹木健康生長，必須對樹木作出定期檢查。當要到達樹木上較高的位置進行檢查和維護工作時，工人必需爬到樹上。由於爬樹是一件危險的高空工作，建立一個爬樹機器人以協助或代替工人的工作有著重大的意義。

目前, 只有少數機器人擁有爬樹的功能，如WOODY和RiSE。但這些機器人都只能攀爬筆直的樹幹。他們不能爬上彎曲的樹幹或樹的分支上。由於樹的分支和彎曲的形狀在一般的樹木都是很常見的，這些機器人在應用上受到了很大的限制。

有見及此，本論文提出了一種新型的爬樹機器人，名為Treebot。Treebot的設計靈感來自樹上生活的動物，如松鼠，尺蠖等。Treebot使用連續式可申縮及彎曲結構，使之具有高機動性及靈活性，能夠在樹上執行各種動作，如攀爬彎曲的樹幹和樹幹與樹枝之間的過渡，有如生活在樹上的動物一樣。此外，Treebot的抓附裝置能夠適應於不同形狀及大小的夾持表面，它使Treebot在大小不一的樹幹和樹枝之間走動而不需更換相應大小的抓附裝置。Treebot的抓附裝置的另一個特點是在靜態夾持時沒有能源消耗，因此，Treebot能長時間停留在樹上進行任務。盡管Treebot擁有很高的機動性，它只由五個微型電機帶動，因此它能達至小型及輕量。通過安裝適當

的設備，Treebot可以協助人類在樹上執行各種任務，如檢查和維修。它亦可以作為移動監視系統來觀察地面的情況和樹上的動物。

令機器人能夠自主地在樹上爬行是一個極具有挑戰性的任務，因為樹的特性和形狀是不規則而且複雜的。然而，一定程度的自主攀爬能力對於簡化Treebot的使用有著重大的幫助。因此，本論文提出了Treebot的自主攀爬方案。當中包括對未知環境的探索與攀爬路徑規劃，以及對已知環境的攀爬路徑規劃。

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Xu Yangsheng. He originally suggested the research topic to me, and gave me continuous encouragement and invaluable guidance. It is impossible to sufficiently express my gratitude to him for taking me on as his student and for sharing with me his vast wealth of engineering knowledge and philosophy of life. I thank him for all that he has taught me.

I would also like to show my gratitude to the members of my dissertation committee, Professor Cai Lilong, Professor Wang Changling, and Professor Liu, Yun-hui. I greatly appreciate them taking the time to evaluate this dissertation.

I am also sincerely thankful to the members of the Advanced Robotics Laboratory at the Chinese University of Hong Kong for their help and enlightening discussions, and especially Mr. Chung Wing-Kwong, Mr. Hou Kai-Wing, Mr. Chen Yongquan, Mr. Qian Huihuan and Mr. Yan Jingyu. I would also like to acknowledge Allan Mok for his technical support.

I would like to thanks my friends at the Chinese University of Hong Kong, Mr. Lam Hiu-Man and Lau Tai-Kit, who shared their valuable knowledge with me and gave me continuous encouragement.

Last but not least, my deepest gratitude goes to my family for their unconditional love and support throughout my life. This dissertation would simply not have been possible without them. I am indebted to my parents for their care, love, spiritual support, and encouraging me to pursue my interests. I also thank them for always being there when I needed them.

Contents

List of Figures

List of Tables

insert the chapters

Chapter 1

Introduction

1.1 Background

Climbing robot is a challenging research topic that has gained much attention from researchers. Most of the climbing robots reported in the literature are designed to work on manmade structures, such as vertical walls and glass windows [1–7, 39], or structural frames [9–12, 40]. Few climbing robots have been designed to work on natural structures such as trees. Trees and manmade structures are very different in nature. Tree surfaces are seldom flat and smooth, and some trees have soft bark that peels off easily. In addition, the inclined angle on trees are usually not vertical. Hence, most of the climbing methods for manmade structures are not applicable to tree climbing.

To perform forestry tasks, workers often attach a tool to the end of a long pole to reach the target position. However, this becomes infeasible if the target position is too high. Alternatively, workers can reach the desired position by using an aerial ladder truck. However, the use of this equipment is not always possible due to the access limitation. In such cases, workers can only climb up the tree to perform the task. As tree climbing is a dangerous task for human being, robots are expected to assist or replace humans in performing these tasks.

1.2 Related Work

The literature contains several reports of robots designed to climb trees as illustrated in Fig. 1.1.

WOODY [13] is one of the climbing robots designed to replace human workers in removing branches on trees. The robot climbs by encircling the entire tree trunk. The size of the robot is thus proportional to the circumference of the trunk. WOODY avoids branches by turning its body and opening the gripper, but the climbed tree trunk should be almost straight.

Kawasaki [14] also developed a climbing robot for tree pruning. It uses a gripping mechanism inspired by lumberjacks, and uses a wheel-based driving system for vertical climbing. Like WOODY, it also needs to encircle the entire tree trunk.

Aracil [8] proposed a climbing robot, CPR, that uses a Gough-Stewart platform to maneuver. It consists of two rings that are joined by six linear actuators through universal and spherical joints at each end. The gripping mechanism also requires the encircling of the entire tree trunk. However, it has greater maneuverability than the aforementioned two robots, and can climb a branchless tree trunk with a certain range of bending.

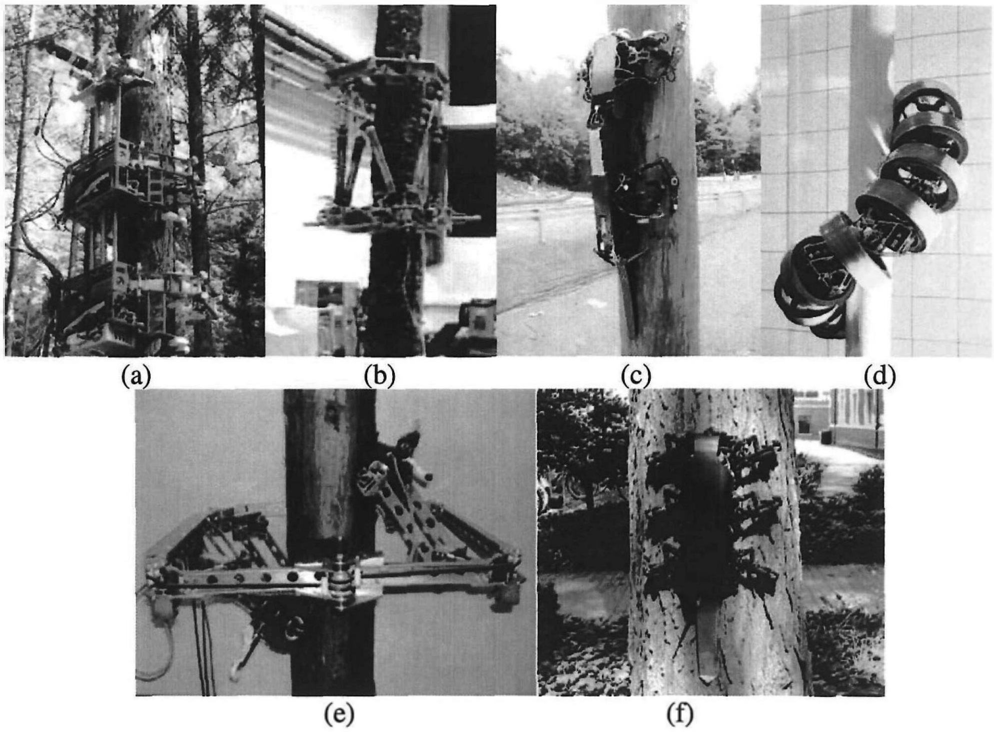


Figure 1.1: Prototypes of the existing tree climbing robot. (a) WOODY [13]; (b) Parallel Climbing Robot (PCR) [8]; (c) RiSE V3 [16]; (d) HyDRAS [55]; (e) Kawasaki's pruning robot [14]; (f) RiSE V2 [15].

RiSE V2 [15] is a wall climbing robot that imitates the movement of an insect in using six legs to maneuver. This robot has also been demonstrated to be able to climb trees. As the gripping mechanism only occupies a portion of gripping substrate, the size of the robot is independent of the climbing target. As a result, it is relatively small. However, it does not claim whether it can perform other movement on a tree, such as branche transitions or turning. RiSE V3 [16] is another type of climbing robot designed to climb straight poles with high speed.

HyDRAS [55] is a hyper-redundant snake-like robot that can climb on pole-like structure. It is believed to have the potential to climb a tree trunk. It climbs by wrapping its body around the pole in a helical shape, rotating its body about its own central axis to roll up the pole.

For those aforementioned tree climbing robots, the workspaces are restricted on tree trunks only. Trees with branches and an irregular shape are not considered. They cannot act like tree living animals such as squirrels that can reach any position on an irregular shape of trees with branches.

1.3 Motivation of the Dissertation

The goal of this dissertation is to develop a novel type of tree climbing robot that can assist or replace humans in performing forestry tasks on trees. To achieve this goal, the robot should meet several requirements.

1. High maneuverability. The robot should be able to perform turning, branch transition, and climb trees with an irregular shape to increase the climbing workspace on trees.

2. Adhesion to tree surfaces with a wide range of curvatures. It enables the robot to climb from a large tree trunk to a small branch without needing to replace the gripper.

3. High payload capacity. It allows the robot to carry equipments for forestry works.

4. Lightweight and compact. This will make the robot portable and easy transport, which is important because the transportation of heavy machinery is not possible in some environments in which it might be used.

5. Energy saving. As a self-contained field robot, power is provided by an integrated battery. As the capacity of battery is limited, an energy saving design is crucial to prolong the robot's working hours.

6. An autonomous climbing ability. A certain level of autonomous climbing ability of the robot is helpful to simplify the use of the robot.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows.

Chapter 2 reviews existing tree climbing principles, both natural and artificial Six design principles are proposed to evaluate these climbing principles. The climbing principles are ranked according to the design principles to help select the best climbing principle on which to base the robot design.

In Chapter 3, the design of the proposed tree climbing robot "Treebot" is presented, including its mechanical design, working principles, and control architecture. Several climbing experiments are also presented to verify the climbing performance of the robot.

Chapter 4 presents an analysis of the gripping force of the robot's gripper. The relationships among the settings of the gripper, the gripping curvature, and the gripping force generated are studied. The settings for the gripper are optimized analytically to generate maximal gripping force. Numerous experimental results on different kinds of trees were acquired to evaluate the actual performance of the gripper.

Chapter 5 presents a kinematic model and workspace analysis of Treebot. The kinematic model is an extension of the kinematics of a continuum manipulator. The workspace represents the admissible gripping positions on a tree surface. The workspace analysis considers both admissible positions and directions of movement on the tree surface. Illustrations of the admissible positions in different situations are presented.

Chapter 6 proposes an autonomous climbing strategy for unknown environments. Tactile sensors are adopted to acquire local environmental information. Strategy for exploring, tree shape approximation, optimal path finding, and motion planning are then developed. Experimental results are presented to evaluate the performance of the proposed autonomous climbing strategy.

Chapter 7 presents a global path planning algorithm based on a known environment, and develops an efficient way of formulating the state space. The optimal path can be obtained by dynamic programming in linear time. A motion planning strategy for path tracking is also presented. Simulation results show the satisfactory performance of the proposed global path planning algorithm.

Chapter 8 concludes the dissertation, and suggests some future research topics.

□ **End of chapter.**

Chapter 2

Principle of Tree Climbing

2.1 Tree Climbing Methods in the Natural World

Arboreal habitats are complex environments that pose numerous challenges to arboreal animals. These include climbing on cylindrical branches with variable diameters and inclined angles, moving in narrow spaces and facing obstacles. Through millions of years of evolution, many types of arboreal animals have evolved and developed diverse methods for moving around in complex arboreal environments.

Many tiny animals, such as snails and worms, adopt wet adhesion to fasten themselves on trees. The wet adhesion includes the capillary adhesion and suction mechanisms. This method only provides limited adhesion force, but it is sufficient to support such tiny creatures. Although these animals use this fastening principle, their locomotion is different.

Snails have a single foot and move by pedal wave locomotion (waves of muscular contraction) [52]. The area of adhesion (the entire foot) is relatively large, which provides a relatively large adhesion force to support their relatively heavy bodies. However, this method allows only relatively slow movement.

Caterpillars have multiple feet that act as suctions to adhere on substrates. They move by using a sinusoidal gait which is quite similar to the pedal wave locomotion of snails.

Another common fastening principle observed in tiny animals is the Van der Waals force [53], which is commonly adopted by insects. Insect's feet have pulvilli (or pad) that can adhere to a smooth surface by the Van der Waals force (dry adhesion). Some insects also have micro claws on their feet to hook themselves onto non-smooth substrates.

For locomotion, insects use a tripod gait, whereby at least three feet are in contact with the substrate at any one time to make adhesion more stable.

Inchworms (also called loopers) use both wet adhesion (on the hind foot) and micro claws (on the fore foot) to attach themselves to surfaces. Their locomotion is unique. When they move, they use either the fore foot or the hind foot to attach to a substrate, and then bend their body to move the other foot to a new position alternatively. This can be treated as a kind of bipedal locomotion. Although inchworm locomotion is fast, the bending of the body makes the center of mass away from the climbing substrate, which generates large pitching force. Whereas, as inchworms are lightweight, the pitching force doesn't affect much of the movement.

The various climbing methods used by tiny animals treat branches as a flat surface rather than a cylindrical shape, and thus nearly any size of branch can be climbed in theory.

Small size animals such as squirrels and birds use claw penetration method to fasten on a tree. Claws can be used to interact with rough substrates. To generate a large gripping force, the angle of grip should be as large as possible. The claw-gripping method can be used to fasten onto smooth or rough branches with a grip of less than 180 degrees, which allows squirrels to climb large tree trunks.

As quadrupedal animals, squirrels use their limbs for locomotion, usually through pulse climbing, which is a type of dynamic stable locomotion that allows very fast movement.

Birds usually do not maneuver continuously throughout a tree, but instead move by hopping along a branch or flying.

Large size animals such as primates do not have sharp claws, and instead use frictional gripping to fasten on trees. They hold the body on a tree by encircling more than half (more than 180 degrees) of a branch with the forelimbs to pull the body toward the branches while both hind limbs push the body upward. Theoretically, this method provides a larger gripping force than when less than 180 degrees of the branch is gripped such as claw penetration, which is why it is more suitable for large animals [48]. This fastening method depends on the angle of the frictional force, and thus depends on the diameter of the branch. As a result, climbing on a larger branch results in a reduced gripping force.

Primates maneuver by moving diagonally opposing limbs at the same time, resulting in symmetrical footfall patterns [50].

Snakes also adopt frictional gripping for tree climbing, but the gripping mechanism is totally different to that of primates [47,49]. As snakes are long and thin, they fasten on a tree by sticking their body into the deep vertical furrows in the bark and then move by internal concertina locomotion [54]. The frictional area (contact area) with this method is large and the center of mass is extremely close to the substrate, which generate sufficient fastening force. However, it means that snakes can climb very rough-barked but not smooth-barked trees [51].

There are numerous approaches to tree climbing in the natural world, each of which is suitable for particular animals in a certain kind of arboreal habitat. Table 2.1 summarizes these natural tree climbing methods, and Fig. 2.1 illustrates the associated tree climbing animals and their means of locomotion.

2.2 Artificial Tree Climbing Methods

Arboreal animals provide many ideas for the design of tree climbing robots. There are several tree climbing robot have been developed as mentioned in Chapter 1. Most of these designs have adapted climbing approaches from thc natural world with suitable modifications, simplifications and creations, instead of implement directly to suit specific purposes and artificial design spaces.

WOODY, PCR and Kawasaki's pruning robots apply frictional gripping to attach to tree while RiSE V2 and V3 use claw penetration method. The locomotion of RiSE V2 imitates the insect's locomotion (tripod gait) while the locomotion of RiSE V3 imitates the primate's locomotion (diagonal foot-

Table 2.1: Summary of the tree climbing methods in the natural world.

Fastening principle	Maneuver Locomotion	Examples	Advantages	Disadvantages
Wet adhesion	Pedal wave	Snail	Provide large adhesive force	Slow locomotion
	Sinusoidal gait	Caterpillar		
Van der Waals force	Biped	Inchworm	Fast locomotion	Suffer from pitching force
	Tripod gait	Insect	Stable locomotion	Provide less fastening force
Claw penetration gripping	Hopping, flying	Bird	Wide range of gripping curvature	Fastening force depends on the property of gripping substrate
	Pulse climbing	Squirrel		
Frictional gripping	diagonal footfall patterns	Primate	Provide large fastening force	Need to encircle more than 180 degrees; Not easy to climb big tree trunk
	Internal concertina locomotion	Snake	Locomotion adaptive to many kinds of terrain	Restricted on bark with deep vertical furrows; Slow in motion

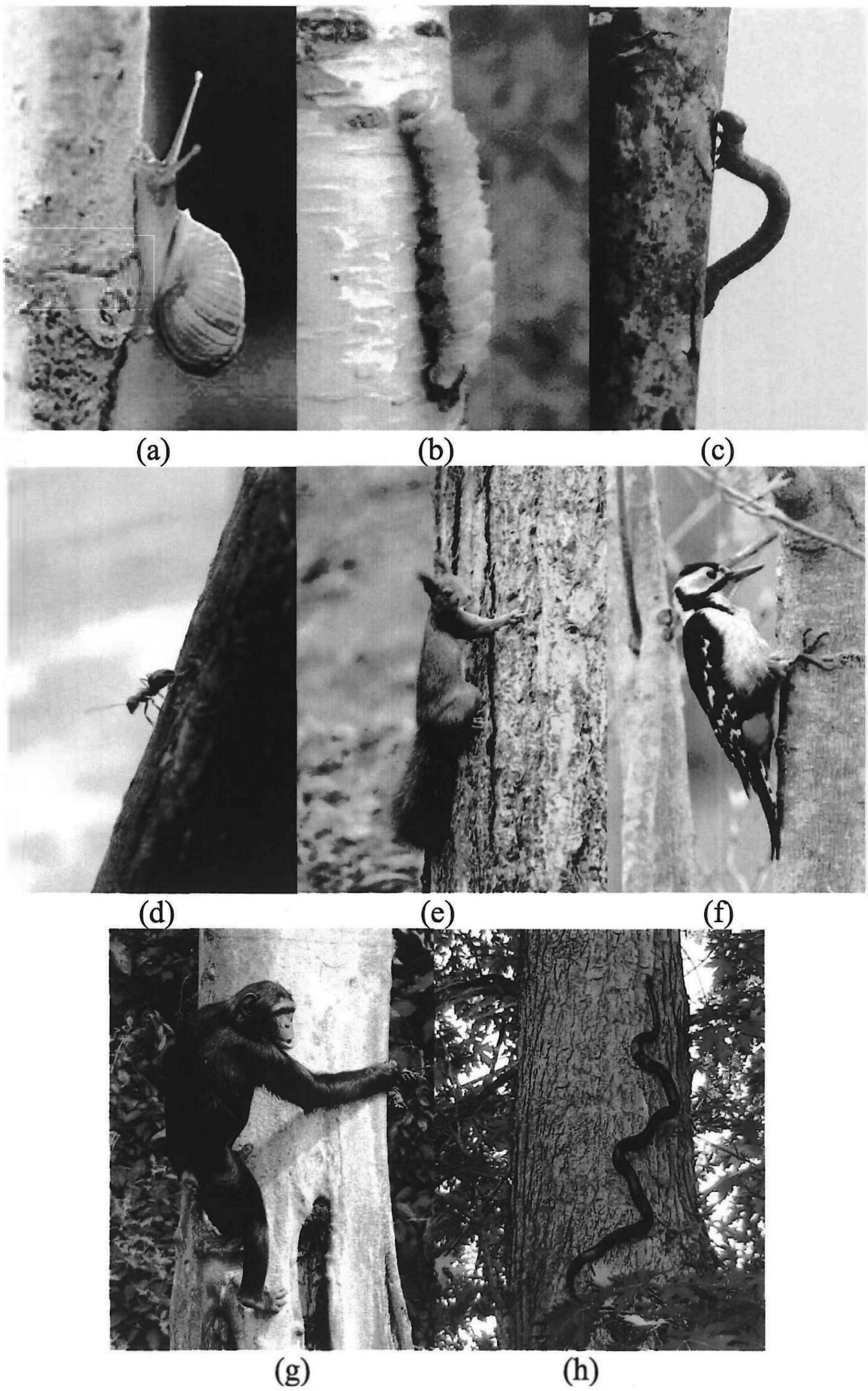


Figure 2.1: Examples of tree climbing animals. (a) Snail; (b) Caterpillar; (c) Inchworm; (d) Insect; (e) Squirrel; (f) Birds; (g) Primate and (h) Snake.

fall patterns). WOODY and PCR move by contract-extension their body in a manner similar to inchworm locomotion, except that the center of mass is closer to the gripping surface.

Scientists have also created climbing robots from completely new principles. For example, the Kawasaki's pruning robot uses wheel-driven principle to move, which would never be seen in the natural world. This wheel-driven method allows very fast movement. The locomotion that the snake-like robot (HyDRAS) uses to climb a pole is also creative. The robot encircles the pole by winding its body in a helical shape to fasten itself on the pole using the frictional gripping principle. It climbs by rotating its body along its own central axis, which can be treated as a kind of wheel-driven locomotion. This locomotion is faster than that of a real snake and real snake never move like this.

Table 2.2 summarizes the artificial tree climbing methods.

2.3 Design Principles for Tree Climbing Robots

The previous sections have ıntroduced natural and artificial tree climbing. This section presents a design principle for making a tree climbing robot. It helps designer to select suitable fastening principle and locomotion. To design a tree climbing robot for specific application, six major design principles should be considered· *maneuverabılıty*, *robustness*, *complexıty*, *adaptıveness*, *sıze*, and *speed*. These principles are correlated at certain levels and trade-offs will need to be made among them.

Maneuverability: As the structure of a tree is complex in terms of geome-

Table 2.2: Summary of the artificial tree climbing methods.

Fastening principle	Maneuver Locomotion	Examples	Advantages	Disadvantages
Claw penetration	Tripod gait	RiSE V2	Stable gripping, wide range of gripping curvature	Slow locomotion
	Diagonal footfall patterns (Quadrupedal)	RiSE V3	Fast locomotion	Not stable
Frictional gripping, Encircle	Wheel-driven	Pruning robot	Fast locomotion	Low maneuverability
	Extend-contract	WOODY, PCR	Stable grip	Gripping range is limited
	Rolling, helical shape	Snake robot	Adapt to different terrains	Complex, multiple actuators

try, reaching a wide range of positions requires high maneuverability. However, simplified artificial locomotion methods always provide limited degrees of freedom and hence restrict the climbing space to straight tree trunks. A locomotion method with higher maneuverability must be devised if the robot is to be applied to a wide range of climbing workspaces. However, mechanical designs with a higher maneuverability are always results in higher complexity.

Robustness: Robustness in this case means the ability to hold on a tree without falling off. It is related to the payload requirement, fastening methods, and locomotion methods, as some locomotion will generate extra dynamic force for which the fastening method cannot compensate. For example, making the center of mass of a robot higher than the branches on which it is to climb will increase the tendency of the robot to topple sideways or pitch backwards [47, 49].

Complexity: Mechanical and control complexity are important considerations. If a robot is mechanically complex, then it is likely to be larger and heavier. Complex controls reduce the robustness of a robot, and failure is more likely to occur.

Adaptability: Arboreal habitats are varied. A fastening method and size should thus be selected that can adapt to a range of expected workspaces (mainly determined by the diameter of branches).

Size: A small size provides many advantages, such as a lower center of mass and the ability to climb along smaller branches and in narrow spaces.

Speed: Fast climbing is preferred in many applications. Nonetheless, a faster speed will always reduce the robustness of a robot.

2.4 Ranking of the Tree Climbing Principles

Table 2.3 ranks the natural and artificial climbing principles reviewed in the previous section based on these six design principles. This will help with the selection of the most appropriate climbing principle in designing a tree-climbing robot. The details of the ranking are explained as follow.

Maneuverability

For the existing extend-contract and wheel-driven methods, only one direction can be achieve, so the ranking is lowest. It is known that the more the DOF of a robot, the higher the maneuverability can be achieved. In order to make a fair comparison, the maneuverability is ranked according the number of actuators required to achieve the same maneuverability. The ranks can be referred to the complexity and hence the ranking of maneuverability in descending order becomes bending, quadrupedal gait, tripod gait and wave form methods.

Robustness

In view of fastening method, robustness means the magnitude of the fastening force. The frictional gripping interlocks with a branch and hence can provide infinite force. It will be pulled out only if the branch is broken by the pulling force. The claw penetration method actually interlocks with the tree bark with certain depth, which will break easier than a whole branch. Hence, the robustness of claw penetration is ranked lower than frictional gripping. As for the wet and dry adhesion method, they adhere on a surface

Table 2.3: Ranking of fastening and locomotion principles in tree climbing (1=worst, 6=best)

Design principles	Adaptability	Robustness	Size	Complexity	Maneuverability	Speed
Fastening method						
Wet adhesion	6	2	6	2	▨	▨
Van der Waals force	6	1	6	1	▨	▨
Claw penetration	4	5	4	4	▨	▨
Frictional gripping	1	6	1	6	▨	▨
Maneuvering method						
Wave form	▨	6	1	1	2	1
Bending	▨	1	4	4	6	4
Extend-contract	▨	3	5	5	1	4
Tripod gait	▨	5	2	2	3	2
Quadruped gait	▨	4	3	3	4	4
Wheel-driven	▨	6	6	6	1	6

of tree only. It will highly affected by the dustiness of the surface. Hence the fastening forces by these methods are lower than the frictional gripping and claw penetration. According to [60], the range of Van der Waals force and capillary force is about 10^{-11} to 10^{-9} and 10^{-7} to 10^{-3} kgf respectively. In experiments, Kim [61] found that a snail the total maximum adhesive force is $3.1mNmm^{-2}$. Jiao [62] found that the adhesive for on insect leg (Van der Waals force) is $1.7-2.2mNmm^{-2}$. As a result, the wet adhesion force should be larger than Van der Waals force.

As for the maneuvering method, robustness means the magnitude of the dynamic force generated by the maneuvering motion. The smaller the dynamic force, the better the robustness can be achieved. Wave form and wheel-driven have the highest rank as these motions does not change the center of mass relative to the fastening position which is most stable. Tripod gait has higher priority of quadrupedal gait because there has three supporting points when maneuvering in tripod gait while there are only two supporting points in quadrupedal gait. As for the extend-contract method, although the contraction motion makes the center of mass closes to the substrate which eliminates the pitch back moment, the extension motion moves the center of mass apart from the fastening position that will increase the side toppling force and pitch back moment. As for the bending motion, the motion of straightening the body faces the same issue of the extension motion in extend-contract method. While when the body bends, it will make the center of mass out of the gripping substrate, it will increase the pitch back moment and side toppling force. As a result, it has the lowest rank.

Complexity

For the fastening method, the control of frictional gripping is simplest as it uses the interlock method. Putting robot limbs to the proper position can achieve the fastening effect. Although the claw penetration method can be fabricated easily, it is complex in control as the claw should put directionally to the substrate to generate optimal force [20]. The wet adhesion method requires special liquid between the object and the substrate [57]. The continuous provision of this consumable substance makes the application not easy to apply. In addition, by using the suction force, it is needed to remove the air between the pad and substrate to reduced internal pressure which is not easy to achieve. As for the Van der Waals force, the micro/nano meter scale biomimetic pad is difficult to make. Although there are a lot of works focusing on it such as [1] [58] [56], it is still not mature enough. The adhesive force will degraded with time due to dust and dirt [59]. So the rank is lowest.

For the maneuvering method, the complexity of maneuvering is defined as the control effort. The wheel-driven method maneuvers by continuous rolling motion. One control command is enough for maneuvering, so the rank is highest. The extend-contract method consists of two motions, i.e., extend and contract. Two motion step is needed which is a little complex than wheel-driven method. The bending method also needs to control the bending in three-dimensional space. It needs six control input to define the position and orientation in three-dimensional space. It is much more complex in control. In quadrupedal gait, it needs to control four limbs. In addition, the motion of the limbs must be synchronization when moving. In tripod gait, the situation is similar to quadrupedal gait, but it needs to control six limbs. As for the wave form generation, it is achieved by millions of muscles in snail. Hence it ranks the lowest.

Adaptability

Only the frictional gripping requires over 180 degrees gripping. Hence the rank is lowest. Claw gripping is only suitable for convex surface while wet adhesion and Van der Waals force can adhere on any shape of substrate. Hence the ranking of wet adhesion and Van der Waals force is the same and higher than claw gripping method.

Size

As for the fastening method, the size of robot using frictional gripping is similar to the size of gripping substrate, while by using the claw penetration, under half of the size can be achieve. As for the wet and dry adhesion, they are simply independent to the size of gripping substrate. As a result, the ranking of size for fastening method is same as the adaptability for the fastening method.

The size of the maneuvering methods is related to the complexity of the mechanism. It can be quantified by the number of actuators required. The simpler the mechanical structure the more small size robot can be made. A wheel-driven method is the simplest mechanism as it maneuvers by rolling. One control command is enough, so the rank is highest. The extend-contract method is a one DOF motion which can be achieved by single actuator. As a result, the rank is same as wheel-driven method. The bending method needs to control the bending in three-dimensional space. It at least needs six

DOF to define the position and orientation in three-dimensional space. In Quadrupedal gait, even each limb only has two DOF (which can be more in actual case), eight actuators required in total. The same concept for tripod gait, twelve actuators are required. As for the wave form generation, it is achieved by millions of muscles in snail. Hence it ranks the lowest.

Speed

The speed of the maneuvering method is compared by given a same size of robot, how far the robot can move forward by one climbing gait. Since wheel-driven method do not have the concept of climbing gait, it has highest rank. In each climbing gait, the bending, extend-contract and quadruped method can move same as the robot length while tripod method can only move forward smaller than half of the robot length. As for the wave form method, it even moves much smaller than half of the robot length.

□ **End of chapter.**

Chapter 3

Development of a Novel Tree Climbing Robot

3.1 Approach to the Robot Design

Table 2.3 can be used as a reference to design the tree climbing robot. Referring to Chapter 1, the design principles needed to be considered with priority included maneuverability, adaptability, robustness, and size.

In terms of fastening method, only claw penetration method has high rankings in all considered design principles. As a result, this method is adopted in the robot design

In terms of locomotion, the table shows that bending methods has the best performance in maneuverability but the robustness is the worst. In order to improve the robustness, a novel form of locomotion that combines the bending and extend-contract locomotion is proposed. It is accomplished by using an novel extendable continuum manipulator, which combines the advantage of the high maneuverability of bending locomotion and the robustness of the artificial extend-contract locomotion.

3.2 Structure of the Tree Climbing Robot

Fig. 3.1 shows the overall structure of the proposed tree climbing robot "Treebot". Treebot is composed of three main elements: a *tree gripper*, a *continuum body* and a *semi-passive joint*. Two grippers connect to the ends of the continuum body respectively, and the semi-passive joint is installed between the body and the front gripper. Several *sensors* are also installed for autonomous climbing.

3.2.1 Tree Gripper

The proposed gripper is designed to fasten onto a wide variety of trees with a wide range of sizes. There are many innovative approaches for generating adhesive force, such as vacuum suction [4, 37–39], magnetic attraction [5, 6, 18, 35, 36], elastomeric adhesive [7, 41], electroadhesion [2] and fibrillar adhesion [1, 3]. These methods work well on manmade structures such as vertical walls and glass windows that are smooth and flat. However, they are not applicable to tree surfaces, which are completely different in nature. Observation of the arboreal animals indicates that the claw gripping method is reliable on tree surfaces, and this method is thus adopted to provide the fastening force for the gripper.

The gripper is composed of four claws equally separated by 90 degrees. Fig. 3.2 shows the overall design of the gripper in different views. The grip-

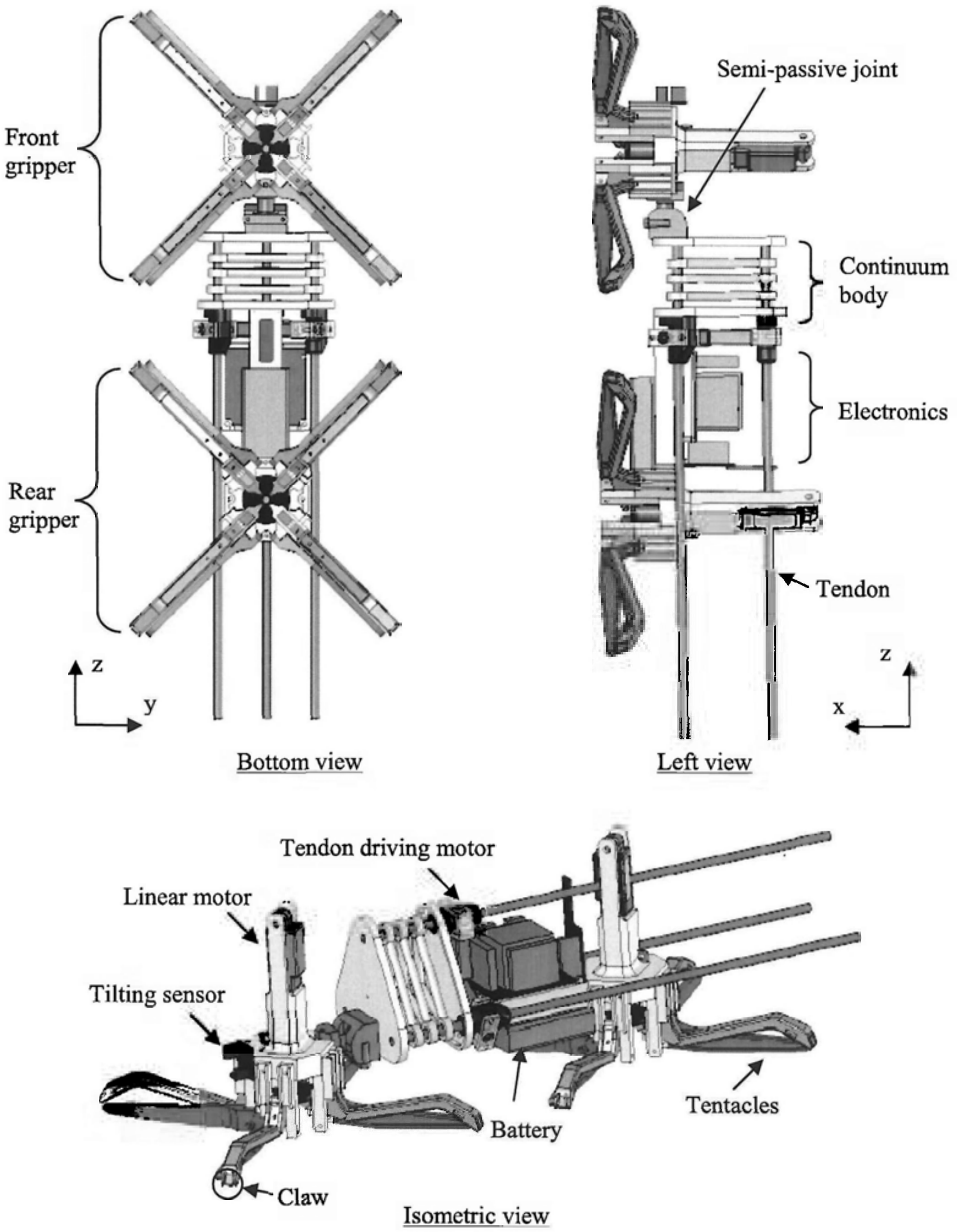


Figure 3.1: Structure of Treebot.

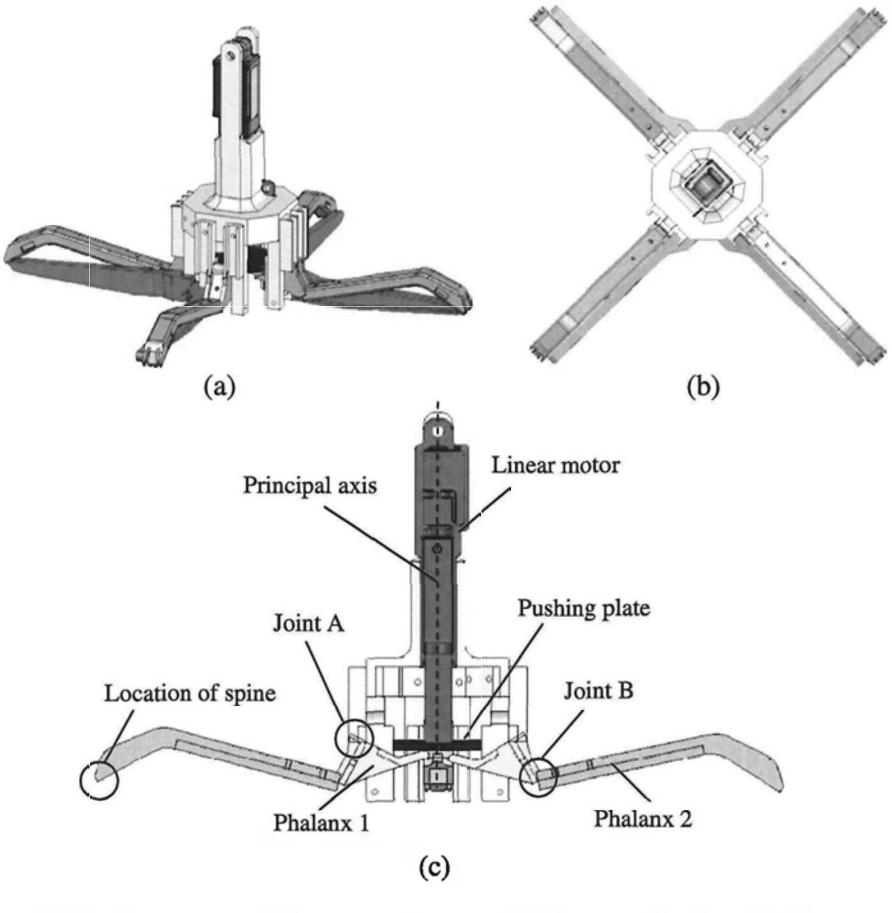


Figure 3.2: Structure of the tree gripper. (a) Isometric view; (b) Top view; (c) Cross-sectional view.

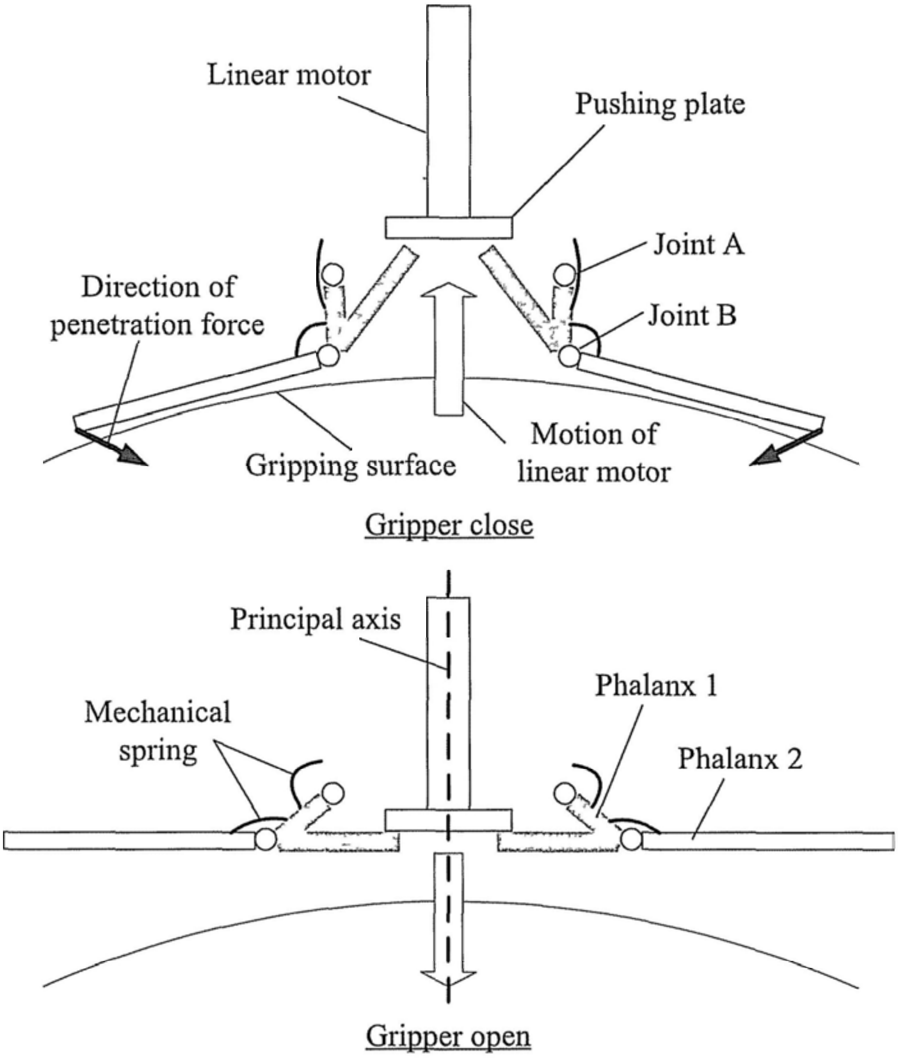


Figure 3.3: Illustration of the mechanism of the tree gripper.

per should be appressed to the tree surface (the center of the gripper makes contact with the gripping surface and the centerline of the gripper is collinear with the surface) to generate maximal fastening force. Each claw is composed of two parts (Phalanx 1 and 2) and has surgical suture needles installed at the tips. The claws adopt a two-bar linkages mechanism to generate optimal direction of acting force. Each gripper has one linear motor that actuates all four claws. A pushing plate is mounted at the end of the linear motor. When the linear motor extends, the plate pushes all the phalanges 1 downward and hence makes the phalanges 2 upward. The spring on joint A is further compressed and the spring on joint B is released at the same time. This motion pulls spines off the gripping substrate. When the linear motor contracts, the compressed springs on the joints generate a force that pushes the claws back into the gripping substrate and at the same time the spring on joint B will further be compressed. Fig. 3.3 illustrates the gripping mechanism. As the gripping force is generated by the compressed springs only, static gripping can be achieved with zero energy consumption. A constant force spring (a flat spiral spring) is adopted to ensure that the force is independent of the claw traveling angle. In addition, as the mechanism of each claw is independent, the claws can travel in different angles. This ensures that all of the claws penetrate into the gripping substrate to generate the maximal force even if it has an irregular shape.

Tentacles are installed beside each claw. These have various functions, including acting as tactile sensors, helping to ensure that the gripper is appressed to the tree surface, and helping the claws to retract from the gripping substrate. The gripper allows omni-directional gripping about its principal

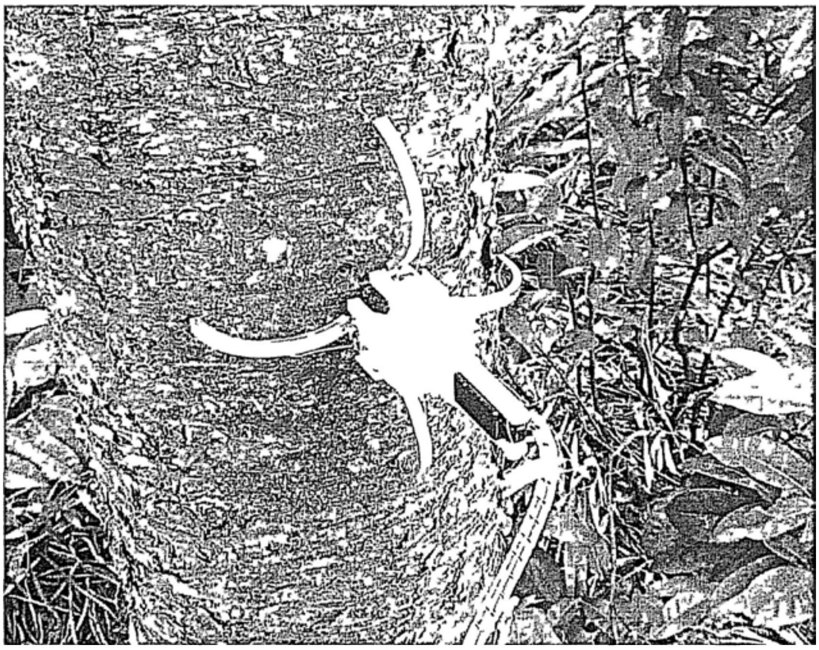


Figure 3 4 Prototype of the tree gripper

axis so that no additional orientation actuator or control is needed. Additionally, the gripper is actuated by only one motor, which makes it light, compact, and easy to control

Fig. 3.4 shows the prototype of the proposed tree gripper The gripper is 130mm (height) × 160mm (width) × 160mm (Length) in size and weighs 120grams It can be observed from the figure that the claws can travel in different angles to adapt curved surfaces

3.2.2 Continuum Body

There are many types of continuum manipulators that utilize wire-driving or pneumatic-driving mechanism [21–23, 25, 26], as shown in Fig 3 5 Most of them are able to bend in any direction and some of the pneumatic-driving manipulators are even able to extend to a certain extent. Most of the researches use the continuum structure as robot arms, but seldom researchers have realized that it can also be applied to maneuvering. The continuum mechanism is a compliant structure, as it does not contain fixed joints [24]. Its inherent passive compliance is of particular benefit for maneuvering in an irregular arboreal environment, as it often eliminate the need for complex force sensing and feedback control [19]. For climbing purposes, the manipulator should be compact and lightweight. There are many types of continuum manipulator, but none of them fulfills all of these requirements. Existing continuum manipulators need to connect to large external boxes that contain wire, drivers, motors, or air pumps. Although some pure wire-driven continuum manipulators [25, 26] have the potential to be more compact and lightweight, they are not extendable. Extendibility is important, as Walker [27] shows that the inclusion of extension ability for continuum manipulators highly extends the workspace.

Due to the limitations, a novel design of continuum manipulator with both bendable and extendable functions is proposed as a robot's body. The continuum body is a type of single section continuum manipulator [17] with a novel mechanism. It has high degrees of freedom (DOFs) and a superior ability to extend that existing designs cannot achieve. This allows the robot to maneuver on irregularly shaped trees. The continuum body has three DOFs that can extend and bend in any direction. In Treebot, it acts as a maneuvering mechanism to place one end of the gripper on a target position, allowing the robot to reach many places on a tree. The locomotion of Treebot is similar to that of an inchworm robot [18], except that the moving motion

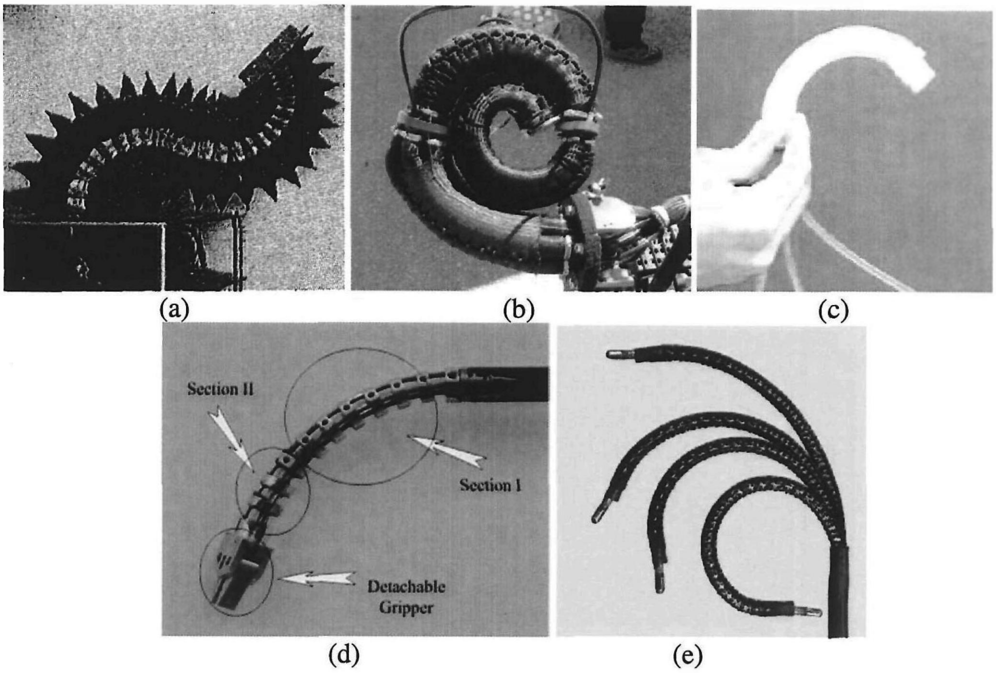


Figure 3.5: Different types of continuum manipulator. (a) Immega 1995 [21]; (b) McMahan 2006 [22]; (c) Chen 2006 [23]; (d) Xu 2006 [25]; (e) David 2008 [26].

is achieved by body extension and contraction, rather than body bending. As the extension and contraction maneuvering mechanisms place the center of mass of Treebot close to the climbing surface, a smaller pitch-back moment is produced when climbing. The inherent passive compliance of the continuum body allows it to be sheared in 2-DOF along x and y axes and be twisted about z axis by external force. The definition of the coordinate system is shown in Fig. 3.1. The compliance is useful for adapting to the irregular shape of trees, as it eliminates the need for complex force sensing and feedback control [19]. The proposed continuum manipulator is a self-contained module with integrated actuators, and thus no external control box is required. This makes the manipulator compact and lightweight.

Fig. 3.6 shows a CAD model of the proposed continuum body. It is composed of three mechanical springs that are connected in parallel. The distances between the center of the continuum manipulator and the springs are equal, and the springs are equally separated by 120 degrees. One end of each spring is fixed on a plate, and the other end has no fixed connection. The springs pass through a plate that contains three DC motors to control the length of the springs between the two plates independently. By controlling the length of each spring, the continuum manipulator can perform bending and extension motions. Commonly, the number of actuators required for each section of the continuum manipulator is more than the number of admissible degrees of freedom. However, in the proposed structure, only three actuators are used but three DOFs are provided. This structure provides the maximum DOF with the minimum number of actuators. Fig. 3.7 shows the prototype and illustrates the admissible motion of the continuum body.

The actuation mechanism is similar to a rack and pinion mechanism, which allows the unlimited extension of the continuum manipulator theoretically. In practice, extension is limited by the length of the springs. Each spring can be treated as a bendable rack, and is only allowed to bend in any direction but not to compress or extend to keep a constant gap distance within which the pinion can drive. Maintaining the springs at constant intervals across the manipulator is important to retain a uniform shape, and several passive spacers are installed in the middle of the manipulator for this purpose. The maximum distances between the spacers are constrained by wires.

The inherent passive compliance also allows the continuum manipulator to be sheared along the x- and y-axis and twisted about z-axis by external force, as shown in Fig. 3.8. This passive compliance results from the bendable characteristics of the mechanical springs. The amount of compliance increases when the continuum manipulator extends.

3.2.3 Semi-passive Joint

To appress the gripper to a gripping surface, the gripper should have a certain turning ability about the y-axis and z-axis. However, the inherent compliance of the continuum body does not include rotational motion about the y-axis and only affords a limited twisting angle about the z-axis. An additional device is thus needed to provide enough degrees of freedom. To provide the requisite degrees of freedom and maintain the light weight of the robot, a semi-passive joint is developed that comprises a passive revolute joint with two DOFs. A passive joint is used rather than an active joint to eliminate the need for complex controls to orient the joints and to reduce the number

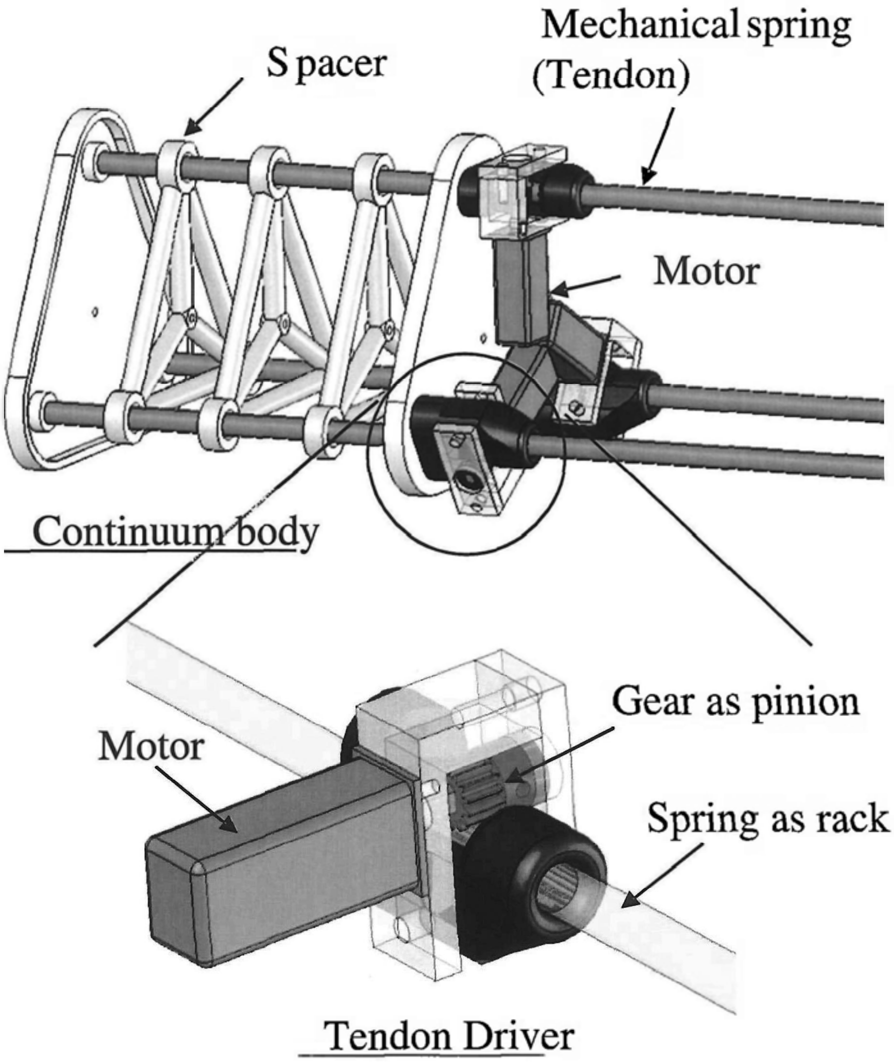


Figure 3.6: Design of the proposed continuum manipulator.

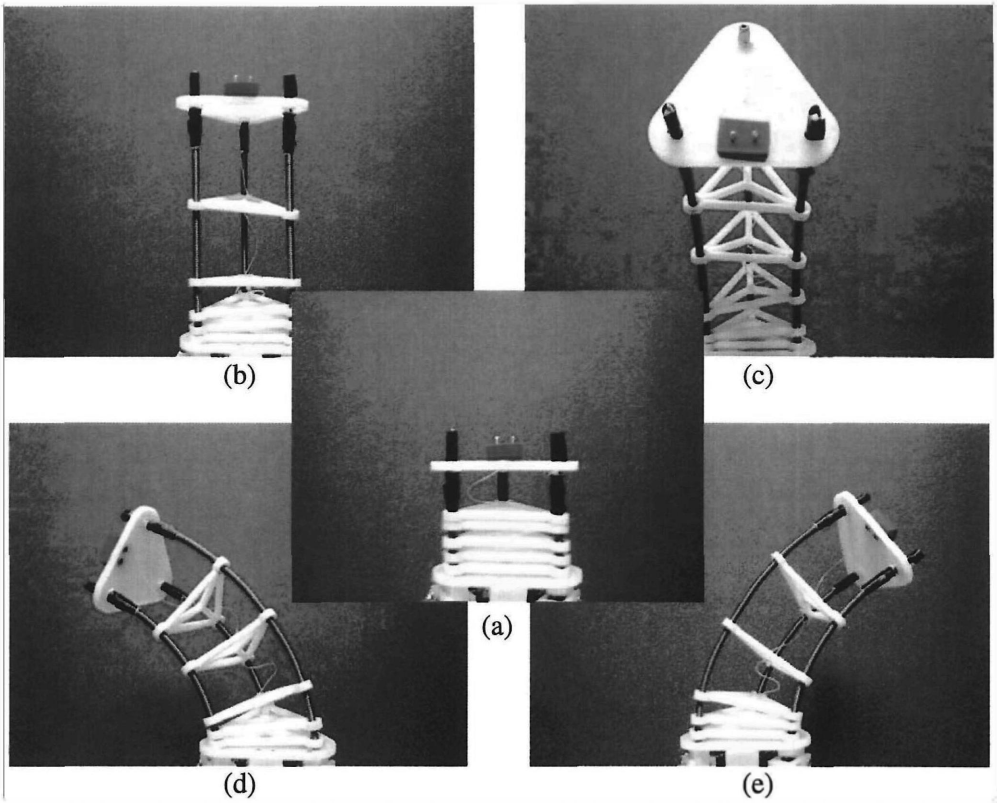


Figure 3.7: Prototype of the continuum body and illustration of the degrees of freedom: (a) contraction; (b) extension; (c) forward bending; (d) right bending; (e) left bending.

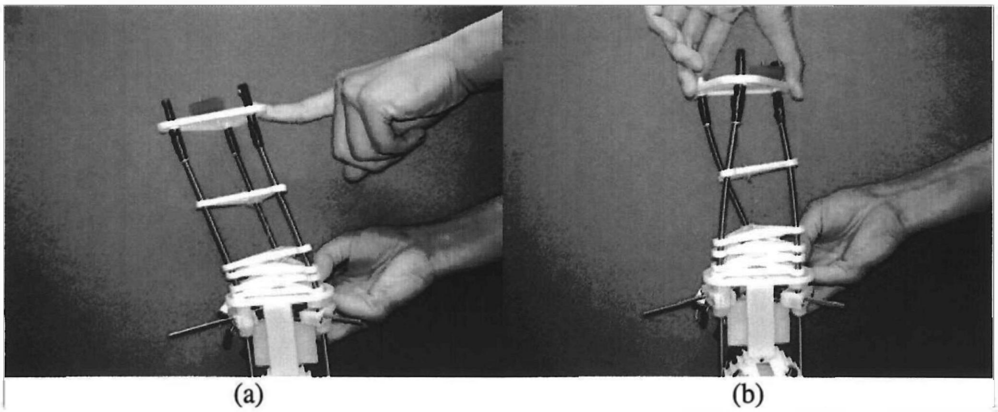


Figure 3.8: Compliance of the continuum body: (a) shearing and (b) twisting.

of actuators, thereby keeping Treebot lightweight. The joint is installed between the front gripper and the continuum body, and can actively lock and unlock. To reduce the number of actuators required, the lock/unlock action is controlled by a linear motor that controls the gripping motion of the front gripper. When the joint is unlocked, the joint can be rotated about the y- and z-axes. When the joint is locked, it actively returns to the initial orientation in which rotation about the y- and z-axes is zero. The locking mechanism is needed to fix the front gripper for exploring purpose.

The semi-passive joint is composed of three parts, Part A, B, and C, as illustrated in Fig. 3.9. Part A is connected to the gripper, Part B is connected to the continuum body, and Part C keeps Part A inside Part B. It can be observed that the joint can freely be rotated in the y- and z-axes only. The range of the twisting angle on the y-axis is ±45 degrees. There is no mechanical constraint on the twisting angle for the z-axis. However, if the joint turns over ±60 degrees, the locking system is unable to force the joint to return to the initial orientation. As a result, the angle of twist about the z-axis must be constrained to ±60 degrees.

At the contact surfaces of Part A and B, there is a convex and a concave triangle cone respectively. The convex triangle cone fits inside the concave triangle cone, and together the cones work as a locking system to return the joint to the initial orientation. The locking mechanism is illustrated in Fig. 3.10. A wire passes through the center of Parts A and B along the z-axis. One end of the wire is fixed on the continuum body, and the other end is connected to the linear motor at the front gripper. The semi-passive joint shares an actuator with the front gripper. When the linear motor extends,

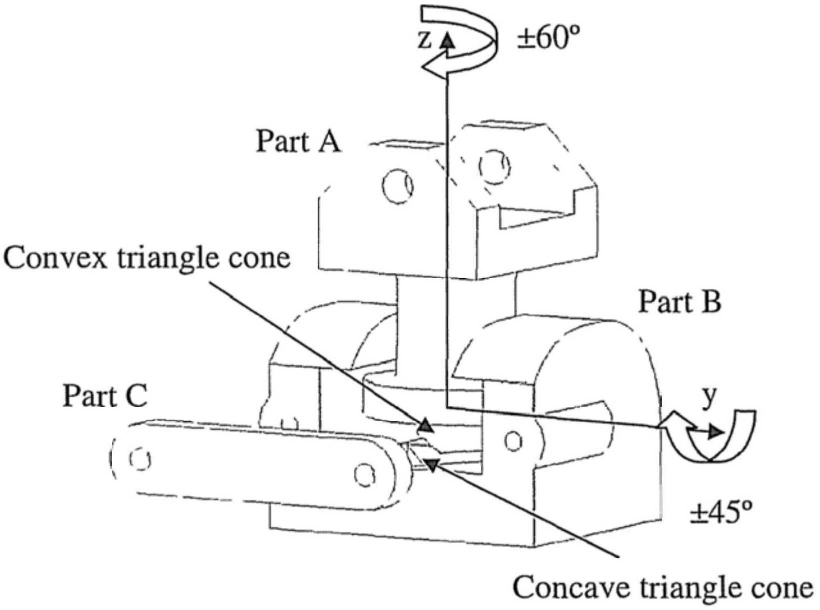


Figure 3.9: Design of the semi-passive joint.

the wire pulls Part B close to Part A. The convex and concave triangle cones between the two parts then push together and force the joint to return to the initial orientation.

Fig. 3.11 shows the hardware prototype of the semi-passive joint in an unlocked state. It can be observed that the joint can be twisted about the z-axis. Fig. 3.12 also illustrates the gripper in unlocked and locked state. In Fig. 3.12(a), the front gripper is closed. The semi-passive joint is unlocked, and the front gripper is not in the initial orientation. When the front gripper is opened, the semi-passive joint is locked (Fig. 3.12(b)) and the front gripper returned to the initial orientation.

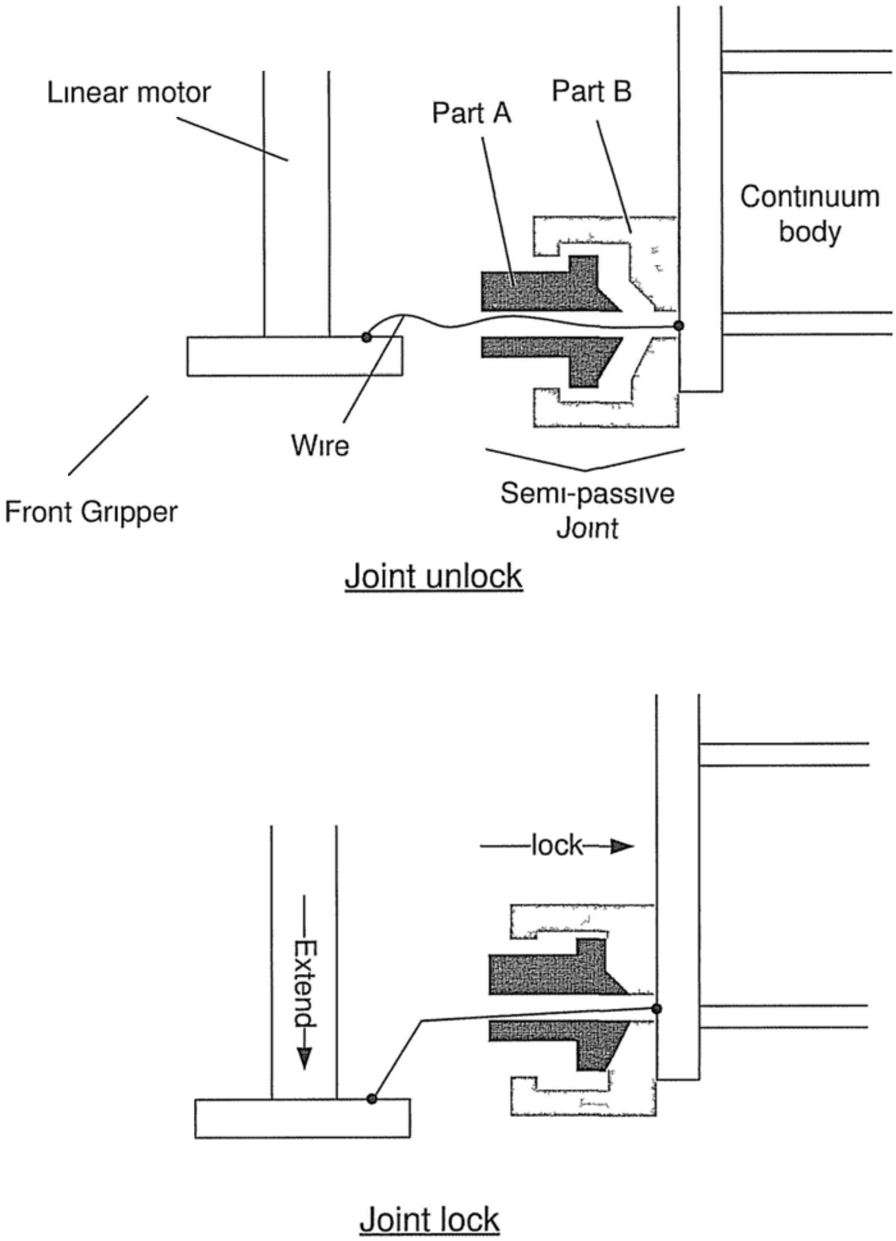


Figure 3 10 Locking mechanism of the semi-passive joint

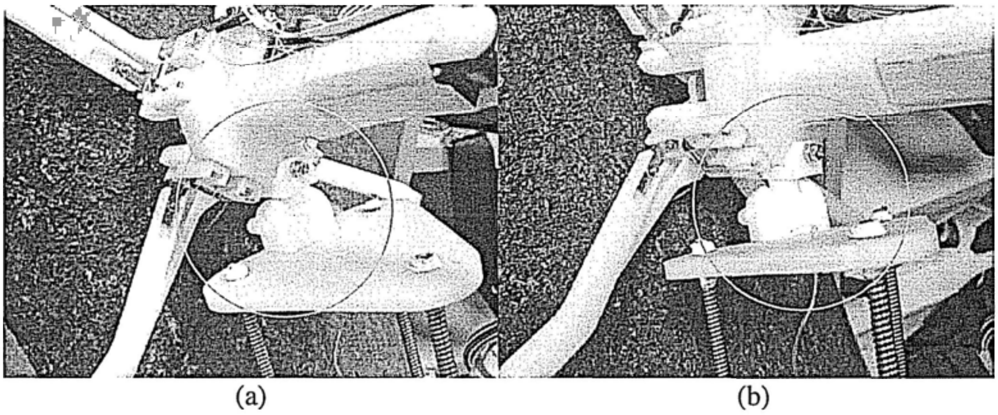


Figure 3.11: Hardware prototype and the installation position of the semi-passive joint. (a) Initial orientation; (b) Twisted about the z-axis.

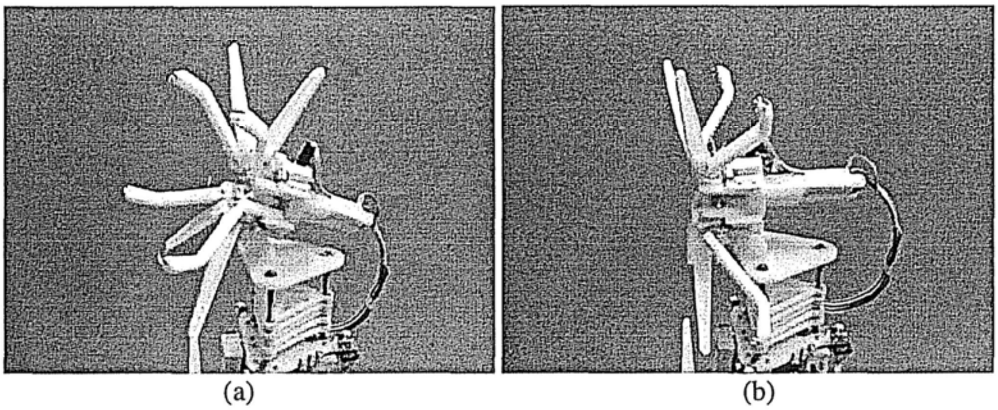


Figure 3.12: Semi-passive joint: (a) unlocked; (b) locked.

3.2.4 Sensors

To realize the motions of Treebot and allow explore the environment, three types of sensors are installed: encoders, tactile sensors, and tilting sensors. Encoders are installed on each tendon of the driving motor to measure the posture of the continuum body. Four tactile sensors (colored blue in Fig. 3.1) are installed on each gripper to detect the interaction between the gripper and the climbing surface. A triple-axis tilting sensor is also attached to the front gripper to measure the direction of gravity.

In summary, Treebot has three active DOFs, i.e., the continuum body and the two passive DOFs, i.e., in the semi-passive joint, which is a kind of underactuated robot. On the other hand, only five actuators are used, two for the motion of the grippers and three for the motion of the continuum body.

3.3 Locomotion

The locomotion of Treebot is similar to inchworms which is a kind of bipedal locomotion. Fig. 3.13 shows a complete climbing gait of the locomotion. It is composed of six climbing steps. The square colored in grey represents the closed gripper that attached on the substrate while the square colored in white represents the opened gripper that detached on the substrate. The order of motion in the figure represents the locomotion of moving forward. The locomotion of moving backward is just in reverse order.

Treebot is able to change a moving direction in three-dimensional space by bending the continuum manipulator. It allows Treebot to climb along a

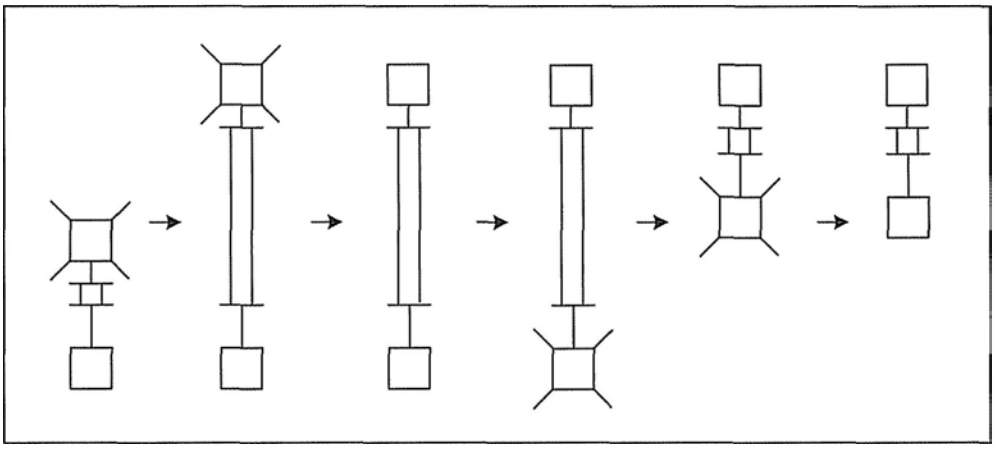


Figure 3.13: A complete climbing gait of Treebot (moving forward).

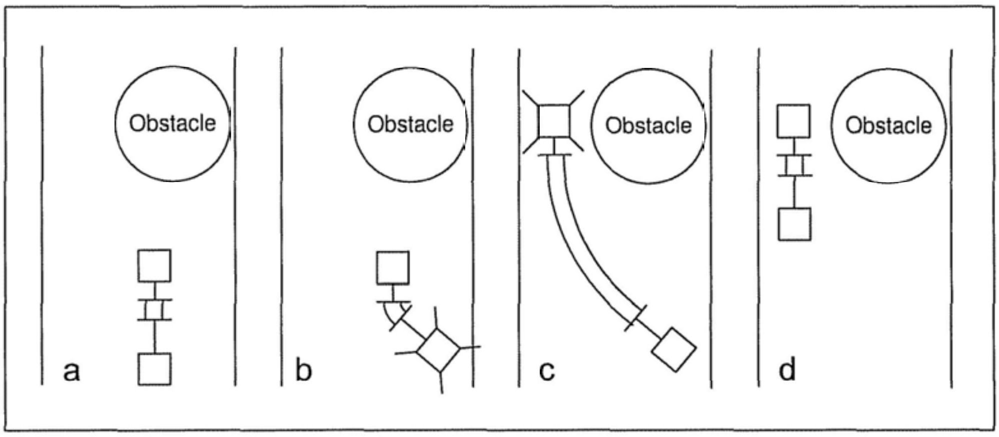


Figure 3.14: Motions to avoid an obstacle.

curved shape of tree or avoid obstacles such as non-passing through branches. This ability makes Treebot has high maneuverability that surpass the existing tree climbing robots. Fig. 3.14 shows part of climbing motions to avoid an obstacle on a tree. Treebot can first adjust the direction of the rear gripper and then climb along this direction to avoid the obstacle. This method is also applicable for turning to another side on a branch or selecting a branch to climb.

3.4 Hardware Prototype

Fig. 3.15 and Fig. 3.16 show the prototypes of Treebot. The first prototype named Treebot is a remote-control robot with no sensor installed. It is used to verify the mechanical design by human control. The components are mainly made of polyoxymethylene plastic and aluminum alloy to keep the robot lightweight, and the springs on the continuum body are made of steel. The prototype weighs only 600 grams, which is very light compared with other types of tree climbing robot. It is able to extend by a maximum of 330 mm, and has a climbing speed of 22.4 cm/min.

The second prototype, Treebot-Auto, is designed to implement the autonomous climbing strategy, and is equipped with several sensors. Several performances have been improved when compared with the first prototype. The weight of Treebot-Auto, including the battery, is 650 grams, which is slightly heavier than the previous version. The detailed specifications of the prototypes are summarized in Table 3.1.

3.5 Energy Consumption

One of the excellent features of Treebot is that it consumes little energy. It can even consume zero energy when it holds on a tree. This is accomplished by the special design of the gripper and the self-locking characteristic of the actuators. The actuators can attach on a position without consuming any energy as the gripping force is generated by the preload force of the springs in the gripper.

This feature make Treebot suitable for lengthy work on trees, such as

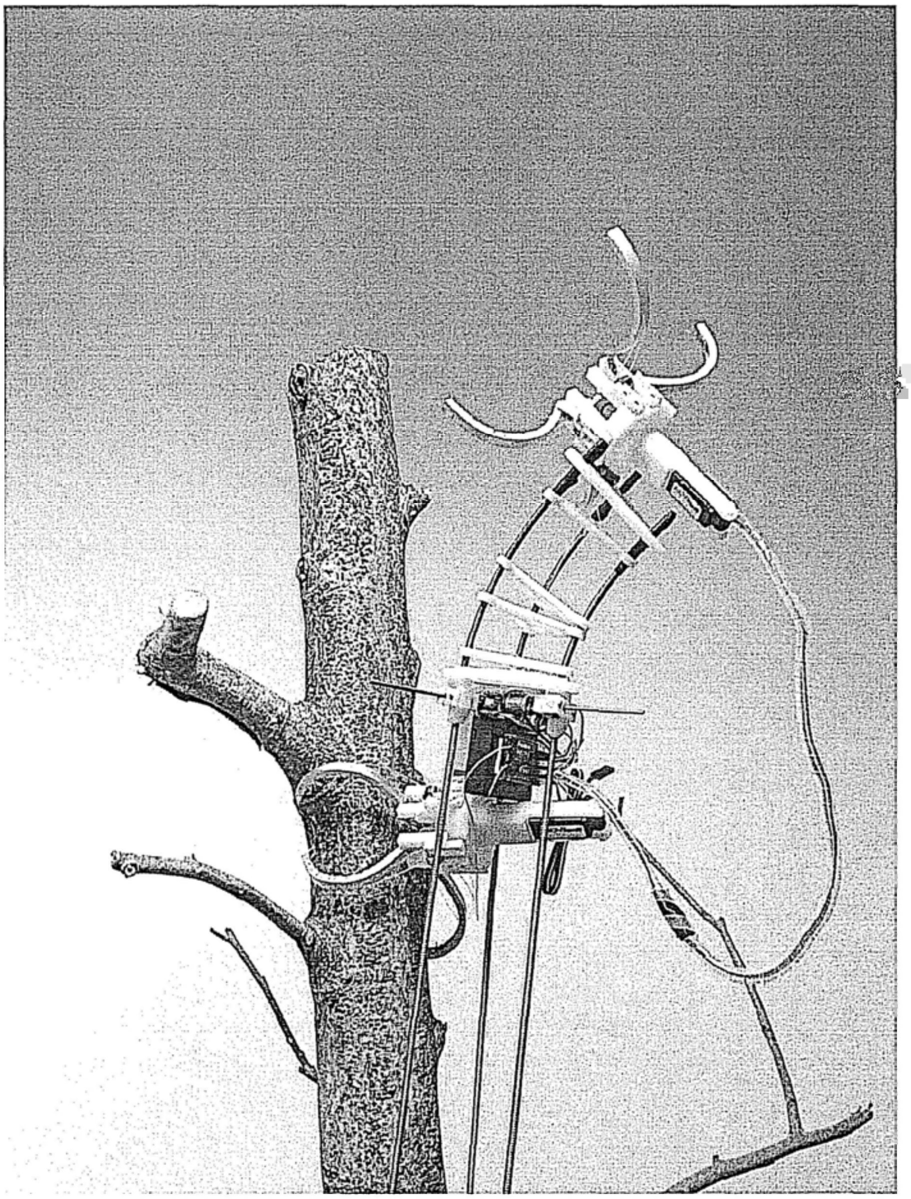


Figure 3.15: First prototype - Treebot.

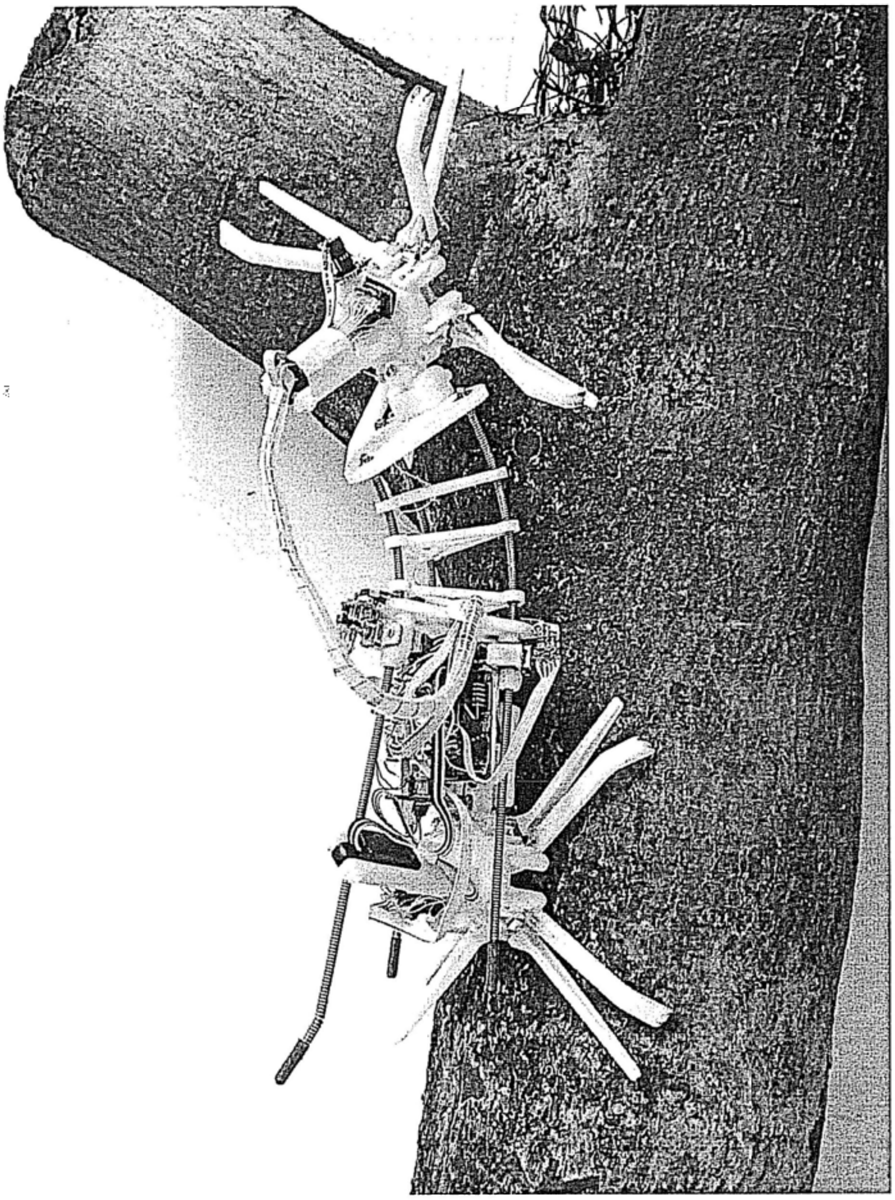


Figure 3.16: Second prototype - Treebot-Auto.

Table 3.1: Specifications of Treebot

Parameters	Treebot	Treebot-Auto
Weight	600grams	650grams
Height	135mm	135mm
Width	175mm	175mm
Length (minimum)	325mm	370mm
Length of extension	340mm	220mm
Maximal bending curvature	$1/30mm^{-1}$	$1/30mm^{-1}$
Power source	NiMh 4.8V 1000mAh	LiPo 7.4V 800mAh
Runtime	2.5 hours	3 hours
Maximal climb-up speed	26.4cm/min	73.3cm/min
Maximal inclined angle	105°	105°
Adaptable tree diameter	64-452mm	64-452mm

surveillance. Treebot also consumes little energy when climbing, and can climb continuously for almost 3 hours on a LiPo two-cell 7.4V/800mAh battery, which weighs about 45 grams.

3.6 Accessories

Treebot is able to be equipped with several accessories, such as a camera and a photovoltaic module as shown in Fig. 3.17 and Fig. 3.18 to enhance its functionality.

The camera can be used to inspect the tree surface or for surveillance. It is fixed on the front gripper. The orientation of the camera can be controlled by the continuum body, and hence additional actuators are not required.

A photovoltaic module can be equipped if Treebot is needed to work on a tree for several days or weeks. As Treebot is designed to work outdoors,

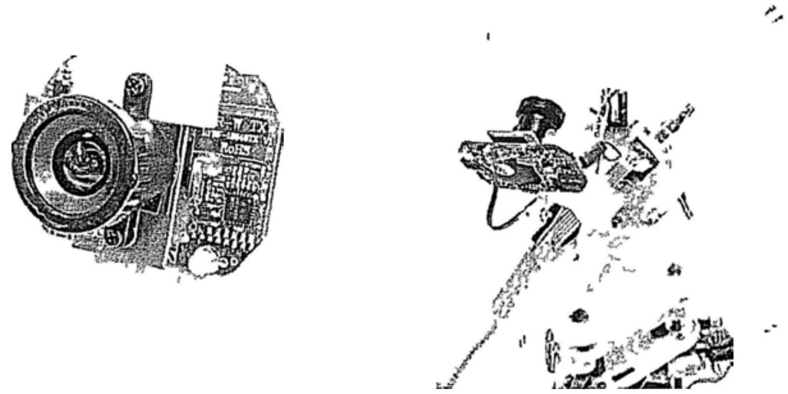


Figure 3.17: Camera module.

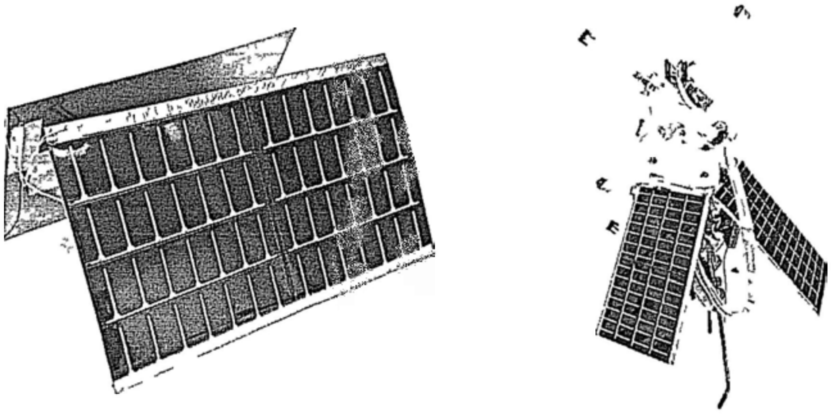


Figure 3.18: Photovoltaic module.

renewable solar energy is the best option for providing unlimited energy. This will eliminate the need to replace the battery and enable Treebot to remain and work on a tree independently.

Both accessories are lightweight. The wireless camera weighs only 12 grams, and the photovoltaic module just 10.9 grams. Hence, neither greatly affects the climbing performance.

3.7 Control

3.7.1 Control Architecture

The control architecture is divided into two parts: the embedded microcontroller in the robot itself and the ground station. Fig. 3.19 illustrates the control architecture and the functions of each part, and Fig. 3.20 shows the interface of the ground station. The control mechanism has a master-slave architecture. The ground station works as the master, making decisions and monitoring the state of Treebot. A video signal from Treebot can be displayed on the ground station. The user can also control Treebot manually through the ground station. The embedded microcontroller works as a slave to implement the motions commanded by the ground station.

In robots, decision-making involves complex algorithms that require a high computational power. If the computational work was performed by Treebot itself, then the robot would need a large processor and would consume much more energy, which would make it heavier. Decision-making is thus carried out on the ground station to allow the robot to be lighter.

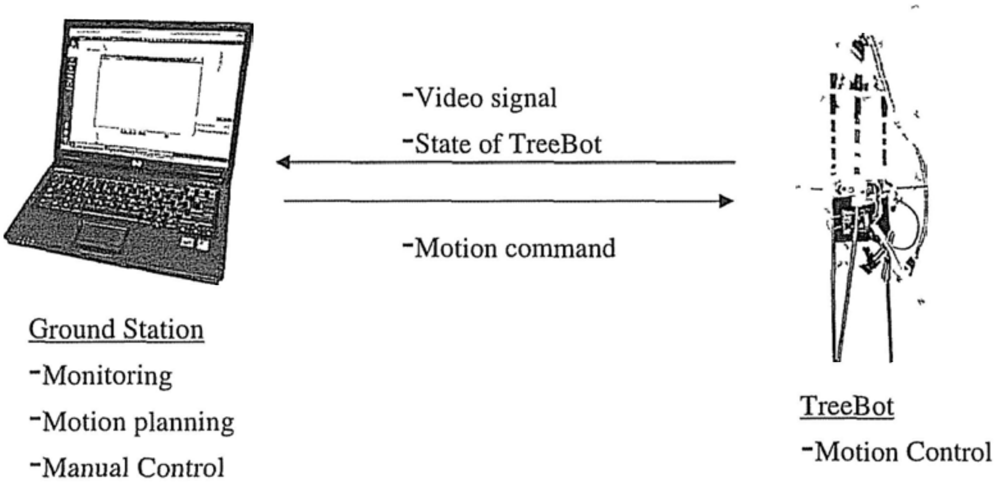
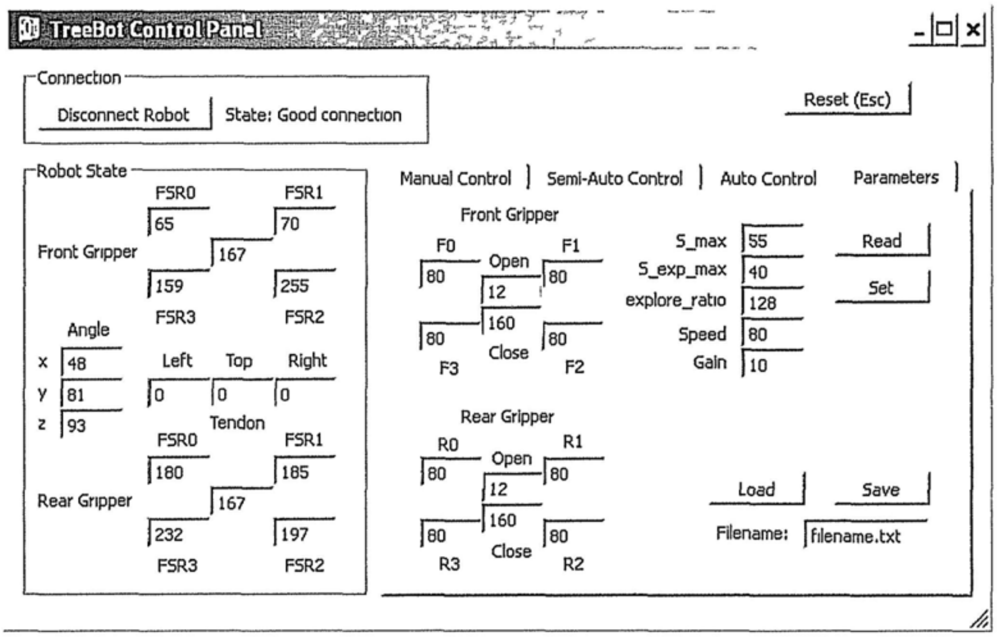


Figure 3.19: Control architecture of Treebot.



State: Good connection

Robot State

Front Gripper: FSR0 65, FSR1 70, 167, FSR3 159, FSR2 255

Angle: x 48, y 81, z 93

Tendon: Left 0, Top 0, Right 0

Rear Gripper: FSR0 180, FSR1 185, 167, FSR3 232, FSR2 197

Front Gripper: F0 80, F1 80, Open 12, Close 160, F3 80, F2 80

S_max 55
S_exp_max 40
explore_ratio 128
Speed 80
Gain 10

Rear Gripper: R0 80, R1 80, Open 12, Close 160, R3 80, R2 80

Filename: filename.txt

Figure 3.20: Interface of the ground station.

3.7.2 Manual Control

Treebot can be a remote control robot. The control input of the gripper is simply an on/off command to make grippers fully open or close. As for the control of the continuum manipulator, since it has three DOF, three channels of input are needed. One way is to directly input the length of each spring. However, it is not an intuitive way for human manipulation. Human being always has a perspective of the direction of motion when controlling something, i.e., the concept of left, right, front and back. As a result, to make an intuitive controller, we define three control inputs, i.e., S_{input}, κ_{input}^{FB} and κ_{input}^{LR}. S_{input} controls the length of the continuum body, κ_{input}^{FB} controls the magnitude of front and back bending while κ_{input}^{LR} controls the magnitude of left and right bending. The concept of front is defined as the direction of positive x-axis while the concept of left is defined as the direction of positive y-axis. The mapping from the control inputs to the posture of the continuum body are:

$$\begin{bmatrix} S \\ \kappa \\ \phi \end{bmatrix} = \begin{bmatrix} S_{input} \\ \min\left(\sqrt{{\kappa_{input}^{FB}}^2 + {\kappa_{input}^{LR}}^2}, \kappa_{\max}\right) \\ \operatorname{atan2}\left(\kappa_{input}^{LR}, \kappa_{input}^{FB}\right) \end{bmatrix} \tag{3.1}$$

where $S_{input} \in [0, S_{\max}]$ and $\kappa_{input}^{FB}, \kappa_{input}^{LR} \in [-\kappa_{\max}, \kappa_{\max}]$. S, κ and ϕ are the parameters to describe the posture of the continuum body that defined in Chapter 5. Once the posture of the continuum manipulator is determined, the length of each spring can be found by (5.1).

3.8 Experiments and Results

Numerous experiments have been conducted to evaluate the performance of Treebot in various aspects, including *generality*, *transition motion*, *turning motion*, *slope climbing* and *payload*.

3.8.1 Generality

Tree climbing tests were carried out on thirteen species of trees. The tree species, bark texture, diameter, and the number of trials, and number of successful climbing steps are summarized in Table 3.2. A climbing step represents a pair of climbing motions, i.e., the front gripper grips with the continuum manipulator extended and the rear gripper grips with the continuum manipulator contracted. The results show that Treebot performs well on a wide variety of trees. It can be realized that the range of successful climbing diameter of tree is wide, from 20cm to 142cm. However, Treebot did fail to climb several species of trees, including Melaleuca quinquenervia, Cinnamomum camphora, and Bambusa vulgaris var. Striata. Treebot failed to climb on Bambusa vulgaris var. Striata because the surface of this tree is very hard and difficult for the spine on the gripper to penetrate. Melaleuca quinquenervia and Cinnamomum camphora both have peeling bark, and although the gripper could grip them, Treebot fell off as the bark peeled away. These experimental results indicate that Treebot performs well on trees with surfaces that are not particularly hard and on non-peeling bark.

Table 3.2: Climbing performance on different species of trees

Tree	Bark texture	Diameter (mm)	No. of steps (Success/Total)
Bombax malabaricum	Rough	452	20/20
Callistemon viminalis	Ridged and furrowed	315	20/20
Delonix regia	Smooth	309	20/20
Bauhinia blakeana	Smooth	80, 207	20/20
Bauhinia variegate	Smooth	258	20/20
Roystonea regia	Smooth, shallowly fissured	352	20/20
Acacia confuse	Smooth	229	20/20
Grevillea robusta	Scaly	159	20/20
Bambusa ventricosa	Smooth	64, 95	20/20
Araucaria heterophylla	Banded	277	20/20
Cinnamomum camphora	Ridged and furrowed, exfoliating	210, 293	13/20
Bambusa vulgaris var. Striata	Smooth, hard	99	1/5
Melaleuca quinquenervia	Sheeting, soft, exfoliating	446	0/5

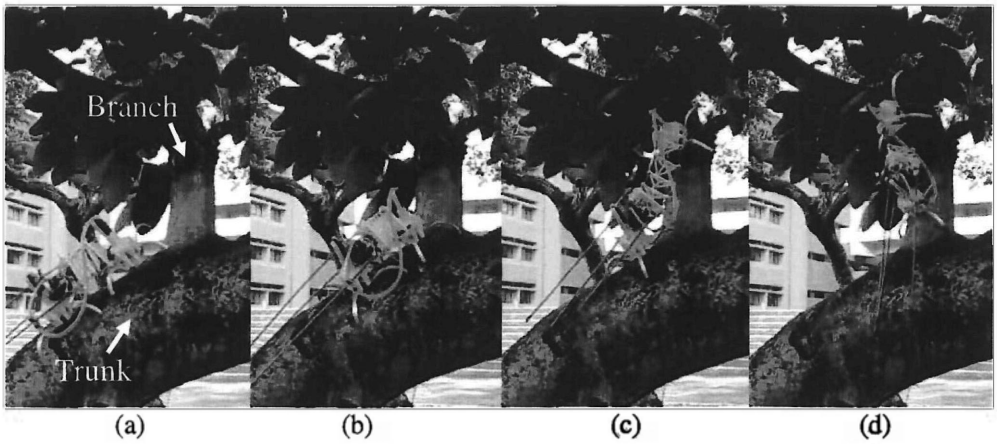


Figure 3.21: Branch transition on a Bauhinia blakeana.

3.8.2 Transition Motion

To verify the maneuverability of Treebot, its transition motion from trunk to a branch was tested on a Bauhinia blakeana. The diameter of the trunk was 280mm and the slope was about 45 degrees. The diameter of the target gripping branch was 118mm and the slope was about 90 degrees. Part of the transition motions is depicted in Fig. 3.21. It shows that Treebot successfully left the trunk and completely climbed on the branch. This transition motion takes three climbing steps under three minutes.

3.8.3 Turning Motion

A turning motion has also been performed to evaluate the maneuverability of Treebot. The experiment was carried out on a Bauhinia blakeana trunk with a diameter of 207mm. Fig. 3.22 depicts part of the turning motions. It can be seen that Treebot successfully moved from the front to the back of the trunk. This motion took five climbing steps and around five minutes.

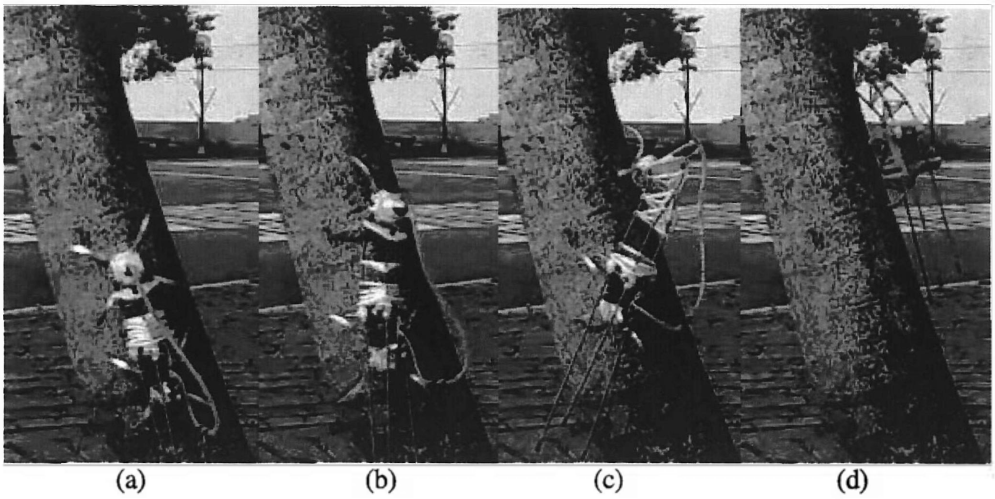


Figure 3.22: Turning motion on a Bauhinia blakeana.

The compliance of the gripper resulted in successful appression to the tree surface (Fig. 3.22(b), (c) and (d)), allowing Treebot to perform the turning motion successfully.

3.8.4 Slope Climbing

The limitation of the slope that Treebot can climb is 105 degrees. Details of this limit can be found in Chapter 5. An experiment was conducted to examine the maximum climbing slope of Treebot. It has been implemented on a Bauhinia blakeana with a diameter of 172mm. The climbing angle was about 103 degrees. Part of the climbing motions is depicted in Fig. 3.23, which shows that Treebot was able to climb the slope successfully. There was no over slope climbing effect appeared in the experiment.

It was also attempted to make Treebot climb a tree with a slope larger than its climbing limit. As shown in Fig. 3.24, Treebot tried to climb a Bauhinia blakeana with a diameter of 207mm and a climbing angle of about

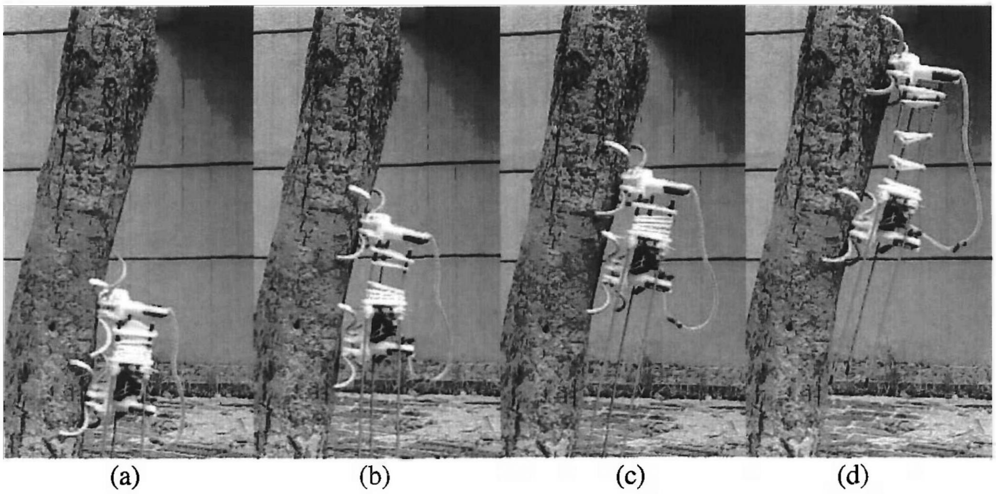


Figure 3.23: 103 degrees slope climbing.

110 degrees. It can be noticed in Fig. 3.24(b) and (d) that an over-slope climbing effect occurred. This position cannot be adjusted as the compliance of Treebot as insufficiently to compensate for the outward angle. As a result, Treebot cannot climb up further.

3.8.5 Payload

As Treebot is designed to carry equipments up a tree, a payload test was implemented to realize the maximum payload of Treebot. The experiment revealed that Treebot can climb with 1.75 kg of extra weight (Fig. 3.25), which is almost three times its own weight.

3.9 Performance Comparison

Table 3.3 makes a clear comparison among Treebot and other existing tree climbing robots. The best performance for each item is highlighted. The "/"

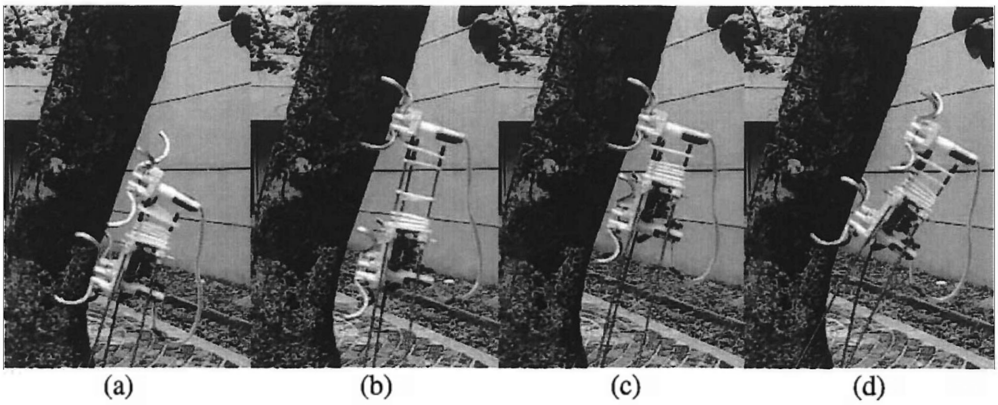


Figure 3.24: 110 degrees slope climbing.

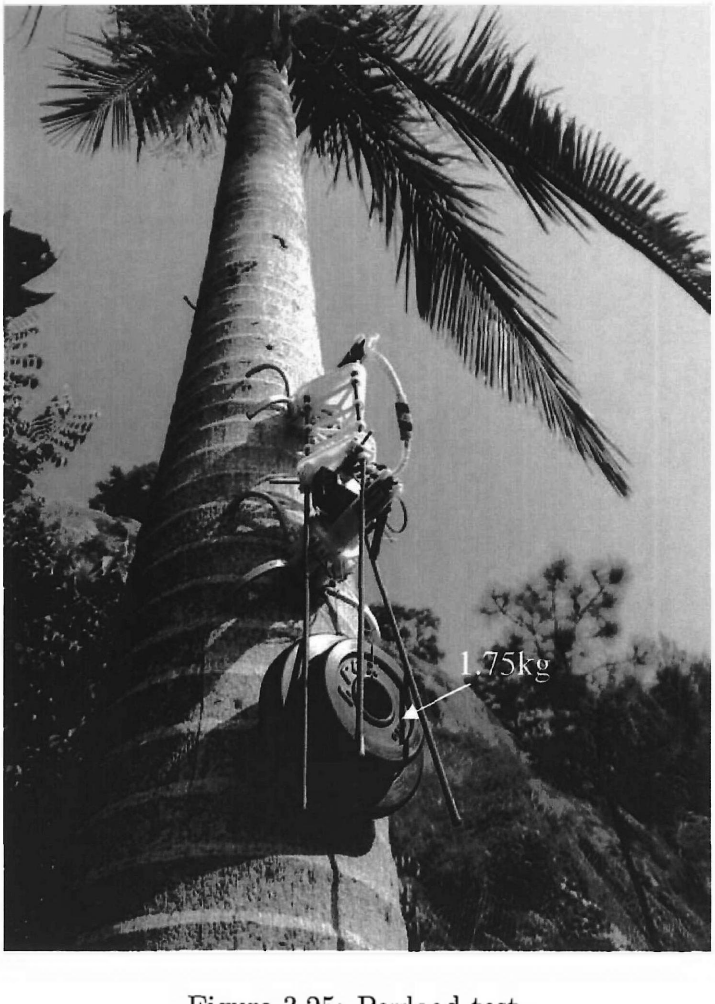


Figure 3.25: Payload test.

sign denotes that the data are not available. By neglecting the unknown data, it can be seen that Treebot ranks best in most aspects except for climbing speed and number of actuators. However, Treebot has only one more actuator than the best performing robot in this respect (Pruning Robot), and although it is slower than the fastest climber (RiSE V3), it can still ascend faster than the other two robots.

3.10 Summary

In summary, a lightweight and compact tree climbing robot, Treebot, has been developed. It is composed of a pair of omni-directional tree grippers that allow the robot to hold onto trees and a novel three DOF continuum body for maneuvering. Numerous experiments were conducted to test its performance. The results reveal that the proposed design is able to climb a wide variety of trees with a high maneuverability. The range of gripping curvatures is also very wide. It is found that Treebot has excellent climbing performance on the trees with surfaces that are not very hard do not have peeling bark.

□ **End of chapter.**

Table 3.3: Comparison among the existing tree climbing robot

	Treebot	Woody [13]	Pruning Robot [14]	RiSE V2 [15]	RiSE V3 [16]	CPR [8]
Weight (kg)	0.65	13.8	15	3.8	5.4	/
Payload capacity (kg)	1.75	/	/	1.5	/	/
Height (mm)	135	310	400	/	/	/
Width (mm)	175	310	400	333	275	/
Length (mm)	370	750	400	583	980	/
Runtime (minutes)	180	/	/	45	30	/
Climb-up speed (mm/min)	733	1164	2400	537	12600	333
Adaptable tree diameter (mm)	64-452	100-150	/<120</	/ - ∞	/<250</	/
Number of actuators	5	/	4	12	8	14
Workspace	curved trunk and branches	straight trunk	straight trunk	straight trunk	straight trunk	slightly curved trunk

Chapter 4

Analysis of the Tree Gripper

4.1 Gripping Force Analysis

The main function of the proposed tree gripper is to attach the robot tightly to a tree surface. The gripper must generate sufficient pull-in force to compensate for the pitch-back moment that results from the action of gravity. The gripping force of the gripper is generated by the penetration of spines into a tree surface.

The gripping position can be divided into two parts: the inner half (smaller than 180 degrees gripping) and the outer half (more than 180 degrees gripping), as illustrated in Fig. 4.1. The figure also illustrates the simplified gripper model and its parameters. When the claws are long enough to grip the outer half, then a large gripping force can easily be achieved, as just maintaining the claw position is enough to lock the gripper onto the substrate (a force-closure grasp). However, if the gripper is to be compact, then the claws cannot be made sufficiently long to always be able to grip the outer

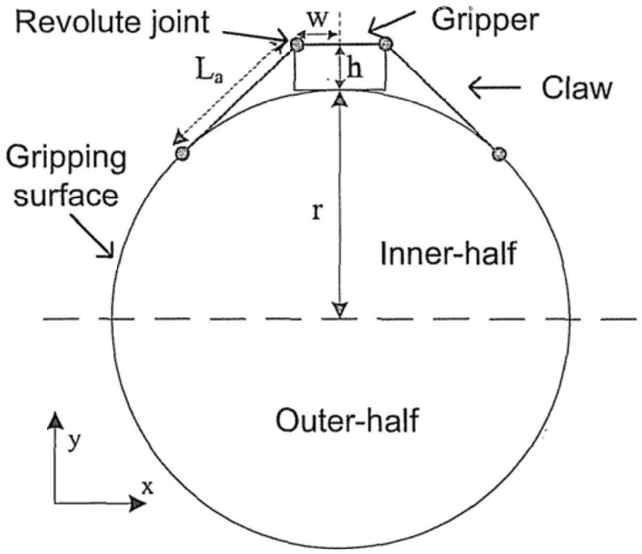


Figure 4.1: Representation of the gripping position.

half of the substrate when the size of tree is large. In such cases, the gripper must generate an adhesive force to avoid being pulled out by the pitch-back moment. The gripper settings must thus be optimized to maximize the fastening force when gripping the inner half of the substrate only.

4.1.1 Generation of the Gripping Force

According to [20], the penetration of a spine can generate both shear force (parallel to the substrate surface) and adhesive force (normal to the substrate surface). The magnitudes of these forces are related to the spine insert angle of the spine and the direction of the acting force. To determine the force generated by the penetration of the spines, the insert angle must be known. It is assumed that the gripping substrate in a cylindrical shape with a radius of r, as shown in Fig. 4.2(a). When a claw penetrates the substrate at a certain orientation (σ), the gripping curvature becomes an ellipse. Fig. 4.2(c) shows the notations for the gripping curvature and the gripper parameters. θ_s is the spine installation angle and θ_c is the the gripping angle of the claw. To find the spine insert angle, the coordinates of the contact point in a $x' - y'$ frame $(x'_\imath, y'_\imath)$ must first be found. This can be achieved by finding the intersection point of the ellipse (E) and the circle (C), which represent the gripping curvature and the motion trajectory of claw respectively:

$$E : \frac{x'^2_\imath}{{a_e}^2} + \frac{y'^2_\imath}{{b_e}^2} = 1 \tag{4.1}$$

$$C : (x'_\imath - x_c)^2 + (y'_\imath - y_c)^2 = {r_c}^2 \tag{4.2}$$

where $b_e = r$, $a_e = r/\cos\sigma$, $x_c = w$, $y_c = r + h$ and $r_c = L_c$.

Once the intersection point has been found, the spine direction vector $(\vec{v}'_s)$ can be determined from the following equation according to Fig. 4.3:

$$\vec{v}'_s = (-\sin\theta_{sd}, \cos\theta_{sd}) \tag{4.3}$$

where $\theta_{sd} = \theta_c - \theta_s$, $\theta_c = \pi - \theta_a - \theta_b$, $\theta_a = \tan^{-1}\left(\frac{h+r}{w}\right)$, $\cos\theta_b = \frac{{R_h}^2+{L_c}^2-{L_\imath}^2}{2R_hL_c}$ and ${L_\imath}^2 = x'^2_\imath + y'^2_\imath$.

To transform the coordinates from the $x' - y'$ frame to the $x - y - z$ frame, a rotation matrix about the y-axis $(Rot_y(\sigma))$ is used. The definition of the rotation matrix $Rot_\imath(\theta)$ can be found in Appendix A.1. The spine direction vector $\vec{v}_s$ and the coordinates of intersection $[\ x_\imath \quad y_\imath \quad z_\imath\ ]$ in the $x - y - z$ frame become:

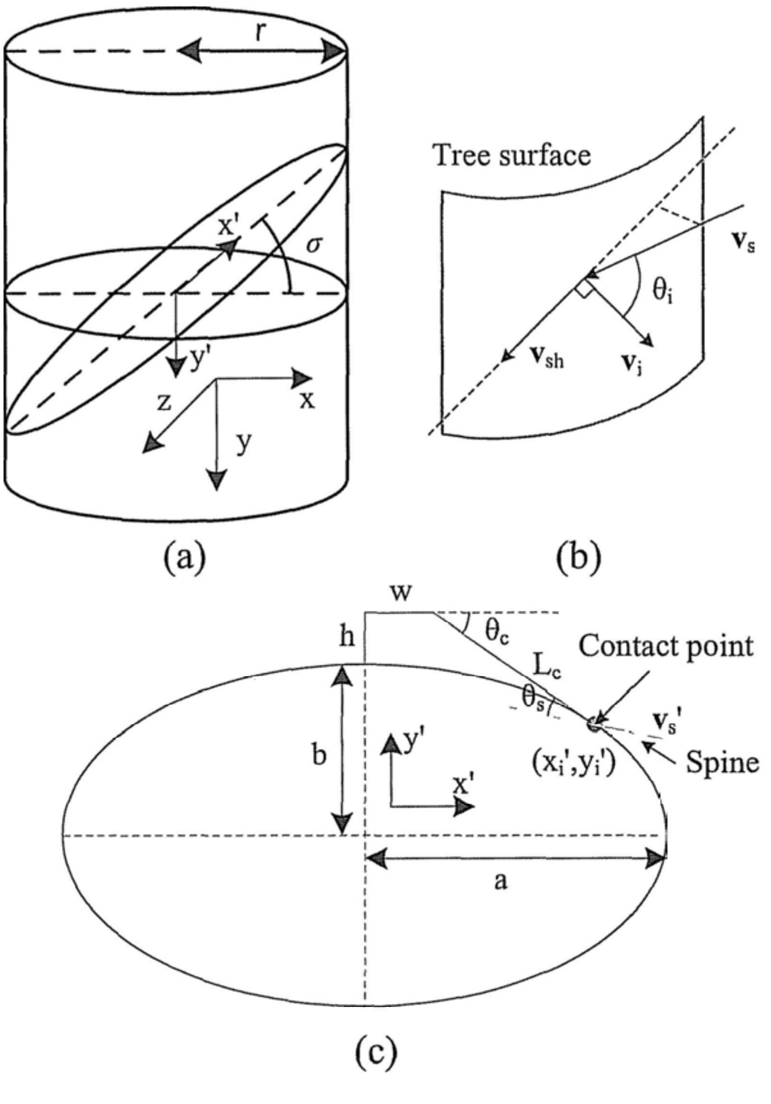


Figure 4.2: Notations for the gripping curvature and gripper parameters.

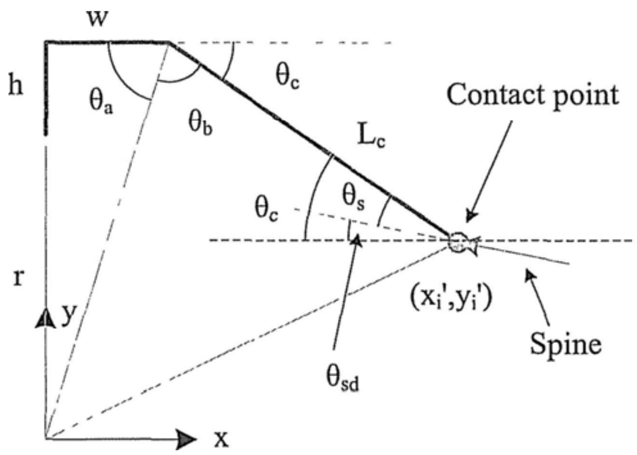


Figure 4.3: Notations for defining the spine direction vector.

$$\vec{v}_s = Rot_y(\sigma)\vec{v}_s' \tag{4.4}$$

$$\begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T = Rot_y(\sigma)\begin{bmatrix} x_i' & y_i' & 0 \end{bmatrix}^T \tag{4.5}$$

As the gripping substrate is assumed to be a perfect cylinder, the normal vector of the intersection point becomes:

$$\vec{v}_i = \left(\frac{x_i}{\sqrt{x_i{}^2 + y_i{}^2}}, \frac{y_i}{\sqrt{x_i{}^2 + y_i{}^2}} \right) \tag{4.6}$$

Referring to Fig. 4.2, the spine insert angle θ_i, i.e., the angle between $\vec{v}_s$ and $\vec{v}_i$ can be obtained from:

$$\theta_i = \cos^{-1}\left(-\vec{v}_i \cdot \vec{v}_s\right) \tag{4.7}$$

As shown in Fig. 4.2(b), the direction of the adhesive force ($\vec{v}_{ad}$) is equivalent to $\vec{v}_i$, and the direction of the shear force ($\vec{v}_{sh}$) can be found by:

$$\vec{v}_{sh} = \vec{v}_i \times \vec{v}_s \times \vec{v}_i \tag{4.8}$$

The total force generated by spine penetration is then:

$$\vec{F} = f_{ad}\vec{v}_{ad} + f_{sh}\vec{v}_{sh} \tag{4.9}$$

where f_{ad} and f_{sh} are the magnitude of the adhesive force and shear force respectively.

According to the experimental results of [20], the relationships among the spine insert angle, adhesive force, and shear force can be approximated as Fig. 4.4. The pull-in force to compensate for the pitch-back moment is contributed by the adhesive force and shear force of the spine along the z-axis, i.e., F_z where $\vec{F} = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}$.

4.1.2 Selection of the Spine Installation Angle

To generate maximal pull-in force, the spine installation angle (θ_s) must be optimized. As the gripper is designed for omni-directional gripping about its principal axis, the gripping force in different gripping orientations must be considered. As the claws are symmetrical in both the x- and y-axis, the three gripping orientations shown in Fig. 4.5 are equivalent to sixteen gripping orientations. The optimization focuses on these three gripping orientations.

The pull-in force of the gripper is contributed by the force of all claws. The directions of the claws of the three gripping orientations are composed of $\sigma = 0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2} = \sigma_a, \sigma_b, \sigma_c, \sigma_d, \sigma_e$ where gripping orientation 1 is composed of $2\sigma_a + 2\sigma_e$; gripping orientation 2 is composed of $4\sigma_c$; and gripping orienta-

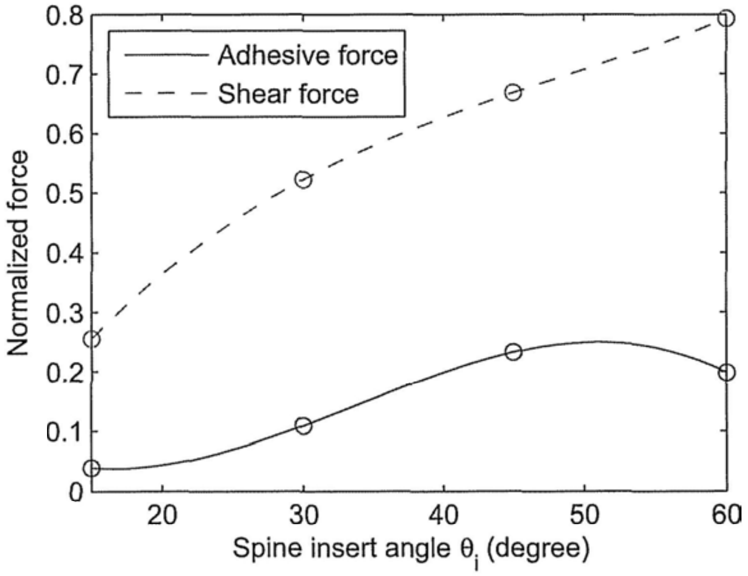


Figure 4.4: Relationships among the spine insert angle, adhesive force, and shear force [20].

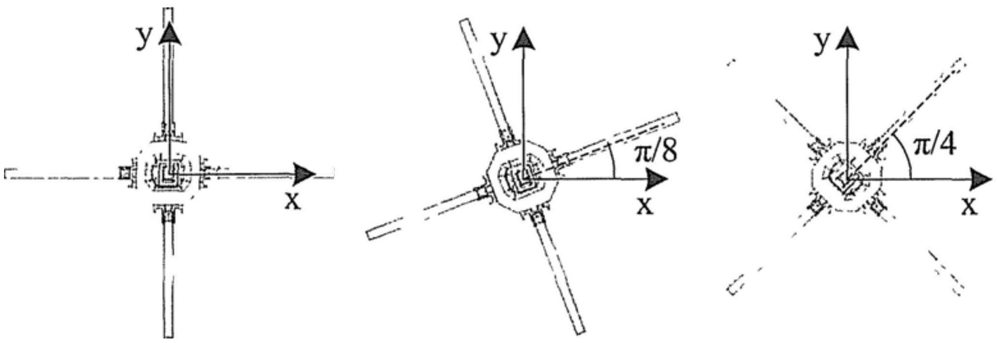


Figure 4.5: Representative gripping orientations.

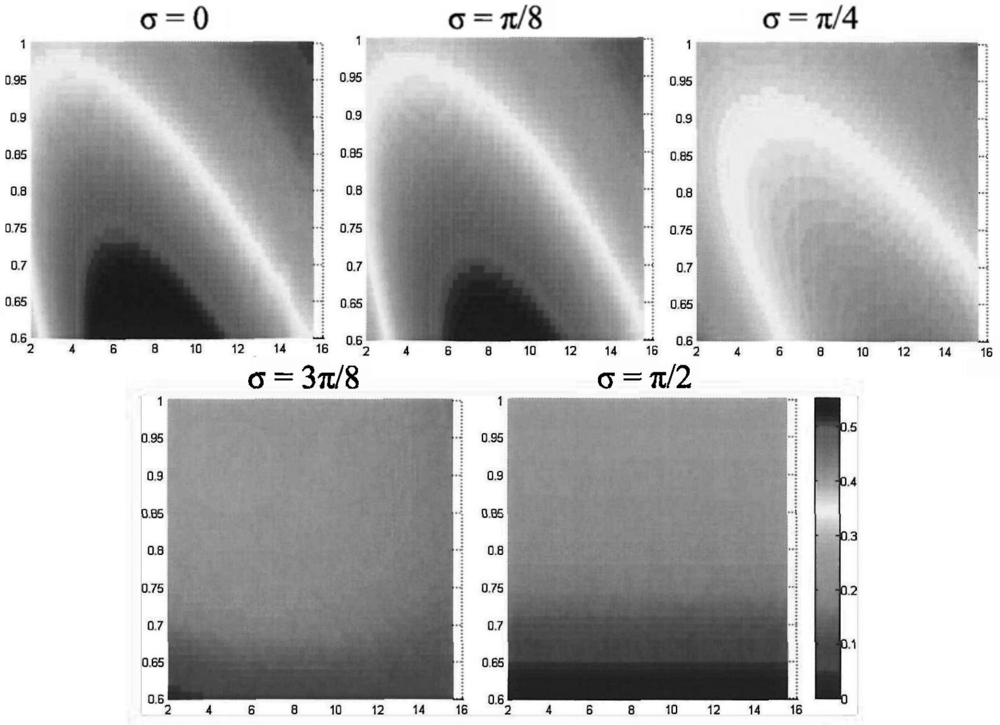


Figure 4.6: Relationships among the spine insert angle in radians (y-axis), the curvature of the gripping surface in m^{-1} (x-axis), and the normalized pull-in force (gradient) for different claw directions.

tion 3 is composed of $2\sigma_b + 2\sigma_d$. Fig. 4.6 shows the relationships among the spine installation angle, curvature of gripping surface and pull-in force (F_i, $i = a, b, c, d, e$), for different claw directions with the gripper parameters: w = 20 mm, L_c = 100 mm, h = 25 mm and the range of tree radius: 64.5354 mm < r < 500 mm (the minimum target radius is the radius at which the gripper can at only grip the inner-half of the substrate). with different claw directions are shown in Fig. 4.6.

The total gripping force generated with the different gripping orientations can be obtained by summing the results in Fig. 4.6 according to the combination of claw directions. To help determine the optimal spine installa-

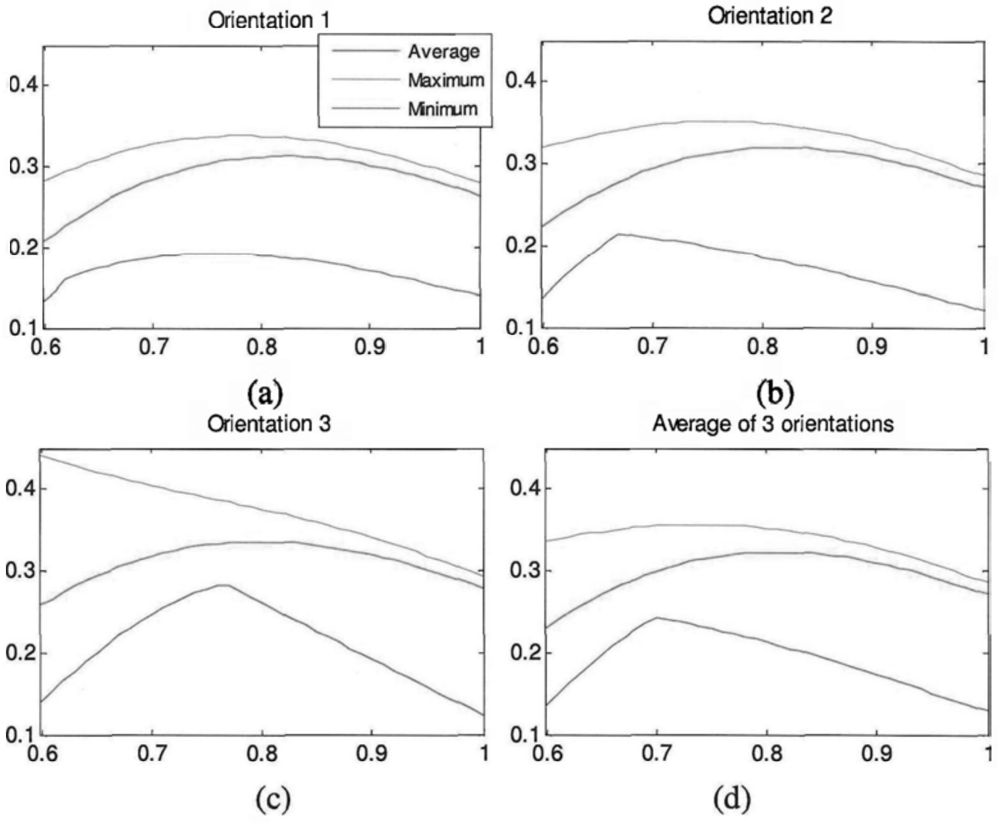


Figure 4.7: Average, maximum and minimum normalized pull-in force of the gripper (y-axis) for different spine installation angles in radians (x-axis).

tion angle, the average, maximum and minimum pull-in force of the gripper for different spine installation angles and gripping orientations are found out as illustrated in Fig. 4.7. Fig. 4.7(d) also shows the average force generated by three gripping orientations.

Fig. 4.7(a), (b) and (c) show no obvious difference in the average pull-in forces. This means that the gripping force is similar for different gripping orientations. The optimal spine installation angle can be determined by referring to Fig. 4.7(d). When $\theta_s = 0.7$ rad, the minimal pull-in force is greatest, and the average force is relatively large. This is thus chosen as the spine

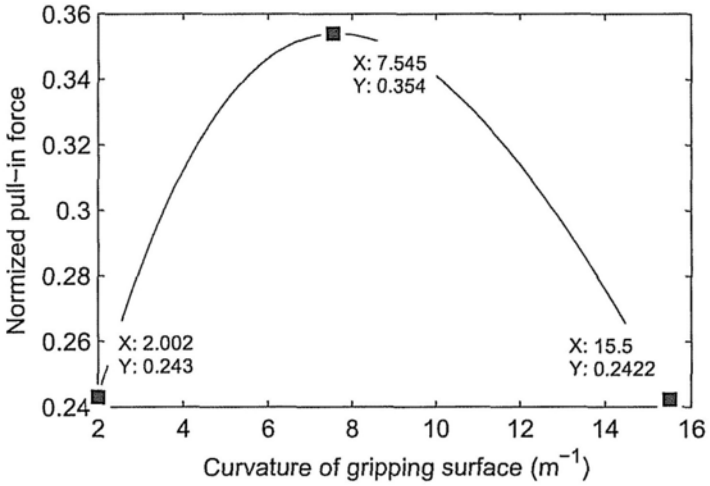


Figure 4.8: Optimized pull-in force of the gripper for different gripping curvatures.

installation angle. Fig. 4.8 shows the normalized pull-in force generated by the gripper for different gripping curvatures when θ_s =0.7 rad. Referring to Fig. 4.7(a), (b) and (c), the gripping orientation 3 generates the largest pull-in force by this optimal setting. Gripping orientation 3 is thus used as the default setting for Treebot.

4.1.3 Generation of the Directional Penetration Force

In [20], it is assumed that the direction of the force acting on a spine F_p is equal to the spine insert angle. Pushing a claw into a substrate with a desired directional force usually requires two actuators. RiSE V2 [15], for example, uses two active joints to accomplish this task. One actuator provides a pushing force toward a surface, and the other pulls the spine toward the central axis of the robot body. Wile [42] proposed a mechanism that provides the desired directional force using one actuator to make the

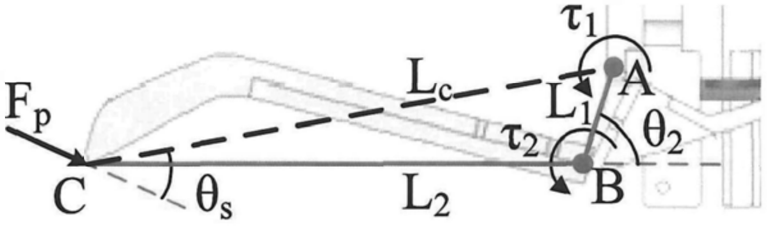


Figure 4.9: Notations for the mechanism of the claw.

gripper lighter. However, this is only applicable on flat surfaces.

In view of this, a preloaded two bar linkage mechanism is proposed to generate the desired directional force, which requires only one actuator and is able to adapt to irregular surfaces. The notations for the mechanism of the claw are shown in Fig. 4.9. L_1 denotes the length of link AB (Phalanx 1), and L_2 denotes the length of link BC (Phalanx 2). Link AC is the simplified claw introduced in Fig. 4.3. Joints A and B are passive revolute joints installed with a pre-compressed mechanical spring. The torque generated by spring on joints A and B are denoted as τ_1 and τ_2 respectively. The turning angle of Joint B is limited such that the distance between A and C does not exceed L_c.

To analyze the direction of the acting force, links AB and BC are divided as shown in Fig. 4.10. It is assumed that in this position, point C penetrates the substrate. Hence, it can be assumed that points A and C are fixed revolute joints. The free body diagrams of links AB and BC are constructed as shown in Fig. 4.10(b) and (c).

From Fig. 4.10(b), the equilibrium equations can be divided as:

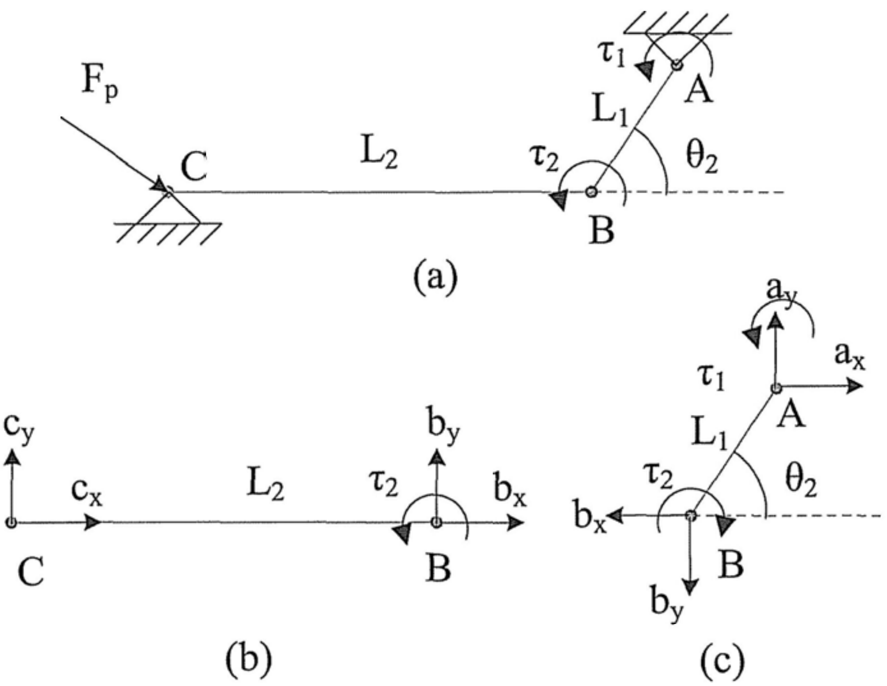


Figure 4.10: Free body diagrams of links AB and BC.

$$c_x + b_x = 0 \tag{4.10}$$

$$c_y + b_y = 0 \tag{4.11}$$

$$\tau_2 - c_y L_2 = 0 \tag{4.12}$$

From Fig. 4.10(c), the equilibrium equations can be divided as:

$$a_x + b_x = 0 \tag{4.13}$$

$$a_y + b_y = 0 \tag{4.14}$$

$$\tau_1 - \tau_2 + L_1(b_y cos\theta_2 - b_x sin\theta_2) = 0 \tag{4.15}$$

Sub. (4.10), (4.11), and (4.12) into (4.15):

$$c_x = \frac{\tau_2 - \tau_1 + \tau_2 \frac{L_1}{L_2} \cos\theta_2}{L_1 \sin\theta_2} \tag{4.16}$$

Divide (4.12) by (4.16):

$$\frac{c_y}{c_x} = \frac{L_1 \sin\theta_2}{L_2 \left(1 - \frac{\tau_1}{\tau_2} + \frac{L_1}{L_2} \cos\theta_2\right)} \tag{4.17}$$

The angle of the pushing force of the spine F_p is equal to the spine insert angle, that is,

$$\begin{aligned} \tan\left(\angle ACB - \theta_s\right) = \frac{c_y}{c_x} &= \frac{L_1 \sin\theta_2}{L_2 \left(1 - \frac{\tau_1}{\tau_2} + \frac{L_1}{L_2} \cos\theta_2\right)} \\ \Rightarrow \frac{\tau_1}{\tau_2} &= 1 + \frac{L_1}{L_2}\left[\cos\theta_2 - \frac{\sin\theta_2}{\tan\left(\angle ACB - \theta_s\right)}\right] \end{aligned} \tag{4.18}$$

With the gripper parameters: $L_1 = 20$ mm, $L_2 = 85$ mm, the torque ratio between joints A and B (τ_1/τ_2) should be around 1.436 to generate the appropriate angle of pushing force.

4.2 Experiments and Results

Various experiments were implemented on a variety of trees to evaluate the performance of the omni-directional tree gripper. In the experiments, the gripper first gripped the tree without any external force being applied. An external pull-out force was then applied normal to the gripping surface to test how much force was needed to pull the gripper out of the tree. The maximum

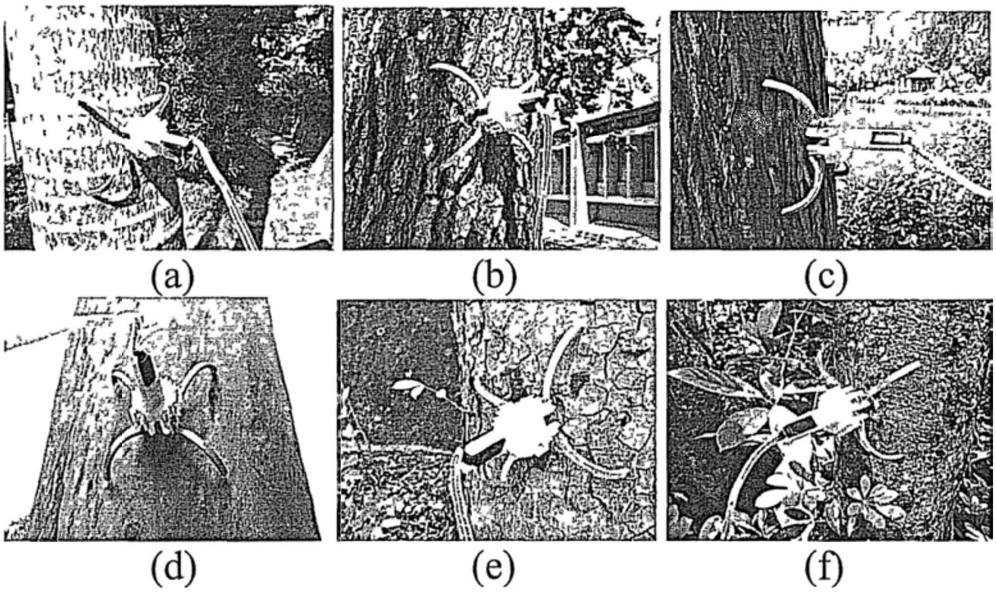


Figure 4 11· Experiments on different types of trees: (a) Roystonea regia; (b) Cinnamomum camphora; (c) Taxodium distichum; (d) Eucalyptus citriodora; (e) Khaya senegalensis; (f) Bauhinia variegata var. candida.

pull-out force was limited to 40N to avoid breaking the gripper. Eighteen types of trees with different surface curvatures were tested. Fig. 4.11 shows some of the tested trees. The curvature of the trees, bark textures, and the maximum pull-out force with different gripping orientations are summarized in Table 4.1. In the table, O1, O2, and O3 represent gripping orientations 1, 2, and 3 introduced in Fig. 4.5, respectively. The curvature of tree (C) is obtained by:

$$C = \frac{1}{D/2} \tag{4.19}$$

where D is the diameter of tree. Table 4 1 shows that on thc fiıst ten types of trees (No. 1-10), the performance was excellent. The gripper was able to generate over 40N of pull-in force in any gripping orientation. However, the results also reveal that the gripper does not work well on some

Table 4.1: Maximum pull-in force on different species of trees

No.	Trees	Bark texture	Curvature (m^{-1})	Pull-in force (N)		
				O1	O2	O3
1	Bombax malabaricum	Rough	4.4	>40	>40	>40
2	Acacia confuse	Smooth	5.6	>40	>40	>40
3	Ficus microcarpa	Smooth	4.6	>40	>40	>40
4	Livistona chinensis	Fissured	8.4	>40	>40	>40
5	Callistemon viminalis	Ridged and Furrowed	6.3	>40	>40	>40
6	Bauhinia variegata var. candida	Smooth	8.1	>40	>40	>40
7	Bauhinia variegate	Smooth	8.8	>40	>40	>40
8	Araucaria heterophylla	Banded	7.2	>40	>40	>40
9	Bauhinia blakeana	Smooth	6.7	>40	>40	>40
10	Roystonea regia	Smooth, shallowly fissured	6.2	>40	>40	>40
			4.8	15	15	20
11	Taxodium distichum	Fibrous, exfoliating	11.2	29	30	30
			6.0	12	10	10
12	Casuarina equisetifolia	Slightly exfoliating	6.9	11	13	12
13	Cinnamomum camphora	Ridged and furrowed, exfoliating	5.2	20	12	5
14	Khaya senegalensis	Blocky, exfoliating	4.0	10	10	10
15	Melaleuca quinquenervia	Sheeting, soft, exfoliating	4.5	5	5	5
16	Delonix regia	Smooth	6.7	24	24	25
17	Mangifera indica	Shallowly fissured	4.1	20	22	25
18	Eucalyptus citriodora	Smooth, soft	4.1	18	16	20

types of trees, and particularly those with bark that peels off easily. In such cases, when a large pull-out force was applied, the gripper was pulled out as the bark peeled off (No. 11-15). Further, for soft trees the pull-out force broke the bark (No. 16-18).

The experimental results indicate that on most of the trees, the maximum pull-in force of the gripper in all gripping orientations is similar, and matches the analytical results reported in Section 4.1. The only exception is tree No. 13. This is because the bark of this trees peels off easily and its surface is not smooth, but rather has many vertical grooves (see Fig. 4.11(b)). Gripping orientation 1 is better in this scenario, as it creates a pair of claws oriented perpendicular to the vertical groove, which allows the claws to penetrate deeper into the tree to generate a larger force.

As mentioned in the previous section, the gripping curvature affects the pull-in force of the gripper. This phenomenon is clearly demonstrated by the experimental result for tree No. 10, where the generated pull-in force with a 6.2 m^{-1} surface curvature is larger than that with a 4.8 m^{-1} surface curvature. However, the result for tree No. 11 does not match the analytical result (Fig. 4.8). This is because the tree with a 6.0 m^{-1} surface curvature was older, and its bark will be peeled off easily , whereas the tree with a 11.2 m^{-1} surface curvature was younger and its bark will not be peeled off easily.

In the experimental results, and especially those for trees No. 16-18, it is clear that using gripping orientation 3 generates the largest pull-in force, which matches the analytical results.

4.3 Summary

In summary, the mechanism of the tree gripper is discussed. The optimization of the gripper in terms of torque distribution and spine installation angle to provide a large gripping force over a wide range of curvatures is presented.

Numerous experiments have been carried out. The experimental results reveal that the proposed gripper performs well on a wide range of trees. However, the gripping performance strongly depends on the properties of the tree surface. The gripper works well only on trees with bark that does not peel off or break easily.

□ **End of chapter.**

Chapter 5

Kinematics and Workspace Analysis

5.1 Kinematic Analysis

Fig. 5.1 shows the configuration of Treebot. l_f and l_r represent the distance from the end of the continuum body to the center of the front and rear gripper respectively. h_g denotes the distance between the base of the gripper and the continuum body. The reference frames for the front and rear grippers are also illustrated in the figure. The direction of a gripper refers to the direction along the positive z-axis, where a normal direction refers to the direction toward the positive x-axis.

Jones [17] developed a kinematic model of a continuum type manipulator. It formulates the mapping between the posture (S, κ, ϕ) and the input coordinates (l_1,l_2,l_3). S, κ, and ϕ denote the length, curvature, and bending direction of the virtual tendon, respectively. The virtual tendon represents

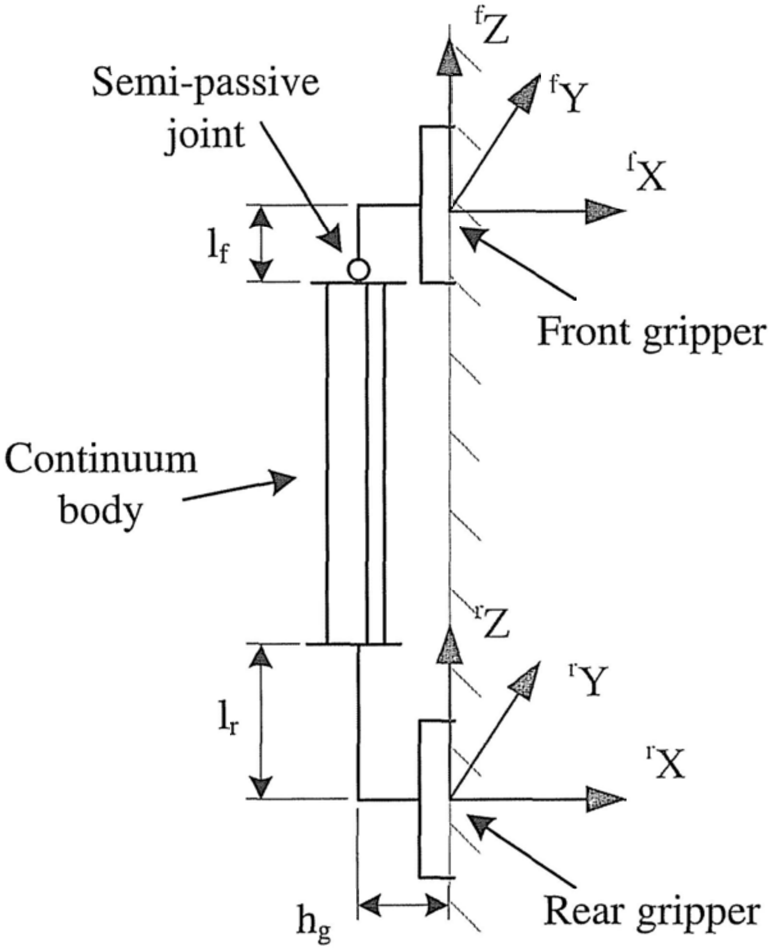


Figure 5.1: Configuration of Treebot.

the centerline of the continuum manipulator. l_i denotes the length of each tendon, and d is the distance between the tendons and the virtual tendon. Fig. 5.2 shows the notation used to represent the parameters of each tendon and the position of the virtual tendon.

The kinematic model is also applicable to the proposed continuum body. According to Jones [17], the forward and inverse kinematics in our convention can be defined as:

Inverse Kinematics $(l_1, l_2, l_3) \leftarrow f\,(S, \kappa, \phi)$:

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = S \begin{bmatrix} 1 + d\kappa\cos\phi \\ 1 - \kappa d \sin\left(\pi/6 - \phi\right) \\ 1 - \kappa d \sin\left(\pi/6 + \phi\right) \end{bmatrix} \tag{5.1}$$

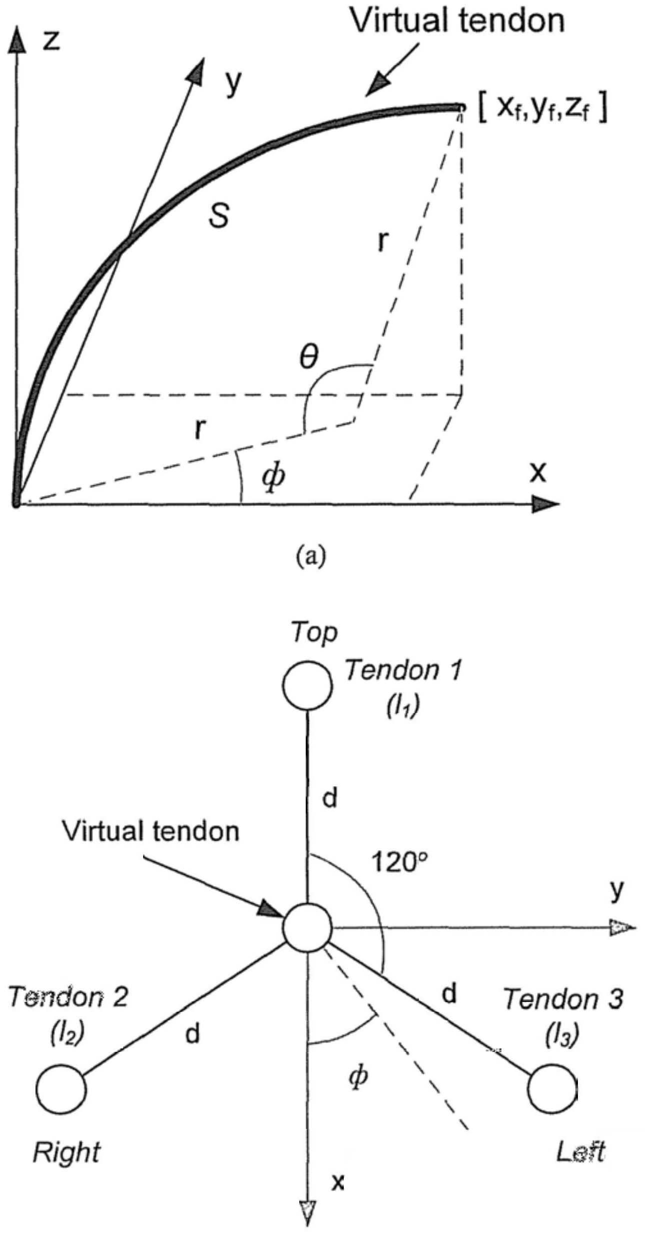
Forward Kinematics $(S, \kappa, \phi) \leftarrow f\,(l_1, l_2, l_3)$:

$$\begin{bmatrix} S \\ \kappa \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{l_1+l_2+l_3}{3} \\ 2\frac{\sqrt{{l_1}^2+{l_2}^2+{l_3}^2-l_1l_2-l_2l_3-l_1l_3}}{d(l_1+l_2+l_3)} \\ \cot^{-1}\left(-\frac{\sqrt{3}}{3}\frac{l_3+l_2-2l_1}{l_2-l_3}\right) \end{bmatrix} \tag{5.2}$$

In addition, $\kappa = 1/r$ and $S = r\theta$.

The mapping between the positions coordinates and the posture of Treebot should also be formulated. To formulate the kinematics of Treebot, l_f and l_r must be considered. As a result, the kinematics of Treebot is developed by extending (5.1) and (5.2).

In view of the rear gripper frame as shown in Fig. 5.3(a), the mapping



(a)

(b)

Figure 5.2: Notations for defining the position and parameters of the continuum manipulator.

between the end point (front gripper) and the posture of Treebot are formulated as,

$({}^r x_f, {}^r y_f, {}^r z_f) \leftarrow f\,(S, \kappa, \phi):$

$$\begin{bmatrix} {}^r x_f \\ {}^r y_f \\ {}^r z_f \end{bmatrix} = \begin{bmatrix} \left(\frac{1}{\kappa}\left[1-\cos\left(\kappa S\right)\right] + l_f \sin\left(\kappa S\right)\right)\cos\phi \\ \left(\frac{1}{\kappa}\left[1-\cos\left(\kappa S\right)\right] + l_f \sin\left(\kappa S\right)\right)\sin\phi \\ \frac{1}{\kappa}\sin\left(\kappa S\right) + l_f\cos\left(\kappa S\right) + l_r \end{bmatrix} \tag{5.3}$$

$(S, \kappa, \phi) \leftarrow f\,({}^r x_f, {}^r y_f, {}^r z_f)$:

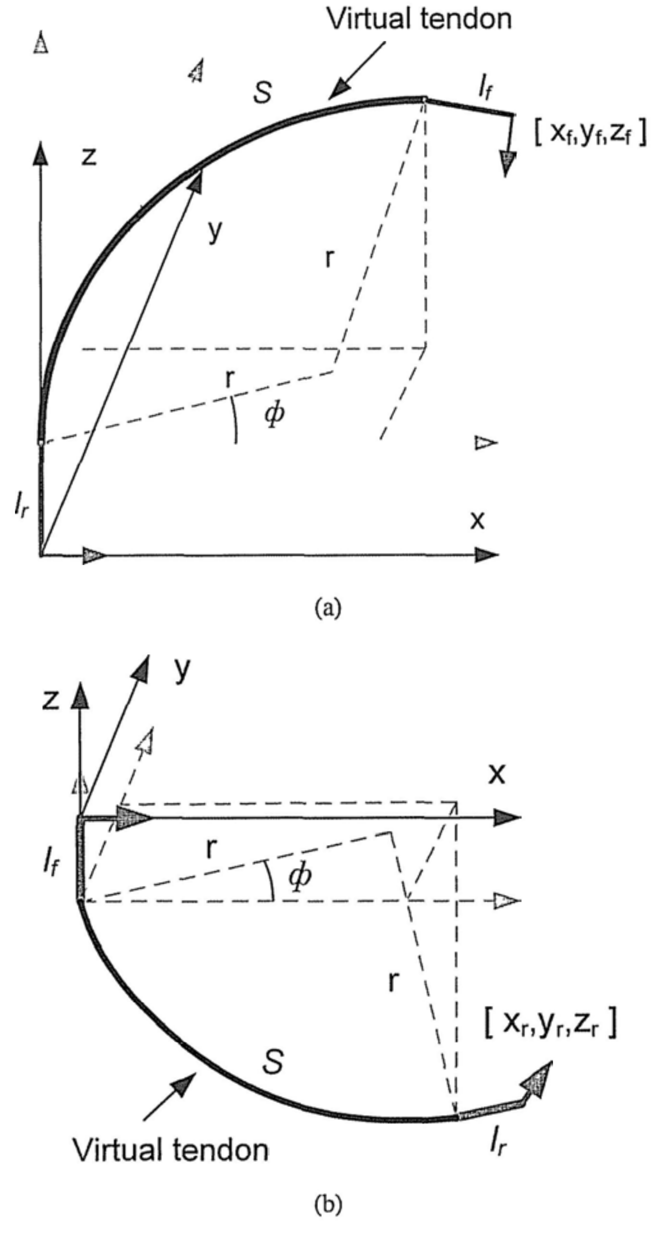
$$\begin{bmatrix} S \\ \kappa \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{1}{\kappa}\tan^{-1}\left(\frac{2{}^r\hat{x}_f\left({}^r\hat{z}_f + l_f\right)}{\left({}^r\hat{z}_f + l_f\right)^2 - {}^r\hat{x}_f^2}\right) \\ \frac{2{}^r\hat{x}_f}{{}^r\hat{x}_f^2 + {}^r\hat{z}_f^2 - {l_f}^2} \\ \tan^{-1}\frac{{}^r y_f}{{}^r x_f} \end{bmatrix} \tag{5.4}$$

where ${}^r\hat{x}_f = {}^r x_f \cos\phi + {}^r y_f \sin\phi$ and ${}^r\hat{z}_f = {}^r z_f - l_r$.

In view of the front gripper frame as shown in Fig. 5.3(b), the mapping between the end point (rear gripper) and the posture of Treebot are formulated as,

$\left({}^f x_r, {}^f y_r, {}^f z_r\right) \leftarrow f\,(S, \kappa, \phi)$:

$$\begin{bmatrix} {}^f x_r \\ {}^f y_r \\ {}^f z_r \end{bmatrix} = \begin{bmatrix} \left(\frac{1}{\kappa}\left[1-\cos\left(\kappa S\right)\right] + l_r \sin\left(\kappa S\right)\right)\cos\phi \\ \left(\frac{1}{\kappa}\left[1-\cos\left(\kappa S\right)\right] + l_r \sin\left(\kappa S\right)\right)\sin\phi \\ -\left(\frac{1}{\kappa}\sin\left(\kappa S\right) + l_r\cos\left(\kappa S\right) + l_f\right) \end{bmatrix} \tag{5.5}$$



Figuıe 5.3: Notations foı defining the kinematics of Treebot.

$(S, \kappa, \phi) \leftarrow f\left({}^f x_r, {}^f y_r, {}^f z_r\right)$:

$$\begin{bmatrix} S \\ \kappa \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{1}{\kappa}\tan^{-1}\left(\frac{2{}^f\hat{x}_r\left({}^f\hat{z}_r+l_r\right)}{\left({}^f\hat{z}_r+l_r\right)^2-{}^f\hat{x}_r^2}\right) \\ \frac{2{}^f\hat{x}_r}{{}^f\hat{x}_r^2+{}^f\hat{z}_r^2-{l_r}^2} \\ \tan^{-1}\frac{{}^f y_r}{{}^f x_r} \end{bmatrix} \tag{5.6}$$

where ${}^f\hat{x}_r = {}^f x_r \cos\phi + {}^f y_r \sin\phi$ and ${}^f\hat{z}_r = -{}^f z_r - l_f$.

On top of that, the vector transformation between the front and rear gripper frame can be achieved by:

$${}^f\vec{v} = Rot_z\left(\phi\right) Rot_y\left(-\theta\right) Rot_z\left(-\phi\right) {}^r\vec{v} \tag{5.7}$$

$${}^r\vec{v} = Rot_z\left(\phi\right) Rot_y\left(\theta\right) Rot_z\left(-\phi\right) {}^f\vec{v} \tag{5.8}$$

In the notations, the superscripts r and f denote the front and rear gripper frame respectively.

The detailed derivation of the equations can be found in Appendix A.

5.2 Workspace Analysis

By using the proposed continuum mechanism, Treebot is able to reach any position in 3D space theoretically. Fig. 5.4 illustrates some of the reachable positions of the continuum body. The arrows (green) in the figure represent the initial direction of the front gripper. However, the workspace of the continuum body is not equivalent to the climbing workspace of Treebot as the gripper works directionally. In addition, the physical constraints of the

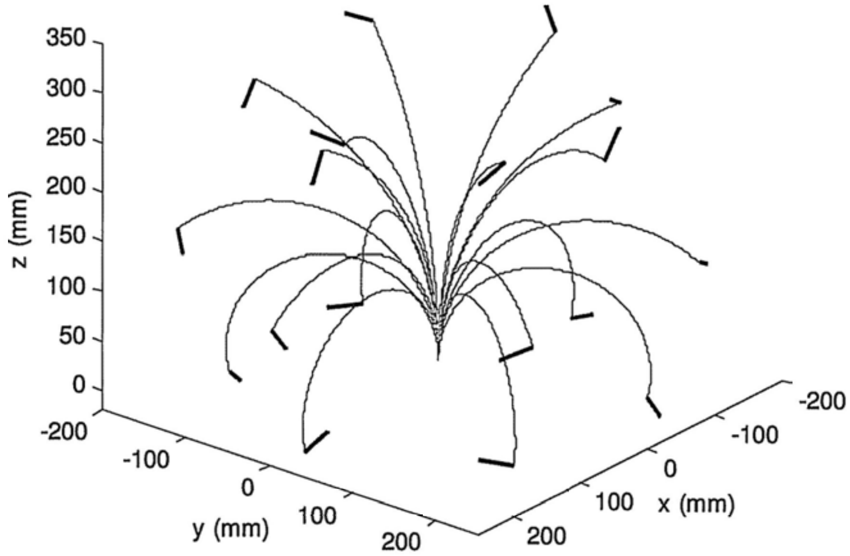


Figure 5.4: Workspace of the continuum body.

continuum body must be considered.

5.2.1 Physical Constraints

Maximum length of extension

In theory, the tendon-driving mechanism can extend infinitely. In practice, it is impossible to have a tendon of an infinite length. The length of tendons is thus restricted in certain length.

Maximum bending curvature

Bending the tendon to a greater curvature requires a larger bending force. The limit of this force is determined by the power of the tendon-driving motor. It is assumed that the tendon driving-motors have same maximum torque and that each tendon is identical. Hence, the maximum bending curvature of Treebot is assumed to be a constant.

Maximum climbing slope

As the additional revolute joint is a passive joint, the maximum climbing slope is determined by the location of the center of mass. Fig. 5.5(a) illustrates the relationship between the location of the center of mass and the limit of the climbing slope. If the climbing slope exceeds the limit, then the rear gripper is pulled out of the gripping substrate by the force of gravity, as illustrated in Fig. 5.5(b). In this case, bending the body can make the rear gripper contact the surface, but there is no method for arranging the bottom gripper normal to the tree surface, as shown in Fig. 5.5(c). If the climbing slope does not exceed the limit by too much, then the rear gripper may still be able to grip the tree surface and provide sufficient gripping force, as some tolerance of gripping direction is allowed. Hence, Treebot may still be able to climb continuously if the front gripper is sufficiently flexible to be appressed to the gripping surface.

Maximal angle of twist

As the gripper contains a semi-passive joint, it has certain flexibility to change its direction. In the semi-passive joint, the range of twist about the y- and z-axis is $\pm\pi/3$ and $\pm\pi/6$ respectively. As a result, the workspace of the gripper is a spherical surface, as illustrated in Fig. 5.6. If the gripper can reach a certain position but the required angle of twist exceeds the range, then it is not an admissible climbing position.

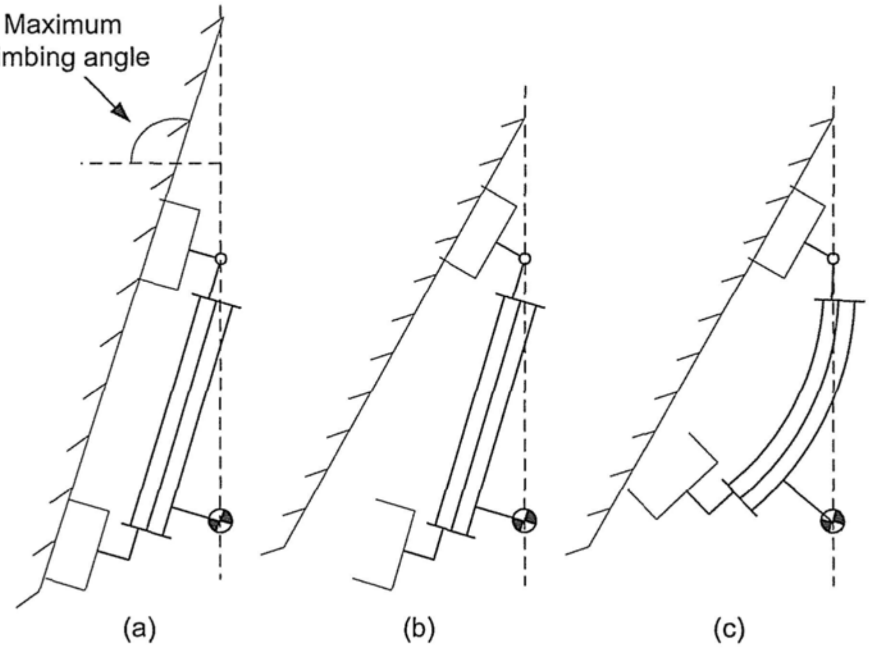


Figure 5.5: Relationship between the location of the center of mass and the climbing slope.

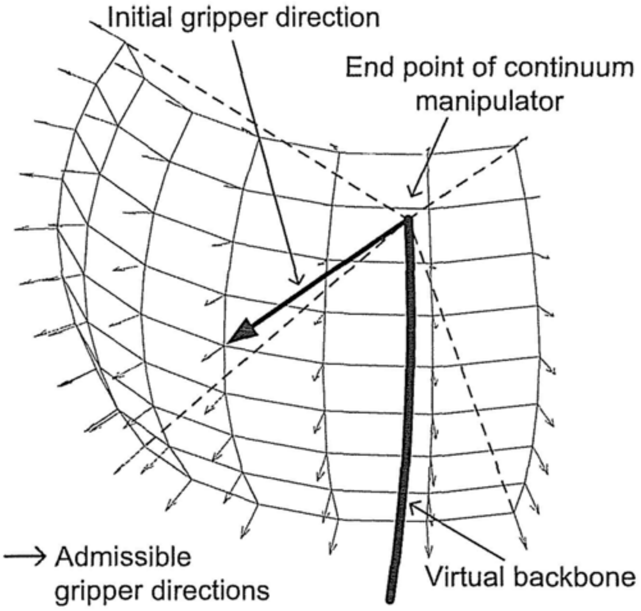


Figure 5.6: Workspace of the gripper at each reachable position.

5.2.2 Admissible Workspace on a Tree Surface

To determine the motion of Treebot, the admissible climbing workspace on the tree surface must be identified. It is assumed that no external force is acting on the robot, and that its weight is negligible.The compliance effect can thus be neglected. To ensure that the gripper can attach to the tree at the target position, certain physical constraints must not be violated. The maximum inclined angle is a constant constraint. The constraints of the continuum body, including the maximum length of extension and bending curvature, can be determined using the kinematic model. The required angles of twist at a particular position can be obtained from the normal direction of the surface.

In reality, the posture of the continuum manipulator may not exactly equal to the analytical result as shown in Fig. 5.4 due to the gravitational force. As mentioned in Chapter 3, the continuum manipulator can be deformed by external force due to the inherent passive compliance. The magnitude of deformation is inversely proportional to the rigidity of the springs and proportional to the weight of Treebot.

Required angle of twist

By giving a shape of tree and the position and orientation of the rear gripper (θ_{rx} and θ_{ry}), the angle of twist required to place the front gripper appressed on the tree surface at the target position can be determined. Fig. 5.7 illustrates the parameters and coordinate relationship between Treebot and a model of tree. The geometry of a segment of tree is modeled as a straight or curved cylinder. The radius of the tree R_{tree}, length of the tree segment S_{tree}, bending direction ϕ_{tree}, and bending curvature κ_{tree} are used to represent the shape of tree. In that, S_{tree}, ϕ_{tree}, and κ_{tree} represents the shape of the centerline of the tree model (similar to the virtual tendon). The target position of the front gripper is defined by the angle of change θ_t and the length of the centerline S_t for the target position. The distance between the continuum body and the tree surface is defined as h_g. The center of the rear gripper is located at the origin of the reference frame of the tree $({}^T x - {}^T y - {}^T z)$. According to 5.3 and 5.8, the coordinates $\vec{P}_t$ and the normal vector $\vec{n}_t$ of the target position can be obtained by:

$$ {}^T\vec{n}_t = -Rot_z\left(\phi_{tree}\right) Rot_y\left(\theta_{tree}\right) Rot_z\left(-\phi_{tree}\right) \begin{bmatrix} \cos\theta_t \\ \sin\theta_t \\ 0 \end{bmatrix} \tag{5.9} $$

$$ {}^T\vec{P}_t = \frac{1}{\kappa_{tree}} \begin{bmatrix} [1-\cos\theta_{tree}]\cos\phi_{tree} \\ [1-\cos\theta_{tree}]\sin\phi_{tree} \\ \sin\theta_{tree} \end{bmatrix} + \left(h_g + R_{tree}\right) \left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - {}^T\vec{n}_t \right) \tag{5.10} $$

where $\theta_{tree} = \kappa_{tree} S_t$.

The target position and normal vector can be transformed to the rear gripper frame $({}^r x - {}^r y - {}^r z)$ by

$$ {}^r\vec{P}_t = Rot_y\left(-\theta_{ry}\right) Rot_x\left(-\theta_{rx}\right) {}^T\vec{P}_t \tag{5.11} $$

$$ {}^t\vec{n}_t = Rot_y\left(-\theta_{ry}\right) Rot_x\left(-\theta_{rx}\right) {}^T\vec{n}_t \tag{5.12} $$

where θ_{rx} and θ_{ry} denote the angles between the tree and the rear gripper frame, as illustrated in Fig. 5.7.

According to 5.7, the normal vector for the front gripper frame can be determined by:

$$^{f}\vec{n}_t = Rot_z\left(\phi_B\right) Rot_y\left(-\kappa_B S_B\right) Rot_z\left(-\phi_B\right) {}^{r}\vec{n}_t \tag{5.13}$$

where S_B, κ_B, and ϕ_B are the length, bending curvature and direction of bend of the continuum body.

Finally, the angle of twist about ^{f}y- and ^{f}z-axis to appress the front gripper to the target surface can be determined by:

$$\theta_{twist\_y} = \tan^{-1}\left({}^{f}n_{t\_z}/{}^{f}n_{t\_x}\right) \tag{5.14}$$

$$\theta_{twist\_z} = \tan^{-1}\left({}^{f}n_{t\_y}/{}^{f}n_{t\_x}\right) \tag{5.15}$$

where $^{f}\vec{n}_t = \begin{bmatrix} {}^{f}n_{t\_x} & {}^{f}n_{t\_y} & {}^{f}n_{t\_z} \end{bmatrix}$. The subscript i $(i \in x, y, z)$ denotes the i component value in a vector.

Admissible target position

The admissible gripping positions can be determined by considering all of the necessary constraints. Fig. 5.8 illustrates the admissible gripping positions of the front gripper on a straight tree for different directions of the rear gripper. In the figures, the arrow at the bottom (red) denotes the direction of the rear gripper. The inner circle (blue) illustrates the circumference of the tree. The dots (blue) are the admissible positions of the front gripper with the small

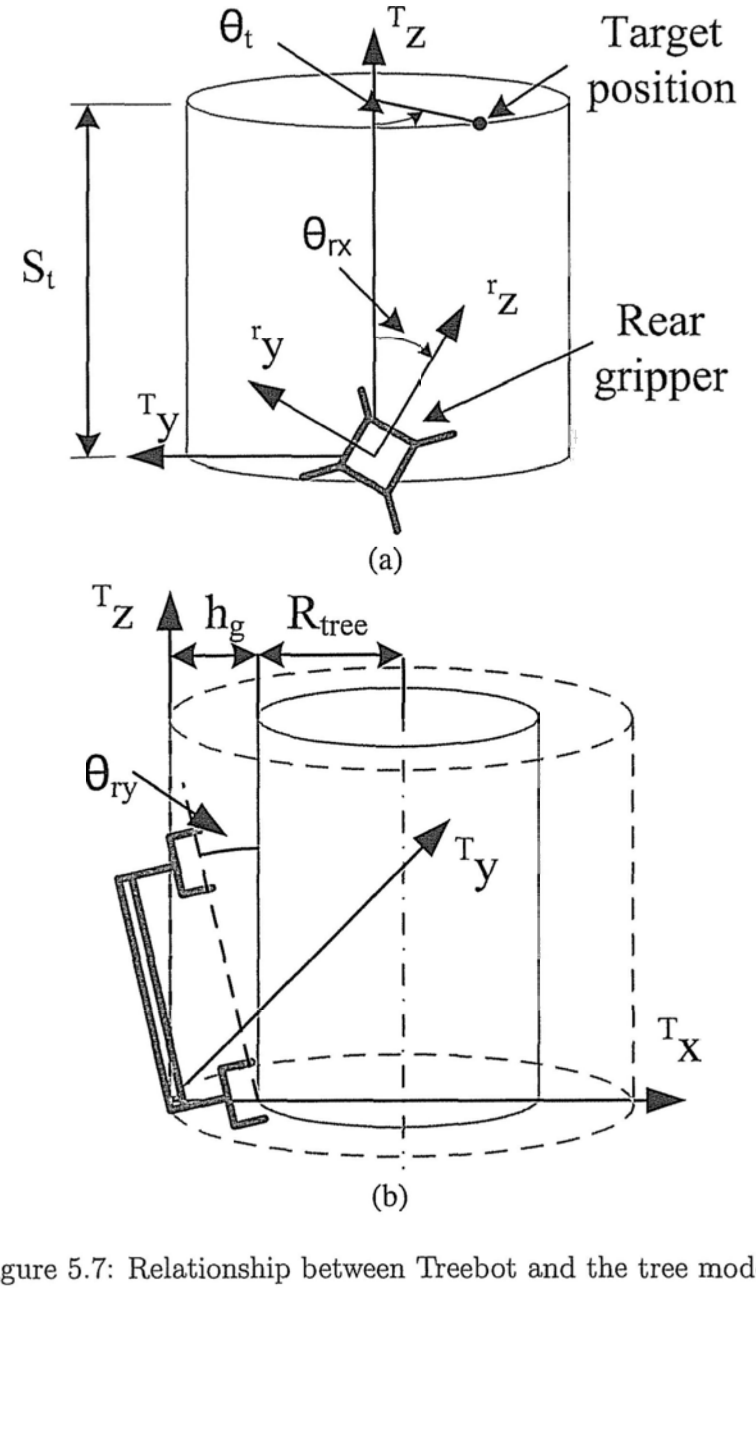


Figure 5.7: Relationship between Treebot and the tree model.

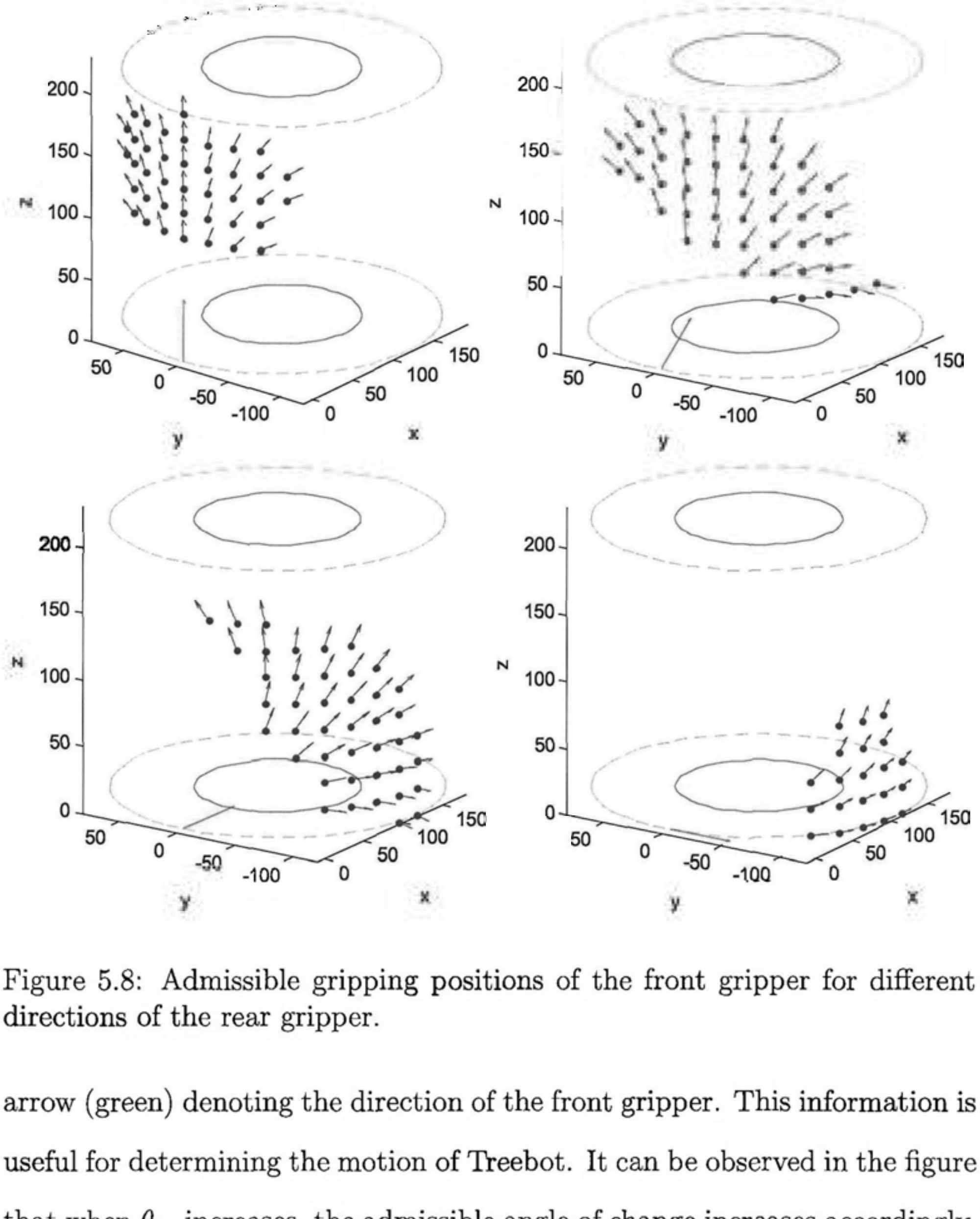


Figure 5.8: Admissible gripping positions of the front gripper for different directions of the rear gripper.

arrow (green) denoting the direction of the front gripper. This information is useful for determining the motion of Treebot. It can be observed in the figure that when θ_{rx} increases, the admissible angle of change increases accordingly.

□ **End of chapter.**

Chapter 6

Autonomous Climbing

The goal of developing of Treebot is to assist or replace human being in performing forestry tasks on trees. A certain level of autonomous climbing ability of Treebot is helpful to reduce the complexity of manipulation by users. An autonomous climbing strategy for Treebot is thus proposed. To determine the motions to climb up autonomously in an unknown environment, a robot must be equipped with sensors that can explore the environment. Vision-based sensors provide rich information about the environment. However, they require a great deal of computational power. Moreover, light conditions vary in outdoor environments, which may affect the accuracy of visual information. There are many living creatures that do not rely on visual information, but can navigate well in their natural environment. Inchworms, for example, navigate on trees by using only their sense of touch. Although the information obtained by tactile sensors is not rich, it is reliable. Furthermore, the processing of tactile information is much simpler than that of visual information. Inspired by arboreal animals, an algorithm is developed to allow Treebot to climb irregularly shaped trees autonomously by using tactile sensors and a tilting sensor only. The development of the algorithm can also reveal how tactile sensors can best be employed in autonomous tree climbing.

The remainder of this chapter is organized as follows. Section 6.1 introduces the structure of the proposed autonomous climbing strategy. Section 6.2 proposes a tree shape approximation method. In Section 6.3, the motion planning strategy is discussed. The experimental results are presented in Section 6.4. Finally, a summary is provided in Section 6.5.

6.1 Autonomous Climbing Strategy

Robots may topple sideways when climbing on an inclined tree. The optimal climbing position to avoid this tendency is above the centerline of the tree, so that gravitational force acts on the robot to direct it to the centerline of tree. [46]. In the following text, "upper apex" is used to describe this optimal position. The autonomous climbing algorithm aims to make Treebot climb a tree along the optimal path. The procedure for the autonomous climbing motion is shown in Fig. 6.1. It is assumed that Treebot is already attached to the tree by the rear gripper, that the front gripper is detached, and that the continuum body is contracted to the minimum length. Completing the main loop of the procedure once is termed as a complete climbing gait cycle. By repeating the climbing gait cycle, Treebot can climb a tree along the optimal path. The following sections discuss this procedure in detail.

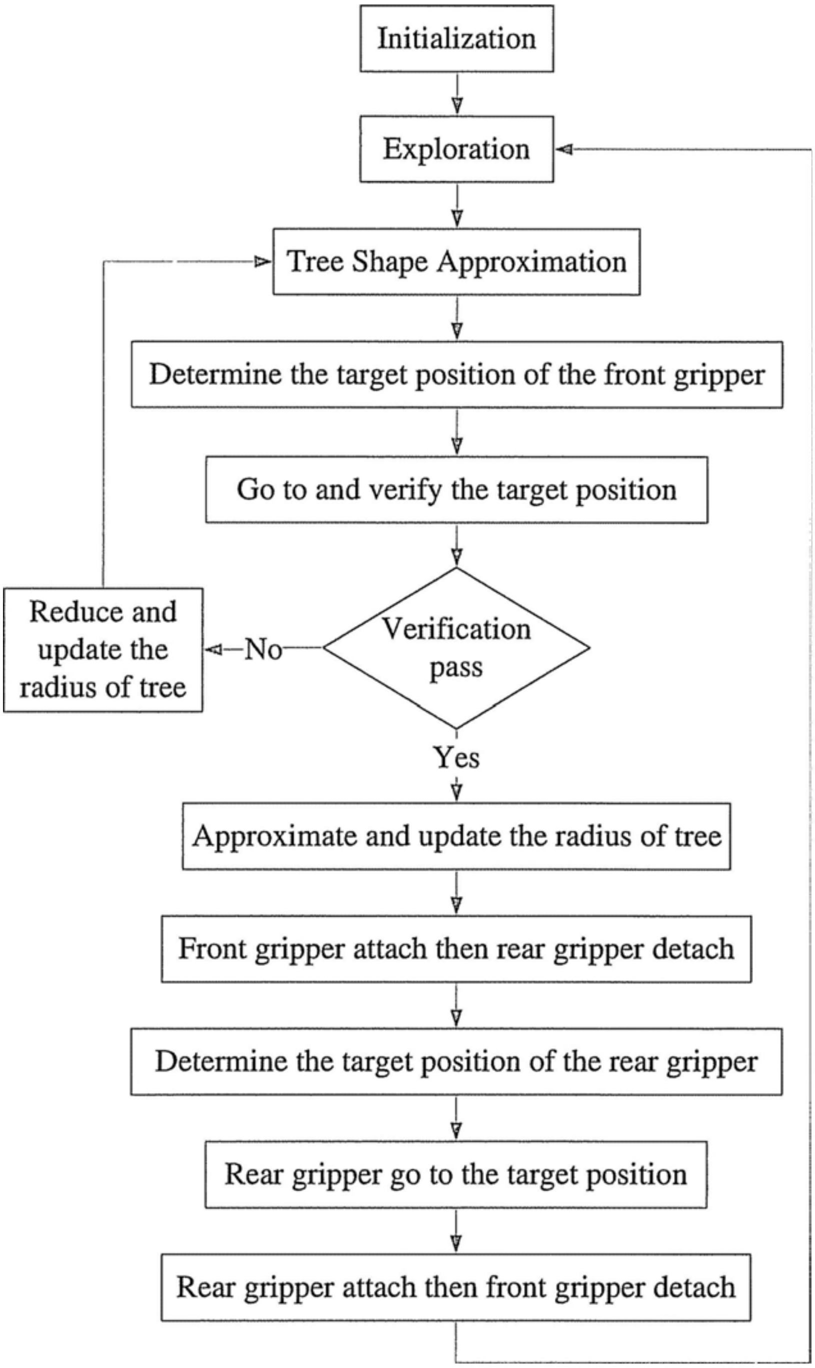

Initialization
Exploration
Tree Shape Approximation
Determine the target position of the front gripper
Go to and verify the target position
Verification pass
No
Reduce and update the radius of tree
Yes
Approximate and update the radius of tree
Front gripper attach then rear gripper detach
Determine the target position of the rear gripper
Rear gripper go to the target position
Rear gripper attach then front gripper detach

Figure 6.1: Flow chart of the autonomous climbing strategy.

6.2 Tree Shape Approximation

The concept of tree shape modeling is mentioned in Chapter 5. This section discusses the method used to approximate the values of the model parameters from information provided by the tactile sensors. Treebot explores the shape of a tree by tentacles attached to the front gripper, and uses the exploration data to approximate the shape of the tree. The exploring motion of Treebot is based on the proposed an exploring strategy. Approximating the shape of the explored portion of the tree is useful to determine the location of the optimal climbing position and to predict the shape of the tree ahead for motion planning. There are many techniques for shape reconstruction using information from tactile sensors. Okamura and Cutkosky [44] proposed a method for extracting the local features of a surface. Jia and Tian [43] reconstructed the unknown local curved surface by using one-dimensional tactile data. Schopfer [45] used a 2D pressure array to reconstruct the unknown shape of an object. All of these methods can successfully reconstruct the unknown shape of an object. Here, as the geometry of a branch is assumed to be a curved cylinder, an efficient reconstruction scheme can be developed based on a known geometric model to speed up the exploration and reconstruction processes.

6.2.1 Exploring Strategy

The proposed exploring strategy aims to trace a growing path of a tree using the front gripper, which is similar to the feature-tracing method presented in [44]. The trajectory of the front gripper can then be used to reconstruct

Table 6.1: Exploring strategy

State	Action
No tentacle is triggered	Bend forward with extension
Both tentacles are triggered	Bend backward with extension
Only left tentacle is triggered	Bend left and backward
Only right tentacle is triggered	Bend right and backward
Length of extension reaches a constant value	Finish exploration

the shape of the tree. The top left and right tentacles attached to the front gripper are used for exploring. The state and action pairs for the exploring motion are listed in Table 6.1. A tentacle acts in a similar fashion to a mechanical switch. It is triggered when a force acts on the bottom part of the tentacle over a certain threshold. The forward and left directions are defined as the positive x^r and y^r directions, respectively.

In the exploring strategy, the front gripper approaches and leaves the tree surface repeatedly. When the front gripper leaves the growing path of a tree, only one side of the tentacle is triggered frequently. The front gripper then moves to eliminate this unbalanced triggering between tentacles so as to keep the front gripper follow the growing path of the tree.

Once a tentacle is triggered, the Cartesian coordinates of the front gripper are recorded, which can be found by (5.3). During the exploring motion, the semi-passive joint is locked to make (5.3) applicable. As only one tactile sensor is installed on each tentacle, there is no way of determining where a force is exactly applied along the tentacles. As a result, the triggering of a tentacle does not necessarily indicate that the center of the front gripper is placed on the tree surface. To obtain accurate data points, the selected points must include only those points at which that both the left and right tentacles are triggered at the same time, or the average position of the points at which the left and right tactile sensors are triggered alternatively.

6.2.2 Arc Fitting

As the shape of a tree is approximated as a perfect cylinder with a uniform bend, the data acquired from exploration are fitted with a 3D arc to help reconstruct the shape of the tree. It is assumed that the arc crosses the first and last data points. As a result, the data are transformed such that the first data point is on the origin and the last data point is on the z-axis (rotatino about the z-axis of $-\theta_z$ and then rotate about the y-axis of $-\theta_y$).

Plane fitting

To simplify the 3D arc fitting problem into 2D, the data are fitted onto a plane, as illustrated in Fig. 6.2(a). This is accomplished by determining the optimal angle of rotation about the z-axis θ_{plane} to minimize the x component value of the transformed data. Let $[x_i, y_i, z_i]$ be the transformed points, θ_{plane} can than be obtained by minimizing:

$$x_i \cos\theta_{plane} - y_i \sin\theta_{plane} \tag{6.1}$$

By using the least square method and considering all of the points:

$$\begin{aligned}\sum \tfrac{d}{d\theta_{plane}}(x_i \cos\theta_{plane} - y_i \sin\theta_{plane})^2 &= 0 \\ \Rightarrow \tan\theta_{plane} &= \tfrac{-o\pm\sqrt{o^2+4}}{2}\end{aligned} \tag{6.2}$$

where $o = \frac{\sum {y_i}^2 - \sum {x_i}^2}{\sum x_i y_i}$.

Hence,

$$\theta_{plane} = \tan^{-1}\left(\frac{-o \pm \sqrt{o^2+4}}{2}\right) \tag{6.3}$$

2D arc fitting

When the fitted plane has been obtained, the data are converted into 2D by projection onto the plane. Then, referring to Fig. 6.2(b), the center of the approximated arc (y_c, z_c), the curvature of the bend κ_{arc}, and the angle of the arcθ_{arc} can be found by using a 2D arc fitting method. Let the number of points be η. Let (y_η, z_η) be the end point. (y_c, z_c) and r be the center and radius of the approximated arc respectively.

To cross the start and end points, the approximated arc should fulfill the following equations:

$${y_c}^2 + {z_c}^2 = r^2 \tag{6.4}$$

$$(y_\eta - y_c)^2 + (z_\eta - z_c)^2 = r^2 \tag{6.5}$$

Combining (6.4) and (6.5):

$$z_c = a - by_c \tag{6.6}$$

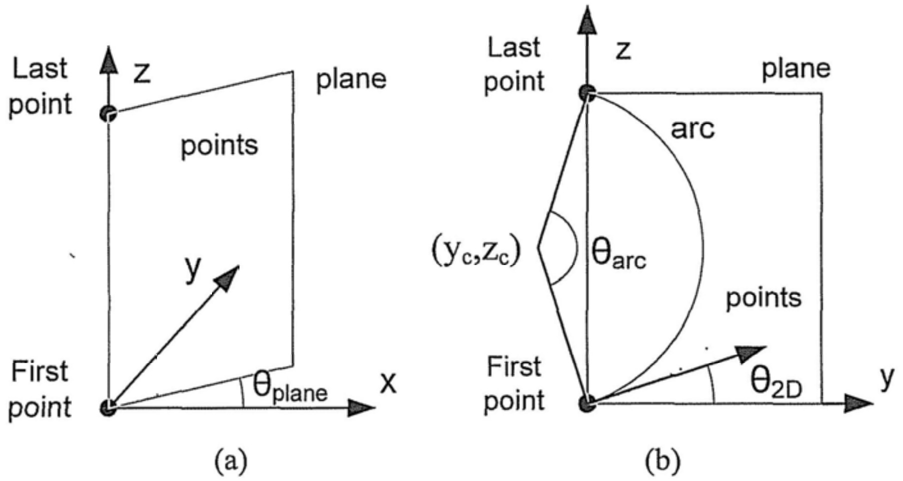


Figure 6.2: Notations for the arc fitting. (a) 3D plane fitting. (b) 2D arc fitting.

where $a = \frac{\left(y_\eta{}^2+z_\eta{}^2\right)}{2z_\eta}$ and $b = \frac{y_\eta}{z_\eta}$.

The distance error $e_\imath$ of a point to the approximated arc can be found by:

$$e_i = \left(y_i - y_c\right)^2 + \left(z_i - z_c\right)^2 - r^2 \tag{6.7}$$

Sub. (6.4) into (6.6):

$$\begin{aligned} e_i &= \left(x_\imath - x_c\right)^2 + \left(z_i - z_c\right)^2 - \left({y_c}^2 + {z_c}^2\right) \\ &= -\left({y_i}^2 + {z_i}^2\right) - 2y_iy_c - 2z_iz_c \end{aligned} \tag{6.8}$$

Sub. (6.6) into (6.8):

$$\begin{aligned} e_i &= -\left({y_i}^2 + {z_i}^2\right) - 2z_ia + 2z_iby_c - 2y_iy_c \\ &= m_i + n_iy_c \end{aligned} \tag{6.9}$$

where $m_i = -\left({y_i}^2 + {z_i}^2\right) - 2az_i$ and $n_i = 2\left(bz_i - y_i\right)$.

By using the least square method and considering all of the points, that is,

$$\begin{aligned} \sum \frac{d}{dy_c}{e_i}^2 = \sum \frac{d}{dy_c}\left(m_i + n_i y_c\right)^2 = 0 \\ \Rightarrow y_c = -\frac{\sum m_i}{\sum n_i} \end{aligned} \tag{6.10}$$

Sub. (6.10) into (6.6):

$$z_c = a + b\frac{\sum m_i}{\sum n_i} \tag{6.11}$$

In summary, the parameters of the fitted arc can be obtained by:

$$\begin{aligned} \left(y_c, z_c\right) &= \left(-\sum m_i \Big/ \sum n_i, a + b\sum m_i \Big/ \sum n_i\right) && (6.12) \\ \kappa_{arc} &= 1 \Big/ \sqrt{{y_c}^2 + {z_c}^2} && (6.13) \\ \theta_{arc} &= 2\cos^{-1}\left(y_c \kappa_{arc}\right) && (6.14) \end{aligned}$$

where $m_i = -\left({y_i}^2 + {z_i}^2\right) - 2az_i$, $n_i = 2bz_i - 2y_i$, $a = \frac{{y_m}^2 + {z_m}^2}{2z_m}$ and $b = y_m/z_m$.

Finally, the tangent vector $\vec{v}_S$ and the bending direction $\vec{v}_{bend}$ (toward the center of the bend) of the arc at the starting point in the rear gripper frame can be determined by:

$$\vec{v}_S = Rot_z(\theta_z)\, Rot_y(\theta_y)\, Rot_z(-\theta_{plane}) \begin{bmatrix} 0 \\ \cos\theta_{2D} \\ \sin\theta_{2D} \end{bmatrix} \tag{6.15}$$

$$\vec{v}_{bend} = Rot_z(\theta_z)\, Rot_y(\theta_y)\, Rot_z(-\theta_{plane}) \begin{bmatrix} 0 \\ -\sin\theta_{2D} \\ \cos\theta_{2D} \end{bmatrix} \tag{6.16}$$

where $\theta_{2D} = \tan^{-1}\frac{z_c}{y_c} + sign\,(y_c)\,\frac{\pi}{2}$.

6.2.3 Tree Shape Reconstruction

To approximate the parameters of the tree model, the fitted arc is transformed into the tree frame, that is, to transform the tangent vector of the arc on the ${}^{T}z$-axis:

$${}^{T}\vec{v}_S = Rot_x(\theta_{rx})\, Rot_y(\theta_{ry})\, \vec{v}_S \tag{6.17}$$

$${}^{T}\vec{v}_{bend} = Rot_x(\theta_{rx})\, Rot_y(\theta_{ry})\, \vec{v}_{bend} \tag{6.18}$$

where $\theta_{ry} = \sin^{-1}\left({}^{T}v_{S\_x}\right)$, $\theta_{rx} = -\tan^{-1}\frac{{}^{T}v_{S\_y}}{{}^{T}v_{S\_z}}$,
${}^{T}\vec{v}_S = \begin{bmatrix} {}^{T}v_{S\_x} & {}^{T}v_{S\_y} & {}^{T}v_{S\_z} \end{bmatrix}$
and ${}^{T}\vec{v}_{bend} = \begin{bmatrix} {}^{T}v_{bend\_x} & {}^{T}v_{bend\_y} & {}^{T}v_{bend\_z} \end{bmatrix}$.

In addition, the bending direction of the fitted arc in the tree frame is

$$\phi_{arc} = \tan^{-1}\frac{{}^{T}v_{bend\_y}}{{}^{T}v_{bend\_x}} \tag{6.19}$$

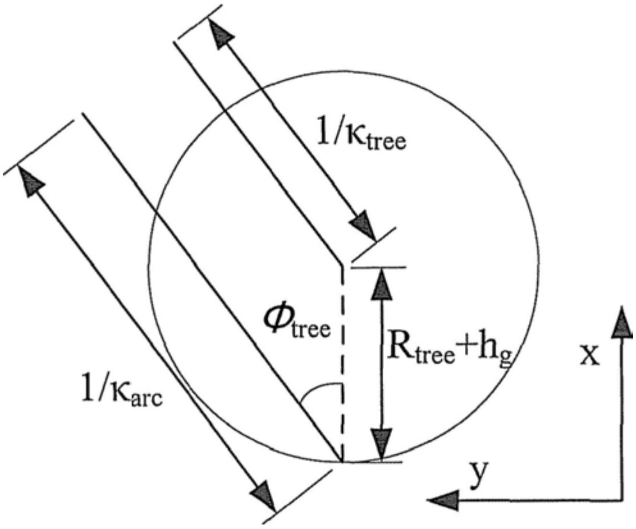


Figure 6.3: Tree shape approximation by the fitted arc.

According to Fig. 6.3, by giving the radius of a tree R_{tree}, the values of the parameters of the tree model, i.e., ϕ_{tree}, κ_{tree} and S_{tree} can be determined as:

$$\phi_{tree} = \phi_{arc} \tag{6.20}$$

$$\kappa_{tree} = 1/(1/\kappa_{arc} - (h_g + R_{tree}) \cos \phi_{tree}) \tag{6.21}$$

$$S_{tree} = \theta_{arc}/\kappa_{tree} \tag{6.22}$$

6.2.4 Tree Radius Approximation

The data from the exploring motion can be used to approximate the shape of tree but not the radius of tree. Hence, two methods are proposed to approximate the radius.

The first method is based on the unsuccessful placement of the front gripper in the target position, as this indicates that the actual radius of the tree must be smaller than the approximated radius. The approximated radius of the tree is then reduced by a certain value, and then the maximum angle of change is recalculated. This trial process is repeated until the front gripper is successfully placed in the target position.

The second method for approximating the radius of a tree is involves comparing the angle of change to the upper apex ($\theta_{optimal}$) for different positions of the front gripper. The details of the procedure for finding the angle of change to the upper apex of the tree will be discussed in Section 6.3. As obtaining $\theta_{optimal}$ does not required the radius of the tree to be known, the new approximated radius of the tree R'_{tree} can be obtained by:

$$R'_{tree} = R_{tree} \left|\theta_{optimal}/\theta'_{optimal}\right| \tag{6.23}$$

where $\theta'_{optimal}$ is the differencc between $\theta_{optimal}$ obtained by two different positions of the front gripper and R_{tree} is the last approximated radius of the tree. Distinct from the first method, this method updates the information of the radius of the tree at every climbing gait cycle. However, the application of this method is not feasible when the inclined angle of tree is $\pi/2$, because in this state the slope in any position is minimum. In such case, the radius of tree can be approximated by using the first method.

6.3 Motion Planning

6.3.1 Angle of Change to the Upper Apex

To find the upper apex of a tree, the direction of gravity must first be established. The gravity vector can be obtained from the tilting sensor attached to the front gripper. As the tilting sensor is fixed to the front gripper, the coordinates with respect to the rear gripper frame can be determined by the posture of Treebot, as discussed in Chapter 5. Once the tree shape has been approximated, the transformation relationship between the rear gripper frame and the tree frame (θ_{rx} and θ_{ry})can be obtained. Hence, the gravity vector can be represented in the coordinate frame of the end of the approximated tree shape as illustrated in Fig. 6.4. The angle of change required to reach the upper apex $\theta_{optimal}$ is equivalent to the angle of rotation about ${}^{TE}z$-axis required to make the gravity vector $\vec{v}_{gravity}$ lies on the ${}^{TE}z - {}^{TE}x$ plane with a positive x component, i.e.,

$$\theta_{optimal} = \tan^{-1}\left(v_{gravity\_y}/v_{gravity\_x}\right) \tag{6.24}$$

where $\vec{v}_{gravity} = \begin{bmatrix} v_{gravity\_x} & v_{gravity\_y} & v_{gravity\_z} \end{bmatrix}$.

In addition, the inclined angle of the tree can be obtained by:

$$\varphi_{incline} = \sin^{-1}\left(|v_{gravity\_z}|\right) \in [0, \pi/2] \tag{6.25}$$

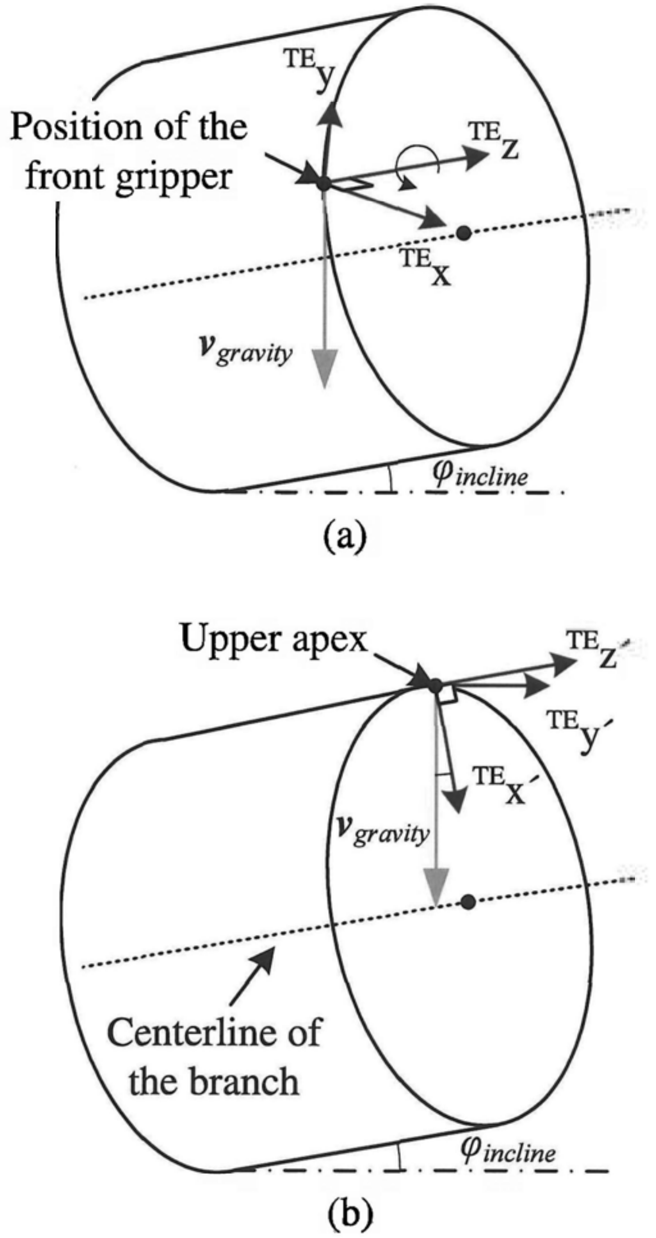


Figure 6.4: The concept of finding an optimal angle of change.

6.3.2 Motion Planning Strategy

The optimal solution for making Treebot follow the optimal path is to place the front gripper on and at the same time set the direction of the gripper parallel to the optimal path. As Treebot is a nonholonomic system [28] that the direction of the gripper and the position of the gripper are coupled, it is impossible to achieve this solution in one gait cycle. However, it can be achieved in two climbing gaits cycles. In the first climbing gait cycle the direction of the rear gripper is adjusted, and in the second the front gripper is set on the optimal path with target direction. Fig. 6.5 illustrates the concept of achieving the target position and direction in two climbing gait cycles. In the figure, the circle denotes the target position and the arrow represents the target direction. Rectangles colored in white and grey represents the attached and detached grippers respectively. After an exploration (a), Treebot acquires the optimal position and direction for the front gripper. The continuum body then contracts (b) and adjusts the direction of the rear gripper (c). Finally, the front gripper moves to the target position in the appropriate direction along the path (d). The forward motion is completed when the continuum body contracts to pull up the rear gripper (e).

It can be seen that it takes four motion steps to move forward, which is quite time-consuming. In view of that, a more efficient motion planning strategy is thus proposed, as illustrated in Fig. 6.6. After exploration (a), the front gripper moves directly to the target position by neglecting the target direction (b). The continuum body then contracts and adjusts the direction of the rear gripper such that the front gripper can move to the next target position and direction (marked as a dotted circle and arrow respectively)

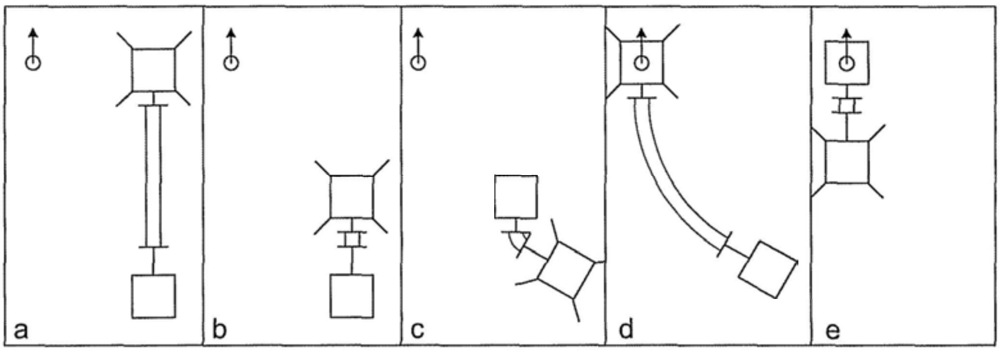


Figure 6.5: Series of motions required to place the front gripper to the target position and direction.

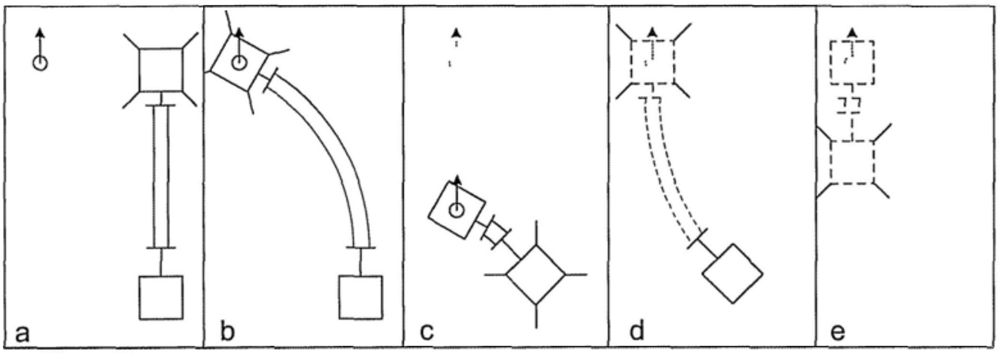


Figure 6.6: A more efficient strategy to increase the climbing speed.

in the next climbing gait cycle (c), (d), (e). The next target position and direction are approximated from the current information. This scheme also requires four motion steps to place the front gripper in the future target position and direction, but the robot moves forward twice in the process, allowing it to climb much faster. However, the drawback of this method is that it may not exactly go to the target position and direction due to the inaccurate estimation of the future target position and direction.

Target position of the front gripper

As has been stated, the optimal solution to ensure that Treebot follows the best path is to place the front gripper directly on the optimal path. However, it is necessary to consider that when the inclined angle is large (nearly vertical), a change of position will not reduce much of the pull-out force generated by gravity. As a result, to avoid Treebot having to make a large change in angle to reduce the pull-out force by only a small amount, the magnitude of the angle of change should decrease when the inclined angle of the tree is large. The target angle of change is defined as:

$$\theta_t = \theta_{optimal}\left[1 - \frac{\varphi_{incline}}{\pi/2}\right] \tag{6.26}$$

If θ_t exceeds the admissible angle of change, then it is replaced by the admissible angle of change that is closest to θ_t. Once the target position $\vec{P_t}$ is defined, the posture of the continuum body to place the front gripper can be obtained by (5.4).

Target position of the rear gripper

To determine the target position of the rear gripper, it is necessary to approximate the future target position and direction of the front gripper. The future length from the current rear gripper position S'_{tree} is $S'_{tree} = S_{tree} + S_{explore}$, where S_{tree} is the length of the current approximated segment of the tree and $S_{explore}$ is the approximated future length to be explored. It is assumed that $S_{explore} = S_{tree} - l_f - l_r$. The future optimal position can then be obtained by the same method of finding the target optimal position for the placement of the front gripper.

Once the future target position and direction of the front gripper have been obtained, it is necessary to find the position of the continuum body that the front gripper is placed on the current target position. The position in the tree frame can be determined by:

$$^{T}\vec{p}_b = {}^{T}\vec{p}_t - l_f {}^{T}\vec{v}_f \tag{6.27}$$

where $^{T}\vec{v}_f$ is the direction of the front gripper in the current target position.

To determine the posture of the continuum body from the future target position and direction of the front gripper to place the rear gripper to the target position of the continuum body, it is first transform $^{T}\vec{p}_b$ to the future front gripper frame $^{ff}\vec{p}_b$, then find the posture of the continuum body (θ_f and ϕ_f) to place its rear part to the $^{ff}\vec{p}_b$ by (5.6).

The direction of the rear gripper in the future front gripper frame $^{ff}\vec{v}_r$

can be found by:

$$^{ff}\vec{v}_r = Rot_z\left(\phi_f\right) Rot_y\left(-\theta_f\right) Rot_z\left(-\phi_f\right) \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{6.28}$$

The angle between the direction of the rear gripper and the growing direction of the tree in the target position γ_r can then be determined by transforming $^{ff}\vec{v}_r$ to the coordinate frame of the target position. To make the exploring motion easy to implement, γ_r should be as small as possible. As a result γ_r is bounded in $\pm\pi/4$.

Finally, the posture of the continuum body to place the rear gripper in the appropriate direction from the target position is determined by,

$$\theta_r = |\gamma_f - \gamma_r| \tag{6.29}$$

$$\phi_r = \frac{\gamma_f - \gamma_r}{|\gamma_f - \gamma_r|}\frac{\pi}{2} \tag{6.30}$$

where γ_f denotes the angle between the direction of the front gripper and the direction of growth of the tree in the target position.

The length of the virtual tendon in placing the rear gripper S_r is defined by avoiding the tendons with a negative length. Eq. 5.1 can be reformulated as:

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} S_r - d\theta_r \sin\left(\phi_r - \pi/2\right) \\ S_r + d\theta_r \sin\left(\phi_r - \pi/6\right) \\ S_r - d\theta_r \cos\left(\phi_r - \pi/3\right) \end{bmatrix} = S_r - \begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \\ \hat{s}_3 \end{bmatrix} \tag{6.31}$$

Hence, to avoid the length of each tendon less than zero, S_r should be defined as:

$$S_r = \max(\hat{s}_1, \hat{s}_2, \hat{s}_3) \tag{6.32}$$

6.3.3 Verification of Target Position

The gripper may not be able to appress to the target position, which may result in an inaccurate approximation or change in the radius of the tree. The tentacle signals can be used to detect whether the gripper is appressed to the surface of tree. The gripper is regarded as appressed when any two tentacles are triggered diagonally.

When the gripper is on the target position, Treebot will try to appress the gripper to the tree surface. The semi-passive joint will first unlock so that it can be rotated freely. The gripper then pushes forward into the tree surface a certain distance to try to appress the gripper to the tree surface. If it cannot be appressed to the surface, the target position is inadmissible. In this case, the approximated radius of the tree will be reduced and the target position recalculated. The process is then repeated until an appressed placement is achieved.

6.4 Experiments and Results

Numerous experiments have been carried out to evaluate the proposed autonomous climbing algorithm in terms of tree shape approximation, optimal path following, and climbing a tree with branches.

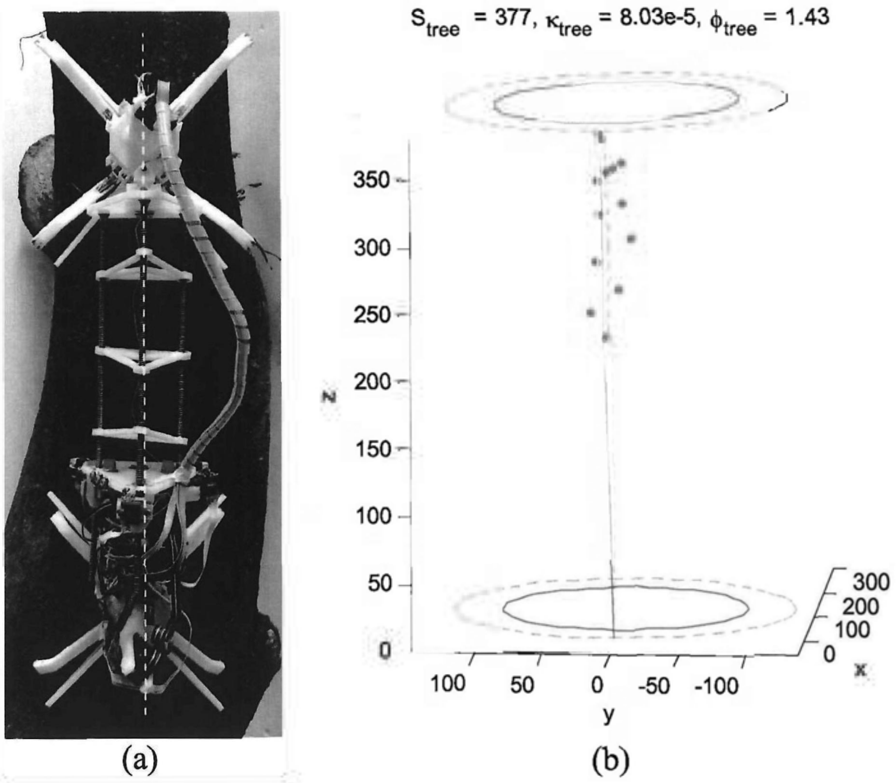
6.4.1 Tree Shape Approximation

The results of three experiments to test tree shape approximation, Test 1, Test 2 and Test 3, are shown in Fig. 6.7, 6.8 and 6.9 respectively. The subfigures (a) show the approximation target and the final exploring posture of Treebot, and the subfigures (b) illustrate the approximation result. In the subfigures (b), the solid arc (blue) denotes the posture of Treebot, the dots (green) are the exploration data, the dotted arc (red) represents the fitted arc and the large circles (blue) represent the circumference of the ends of the segment of tree. The parameters for the approximated shape of the tree are listed at the top of the figures. The dashed line (yellow) in the subfigures (a) denotes the centerline of the tree.

In Test 1, the explored tree segment is straight, and the direction of the rear gripper is parallel to the direction of growth of the tree. In the approximation result, the shape of tree is almost straight which approximate the actual shape of tree correctly.

The setting of Test 2 is the same as that of Test 1, except that the direction of the rear gripper is not parallel to the direction of growth of the tree. It can be seen that although the posture of Treebot does not match the shape of the tree, the tree shape can still be correctly approximated. This property is significant for tackling trees with varying shape.

In Test 3, the explored tree segment is bent leftward and backward. It can be observed that the approximated shape of the tree is also bent in a similar fashion to the actual shape of the tree (the approximated bending direction is 2.24 rad). This demonstrates that the algorithm can successfully approximate a bent tree in 3D.

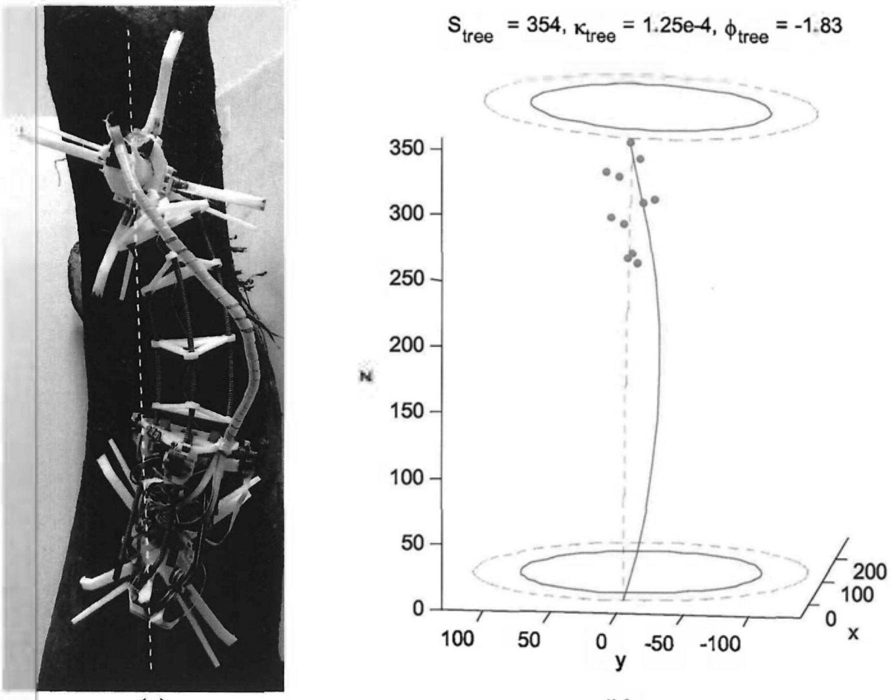


$S_{tree} = 377$, $\kappa_{tree} = 8.03e\text{-}5$, $\phi_{tree} = 1.43$

(a)

(b)

Figure 6.7: Test 1: Tree shape approximation on a straight tree (a) Approximation target and final exploring posture of Treebot; (b) Approximation result.



(a) (b)

Figure 6.8: Test 2: Tree shape approximation on a straight tree (a) Approximation target and final exploring posture of Treebot; (b) Approximation result.

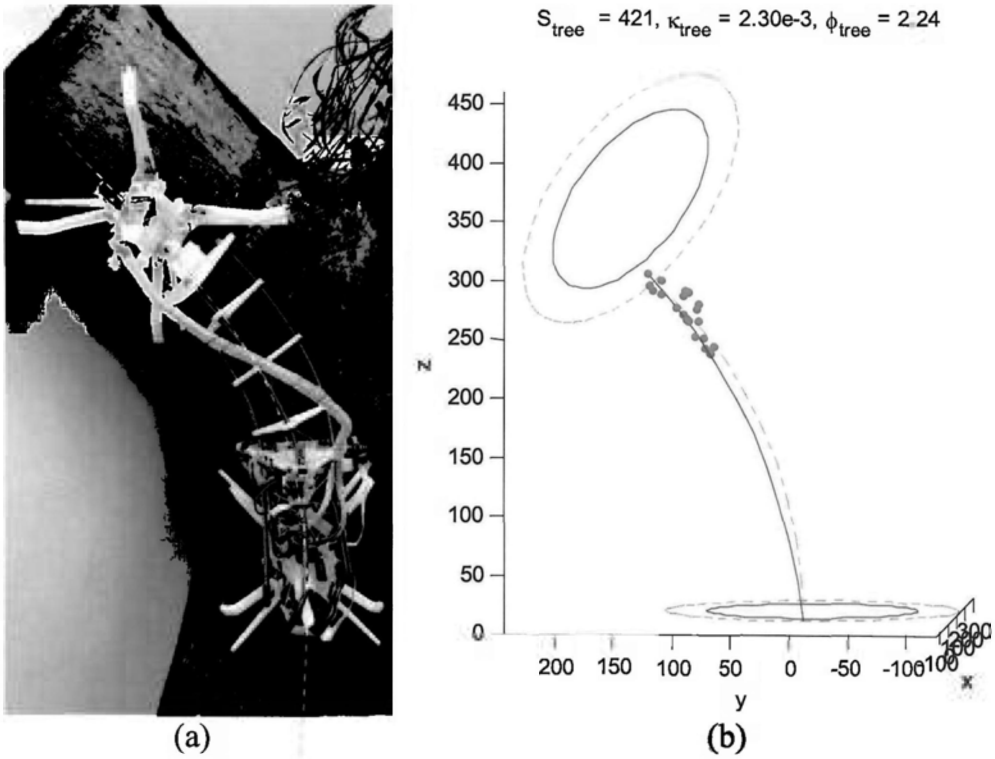


Figure 6.9: Test 3: Tree shape approximation on a curved tree (a) Approximation target and final exploring posture of Treebot; (b) Approximation result.

6.4.2 Optimal Path Following

The proposed autonomous climbing algorithm can guide Treebot to follow an optimal path. An experiment was conducted to evaluate its performance in this respect in which Treebot was commanded to climb a tree with a 70-degree inclined angle. Fig. 6.10 shows the exploring and climbing motions. The optimal path is marked as dashed blue line in the figures. Initially, Treebot was not on the optimal path and climbed according to the approximated shape of the tree only. After two climbing gait cycles, the whole body of Treebot was successfully placed on the optimal path with motions similar to those proposed in the motion-planning strategy illustrated in Fig. 6.6. This result indicates that the proposed autonomous climbing algorithm can successfully guide Treebot to follow the optimal climbing path.

6.4.3 Climbing a Tree with Branches

An experiment has been conducted to evaluate the motion of Treebot on a tree with branches. In the first test, the initial position of Treebot is shown in Fig. 6.11(a). Treebot selected branch A to climb, as shown in Fig. 6.11(b). In the second test, the initial position of Treebot was shifted a little bit to the left, as shown in Fig. 6.11(c). This time, branch B was selected by the exploring motion (Fig. 6.11(d)). These results indicate that Treebot tends to choose the closest branch, and thus the selection of a branch is determined by the position of Treebot.

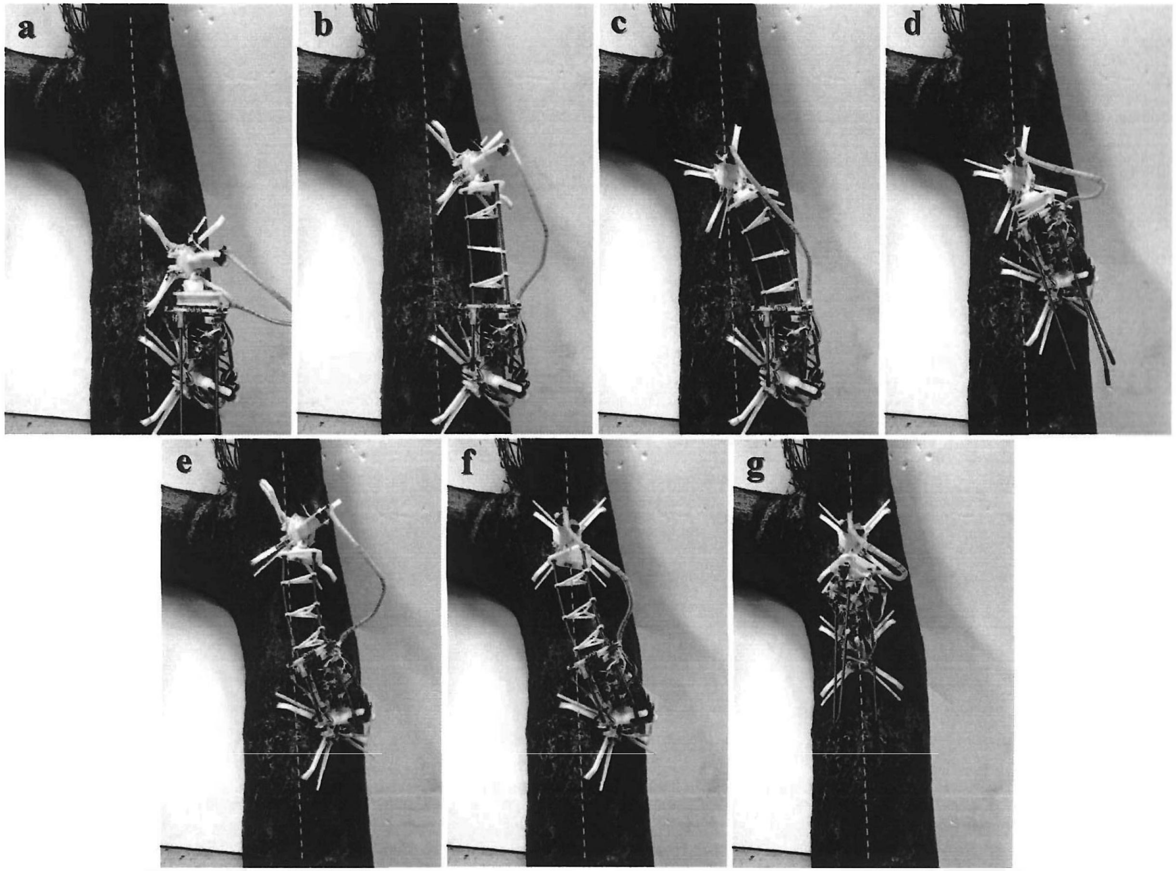


Figure 6.10: Experiment for optimal path following. The dashed line denotes the upper apex of the tree. (a) Initial position; (b), (e) Exploring motion; (c), (f) Front gripper gripping; (d), (g) Rear gripper gripping.

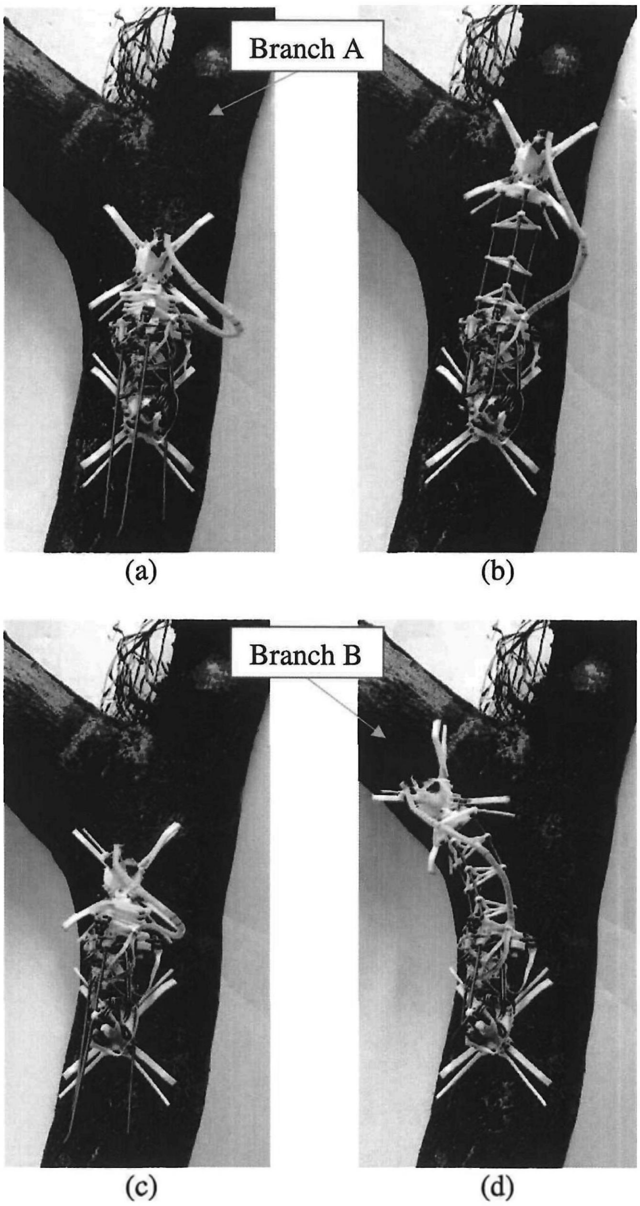


Figure 6.11: Experiment for climbing a tree with branches. (a) Initial position in the first test; (b) Exploring posture in the first test; (c) Initial position in the second test; (d) Exploring posture in the second test.

6.5 Summary

This chapter presents the development of an autonomous tree climbing algorithm that enables Treebot to explore and climb autonomously on an irregularly shaped tree. The algorithm includes a tree shape approximation method and a motion planning strategy. The results of experiments to evaluate the proposed algorithm reveal that it can adequately approximate the shape of a tree. The proposed motion planning strategy also succeeds in guiding Treebot to follow the optimal path to climb the tree.

In the proposed autonomous climbing algorithm, the selection of a branch is determined passively by the position of Treebot. If Treebot does not move to the desired branch, then manual control is needed to guide it there.

□ **End of chapter.**

Chapter 7

Global Path and Motion Planning

The global motion planning problem for tree climbing is challenging, as trees have an irregular and complex shape. To the best of the author's knowledge, there is no related study that focuses on the global motion planning problem for tree climbing robot. Aracil [8] proposed a motion planning method to allow the Parallel Climbing Robot to climb a trunk. However, this work merely discussed the local motion planning problem according to local information. There are many motion planning approaches for climbing in manmade structures, such as walls and glass windows [9, 10, 29]. However, these structures are different from trees, and the approaches are thus not suitable for tree climbing problems.

This chapter proposes an efficient global motion planning algorithm for tree climbing. In the conventional motion planning approach [30], the configuration space (a set of possible transformations that could be applied to the robot [31]) of the problem must be constructed to help solve the problem. However, the formulation of the configuration space is complex as it involves complicated interactions between the environment and the kinematics of the robot. A robot with high degrees of freedom (DOF) and continuous motion requires a high dimensional and huge configuration space, which makes the problem difficult to solve.

Here, the planning problem is solved more efficiently by dividing it into a path planning and a motion planning problem, which are solved separately to reduce the dimensions of the problem space. In the path planning sub-problem, it is assumed that Treebot is of point size and holonomic, such that its kinematics can be ignored. The aim is to find an optimal path to reach the target position on a 2D manifold. The path planning algorithm includes several constrains to make the path easy for Treebot to follow. As it only considers the 2D manifold of the tree surface, the state space has relatively few dimensions.

An intuitive method is developed to represent the climbing space. It highly simplifies the path planning problem in terms of linear-time complexity. A dynamic programming (DP) algorithm is adopted to find the optimal path according to the specified constrains and requirements. The motion planning sub-problem aims to find an appropriate motion for Treebot that allows it to follow the planned path. An effective strategy for motion planning is proposed, the solution to which can be obtained without any searching effort or additional state space formulation.

The remainder of this chapter is organized as follows. The method of state space formulation is presented in Section 7.1. The path planning algorithm is discussed in Section 7.2 while the motion planning algorithm is presented in Section 7.3. Experimental results are presented in Section 7.4. Finally, summary is presented in Section 7.5.

7.1 State Space Formulation

Before working on the path planning problem, the state space to the problem must be formulated. A tree is composed of a trunk and a number of branches. In the proposed algorithm, a trunk is also treated as a branch. It is assumed that the relationship among the branches can be represented by a tree data structure as illustrated in Fig. 7.1. To climb to a target position, a unique sequence of branches must be passed. For example, if the target position is at branch 8 and the initial position is at branch 1, then there is only one way to go: branch 1 → branch 4 → branch 8. This sequence can be easily obtained by using the backward search method in the tree data structure. This means that in path planning, the climbing space of the other non-climbed branches can be ignored, although these branches do need to be considered as obstacles.

The tree surface is discretized by various numbers of points to represent the climbing surface of each branch. The shape of the tree is first decomposed into a number of rings, as shown in Fig. 7.2. The normal direction of a ring is equal to the growth direction of a shape of branch. The distance between each ring takes a certain value such that the rings do not intersect. The shape of each ring is defined by the outer shape of the specified position of the branch, and thus is not necessarily a perfect circle. Finally, each ring is

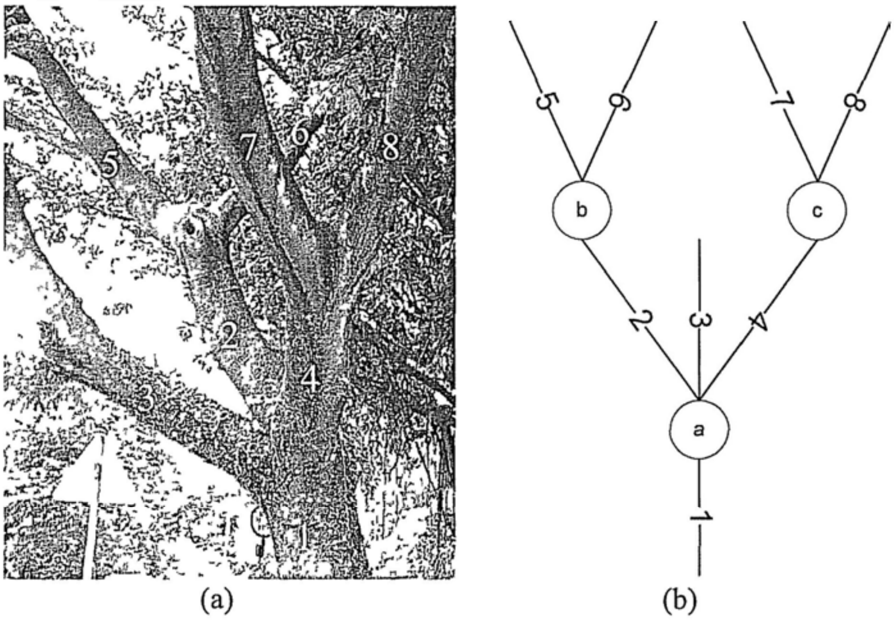


Figure 7.1: Representation of the relationship among branches by using a tree data structure (a) Real tree structure; (b) Branch relationship as represented by the tree data structure.

equally discretized by a certain number of points. Each point contains the information on the 3D Cartesian coordinates and the normal vector of the surface of that point.

The state space to the problem can be are arranged in a matrix form with m rows and n columns when a target position is given, as shown in Fig. 7.3. It is composed of the state spaces of the passing branches in sequence.

There are two situations in which Treebot cannot reach a point. The first is when the upper space of a position is not sufficiently large for the robot to pass through, which may occur when the upper space is occupied by other branches. The second is when the gripping surface of a point is concave such that the gripper cannot grip the surface tightly. The state space contains information on such unreachable points.

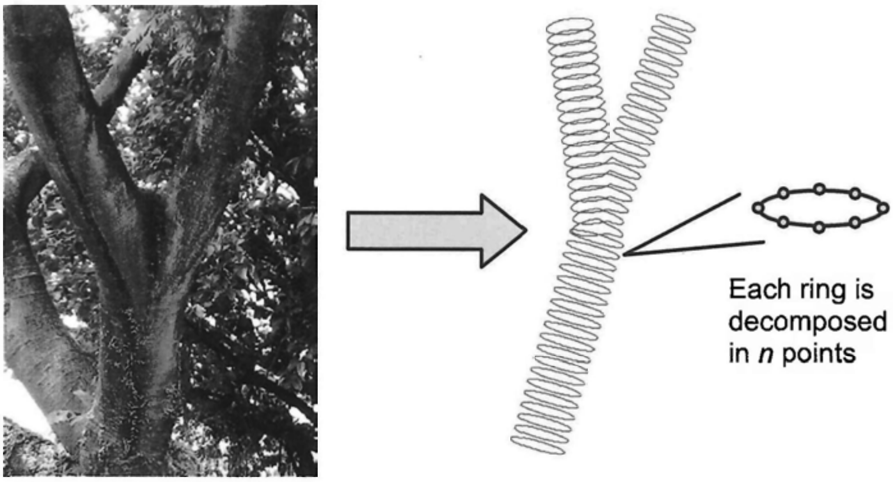


Figure 7.2: Tree surface discretization method.

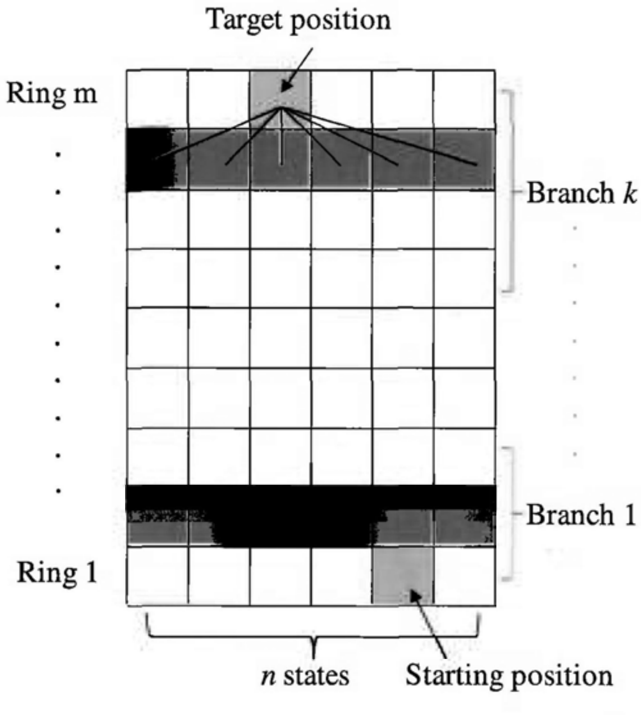


Figure 7.3: State space representation to the problem.

Information on the shape of the tree can be obtained by several means, such as laser- or vision-based sensing [33,34]. However, as this work focuses on the planning problem, it is assumed that the shape of tree is given, and the details of the sensing and state space conversion problems are not discussed.

7.2 Path Planning

Going to a target position and avoiding obstacles are the basic requirements of path planning. In addition, to make a planned path that is easy for Treebot to follow, the planned path should fulfill certain additional requirements. To eliminate the pull out force generated by gravity, Treebot should climb on a upper apex as mentioned in Chapter 6. Furthermore, a shorter path will reduce the robot's energy consumption, and a smoother path will be easier for it to follow. The path should thus be optimized to 1) go directly to the target position, 2) minimize the climbing distance, 3) follow the upper apex of the climbing surface, and 4) avoid obstacles.

7.2.1 Dynamic Programming

Dynamic programming (DP) is an efficient algorithm with a proven ability to find globally optimal solutions to a problem [34]. It works well for discrete states that are difficult to search exhaustively. As a result, the DP algorithm is adopted for the path planning problem. The first step in applying DP is to represent the problem in a DP formulation, that is, to identify the **state**, **action**, **action value**, and the **state value** of the problem.

State $S_{i,j}$: The states of the problem are the discrete points defined in

Section 7.1. A state is denoted as $S_{i,j}$ where i and j denotes the row and column of the workspace respectively. The first row represents the starting ring (ring 1) and the last row represents the ring that contains the target position (ring m). The elements in each row represent the points in that ring.

Action $S_{i,j} \rightarrow S_{i+1,k}$: It is assumed that the target position will not be located on the starting ring, and thus no repeat movement will occur on a ring. Movement can only occur to points on the next ring. This assumption is reasonable, as climbing motions rarely require moving laterally without moving up or down. This assumption significantly reduces the search space of the problem.

Action value $Q\left(S_{i,j}, S_{i+1,k}\right)$: The action value is defined as the sum of the reward values:

$$Q\left(S_{i,j}, S_{i+1,k}\right) = -D(S_{i,j}, S_{i+1,k}) + a_0 G_{i+1,k} + O_{i+1,k} \tag{7.1}$$

where $D(S_{i,j}, S_{i+1,k})$ represents the Euclidean distance between $S_{i,j}$ and $S_{i+1,k}$. $O_{i,j}$ is the obstacle value. The value is taken as zero if there is no obstacle and $-\infty$ if an obstacle is present. An obstacle means an unreachable point, as defined in Section 7.1. a_0 is a positive scalar value to adjust the weight of $G_{i,j}$ in (7.1). $G_{i,j}$ relates to the amount of the pull out force generated by the gravity at that point. Fig. 7.4 shows that the pull out force is directly proportional to the z component of the normalized surface normal vector $z_{i,j}$. Hence, the value of $G_{i,j}$ is defined as:

$$G_{i,j} = z_{i,j} - 1 \tag{7.2}$$

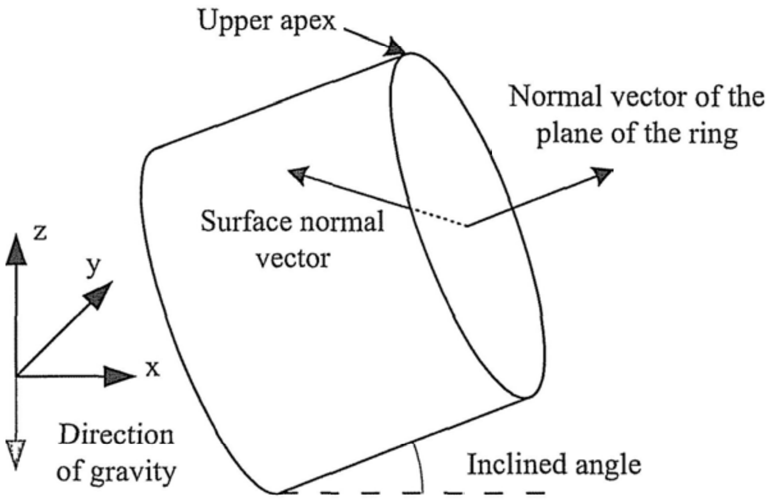


Figure 7.4: Coordinates and notations for the shape of the tree and the gravity vector.

where $n_{i,j} \in [-2, 0]$.

State value $V_{i,j}$: Given a target position and the reward values, the state value of each state can be defined. The state value of row m-1 is simply the distance to the target position, that is,

$$V_{m-1,j} = Q(S_{m-1,j}, S_{m,t}) \tag{7.3}$$

where $S_{m,t}$ denotes the target state.

The state value of the other states can be found by:

$$V_{i,j} = \max\left(V_{i+1,k} + Q\left(S_{i,j}, S_{i+1,k}\right)\right) \tag{7 4}$$

where $k \in [1, n]$ and $i \in [1, m-1]$.

The next possible states for each state are the points in the next row. As a result, using DP, the computational complexity is only O(mn2). In fact, the value n is a problem independent value that does not change with the height of the tree. Thus, the computational complexity to solve the problem is only O(m) that can be solved in linear time.

Optimal Path: Once the state value of each state has been defined, the optimal path can be obtained by starting at an arbitrary position or the first row of the state with the maximal state value, and then selecting the state in the next row for which the sum of the state and action values $V_{i+1,k} + Q\left(S_{i,j}, S_{i+1,k}\right)$ is largest, where $S_{i,j}$ and $S_{i+1,k}$ are the current and next state respectively.

7.2.2 Dynamic Environment

The structure of a tree will rarely changes in a short term. The environment only changes when more accurate information on the shape of tree is obtained when Treebot gets closer to a given region. As the calculation of state value is a top-down process, for an ascending motion, a change in environment in the lower part does not affect the state values of the upper part, and only the state values in the lower part need to be modified. The path can then be updated according to the new state values.

7.3 Motion Planning

The path planning algorithm generates a 3D path on the manifold of tree surface with high likelihood of success. The next task comes to the motion planning to make Treebot follows the planned path. The ideal solution is that all the steps (front and rear gripper) and the body of robot can place on the planned path. However, finding a motion that keeps both the front and rear grippers and the continuum body on the planned path may not feasible due to the nonholonomic constraints of Treebot's kinematics. It is assumed that the path-following problem has a certain tolerance. This assumption is applicable because the path is planned to avoid obstacles at a certain distance. Searching methods can be applied to find the globally optimal motion sequence to fit the planned path. However, this is time consuming. As a result, a computationally efficient strategy to find a suboptimal solution is used rather than exhaustive searching.

7.3.1 Motion Planning Scheme

It may not be possible to have both grippers and the continuum body on the planned path, either one of the gripper can place on the path. As an alternative, either one of the grippers can be placed on the path and used to determine the position of the other gripper to minimize the path-following error. The front gripper-based method, in which all of the steps of the front gripper occur on the planned path, is adopted because it is more intuitive. With this method, the extension motion is used to move the front gripper to the planned path and the contraction motion is used to make the rear gripper adjust the orientation of Treebot to make the next extension motion best fit the planned path. The procedure for the motion-planning scheme is detailed as follows.

Path Segmentation

As it is intended that the front gripper is always placed on the planned path, the first task is to determine the target positions of the front gripper on the path. The path between the target positions of the front gripper are defined as path segments of the planned path. As Treebot has a variable length of step, the problem becomes one of determining the length of the continuum body in each climbing gait. As the gripping motion takes time, to climb efficiently, the body should contracts and extends as much as possibe so as to minimize the number of gripping motions required. To simplify the problem, the length of the contraction motion is set as the minimum admissible length while the distance between the target positions of the front gripper is set as a constant. Once the planned path has been segmented, the next task is to determine the optimal position of the rear gripper at each step according to the path segments.

Plane Fitting

As the continuum body has an arc shape, to find an optimal direction of the rear gripper in which the future motion fits the future path segment, the path segment must be approximated as an arc. To achieve this, the path segment is first fitted onto a plane. It is assumed that the start and end points of the path segment are both on the plane. The path segment is first translated so that the start point is at the origin and then rotated (θ_z about the z-axis then θ_y about the y-axis) to put the end point on the z-axis as shown in Fig. 7.5(b).

To find the fittest plane, as illustrated in Fig. 7.5(c), an optimal θ_p

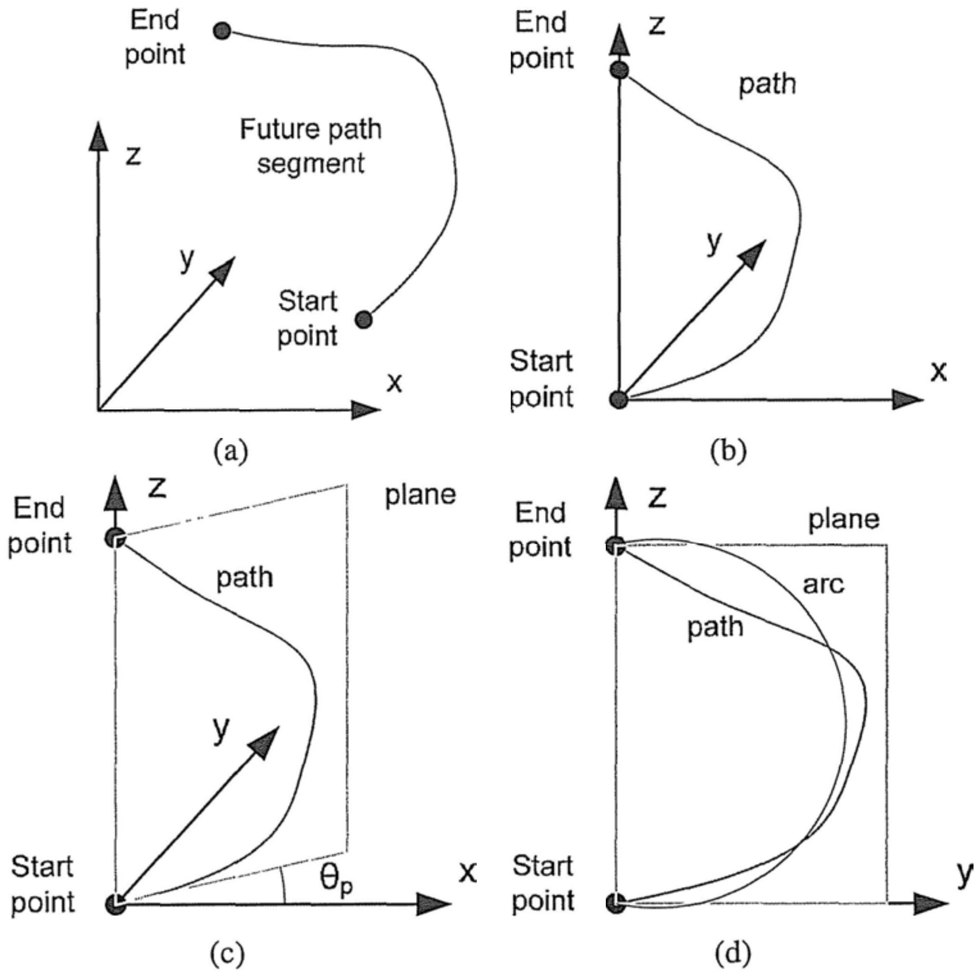


Figure 7.5: Procedures for arc fitting: (a) Path segment; (b) Transformation; (c) Plane fitting; (d) Arc fitting.

must be determined such that rotation about the z-axis of θ_p results in the minimum absolute x component value. Let $[x_i, y_i, z_i]$ be the transformed coordinates of the path segment. Referring to (6.3), θ_p can be obtained by:

$$\tan\theta_p = \frac{-q \pm \sqrt{q^2+4}}{2} \tag{7.5}$$

where $q = \frac{\sum {y_i}^2 - \sum {x_i}^2}{\sum x_i y_i}$.

Arc Fitting

To find an arc to fit the path segment, the path segment is first transformed into 2D. The path is then rotated about the z-axis of θ_p to transform the fitted plane into the y-z plane. The path is projected on the y-z plane thus becomes a 2D path, as illustrated in Fig. 7.5(d). The approximated arc must pass through the start and end points of the path segment. Let the number of sampling points of the path segment be η, (y_1, z_1) and (y_η, z_η) be the start and end points of the path segment. (y_i, z_i) represents the points of the path where $i \in [1, \eta]$. Let (y_c, z_c) and r be the center and the radius of the approximated arc respectively. To cross the start and end points, the approximated arc must fulfill the following equations:

$$(y_1 - y_c)^2 + (z_1 - z_c)^2 = r^2 \tag{7.6}$$

$$(y_\eta - y_c)^2 + (z_\eta - z_c)^2 = r^2 \tag{7.7}$$

Combining (7.6) and (7.7):

$$z_c = a - by_c \tag{7.8}$$

where $a = \frac{\left({y_1}^2+{z_1}^2\right)-\left({y_\eta}^2+{z_\eta}^2\right)}{2(z_1-z_\eta)}$ and $b = \frac{y_1-y_\eta}{z_1-z_\eta}$.

The distance error e_i of a point to the approximated arc can be found by:

$$e_i = (y_i - y_c)^2 + (z_i - z_c)^2 - r^2 \tag{7.9}$$

Sub. (7.6) into (7.9):

$$\begin{aligned} e_i &= (y_i - y_c)^2 + (z_i - z_c)^2 - \left[(y_1 - y_c)^2 + (z_1 - z_c)^2\right] \\ &= \left({y_1}^2+{z_1}^2\right) - \left({y_i}^2+{z_i}^2\right) - 2\left(y_1+y_i\right) y_c - 2\left(z_1+z_i\right) z_c \end{aligned} \tag{7.10}$$

Sub. (7.8) into (7.10):

$$e_i = g_i + h_i y_c \tag{7.11}$$

where $g_i = \left({y_1}^2 + {z_1}^2\right) - \left({y_i}^2 + {z_i}^2\right) - 2a\left(z_1 + z_i\right)$ and $h_i = 2b\left(z_1 + z_i\right) - 2\left(y_1 + y_i\right)$.

By using the least square method and consider all of the data points,

$$\sum \frac{d}{dy_c}{e_i}^2 = \sum \frac{d}{dy_c}(g_i + h_i y_c)^2 = 0$$

$$\Rightarrow y_c = -\frac{\sum g_i}{\sum h_i} \tag{7.12}$$

Once the value y_c has been obtained, the values z_c and r can be found by (7.6) and (7.8) respectively,

$$(y_c, z_c) = \left(-\frac{\sum g_i}{\sum h_i}, a - b\frac{\sum g_i}{\sum h_i}\right) \tag{7.13}$$

$$r = \sqrt{(y_1 - y_c)^2 + (z_1 - z_c)^2} \tag{7.14}$$

Optimal Direction of the Rear Gripper

By fitting the arc, the center and radius of the approximated arc can be obtained. The optimal position and direction of the rear gripper can then be determined as illustrated in Fig. 7.6. In the figure, the green dot and arrow represent the optimal position and direction of the rear gripper respectively, and the red arc represents the contraction posture of the continuum body. The rear gripper can be placed in this position and direction only if the direction of the front gripper is tangential to the starting point of the approximated arc. However, the direction of the front gripper is uncontrollable when the position of the front gripper is fixed as it is a nonholonomic system. As a result, the optimal position is neglected and the target direction of the rear gripper is set to the optimal direction. With this method, the position of the rear gripper shifts away from the optimal position. However, this shift does not much affect the path following result, as it is small compared with the length of the path segment.

The optimal direction of the rear gripper $\vec{v}_{rg}$ in the global frame can be obtained by:

$$\vec{v}_{rg} = Rot_z\left(-\theta_z\right) Rot_y\left(-\theta_y\right) Rot_z\left(-\theta_p\right) \begin{bmatrix} 0 \\ \cos\theta_r \\ \sin\theta_r \end{bmatrix} \tag{7.15}$$

where $\theta_r = \tan^{-1}\frac{z_c}{y_c} - \left(\frac{\pi}{2} + \frac{S_{\min}}{r}\right)$, $S_{\min}$ is the length of the continuum body in contraction (colored in red in Fig. 7.6), and $Rot_i\left(\theta\right)$ denotes the rotation matrix about i-axis in angle θ where $i \in x, y, z$.

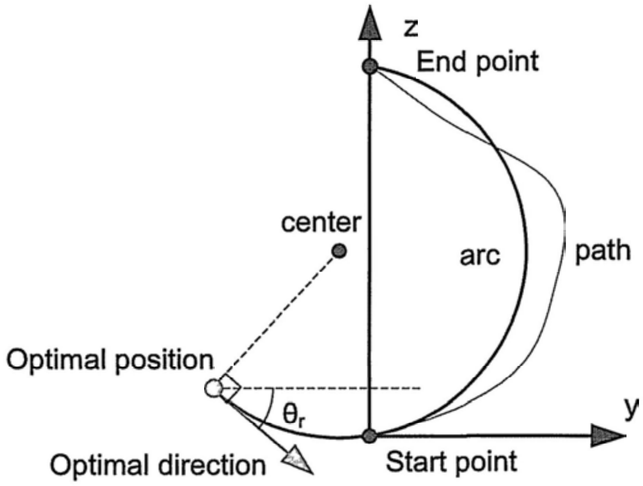


Figure 7.6: Optimal position and direction of the rear gripper.

7.3.2 Motion to the Target Position

Front Gripper

The target positions of the front gripper are defined by the path segmentation process. To determine the appropriate posture of the continuum body for placing the front gripper on the target positions, the target position is first transformed into the rear gripper frame, where the center of the rear gripper is at the origin and the direction vector of the rear gripper is on the z-axis. The posture of the continuum body to place the front gripper to the target position can then be obtained by (5.4).

Rear Gripper

If the rear gripper is in the optimal direction, then it may not be positioned on the tree surface. It is assumed that the surfaces at the target positions of the rear gripper and the current position of the front gripper have similar properties, as the distance between then is short in the contraction motion. To ensure that the target position of the rear gripper is on the tree surface, the optimal direction vector is projected on the plane defined by the surface normal to the front gripper position. As a result, to find the appropriate posture of the continuum body for placing the rear gripper, $\vec{v}_{rg}$ is first transformed to the front gripper frame so that the center of the front gripper is at the origin, the direction of the front gripper is on the z-axis and the surface normal vector is on the x-axis, as shown in Fig. 7.7. Then, $\vec{v}_{rg}$ is projected onto the y-z plane (blue arrow in the figure). Finally, the appropriate posture of the continuum body for placing the rear gripper can be determined as:

$$\phi = -sign\left(\theta_{rg}\right)\frac{\pi}{2} \tag{7.16}$$

$$\kappa = abs\left(\theta_{rg}\right)/S_{\min} \tag{7.17}$$

where $\tan\theta_{rg} = \frac{z'}{y'}$.

7.4 Simulations and Results

To evaluate the performance of the proposed global path and motion planning algorithm, a tree model that is composed of three branches is constructed. The tree surface is discretized as shown in Fig. 7.8. The rings are marked in different colors to distinguish the branches to which they belong to. The obstacles are marked in magenta.

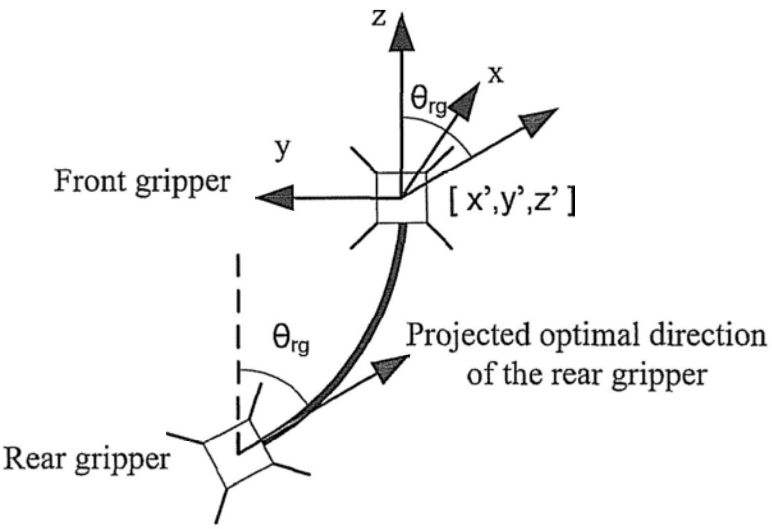


Figure 7.7: The concept to determine the posture of continuum body during a contraction motion.

7.4.1 Global Path Planning

To evaluate the global path planning algorithm, a target position is located at the top of the branch 2 and the initial position is located at the bottom of branch 1, as shown in Fig. 7.8. Fig. 7.9 illustrates the reward value $n_{i,j}$ and $O_{i,j}$ of the selected state space, that is, branch 1 and branch 2. In the figure, the hollow regions represent the location of obstacles. The state space arrangement in Fig. 7.9 may not reflect the actual geometric relationship. The planned path generated by the path planning algorithm is shown in Fig. 7.8 and Fig. 7.9 colored in black and grey respectively. In the figures, it can be observed that the planned path successfully reaches the target position by avoiding the obstacles and passing through positions with a high reward value.

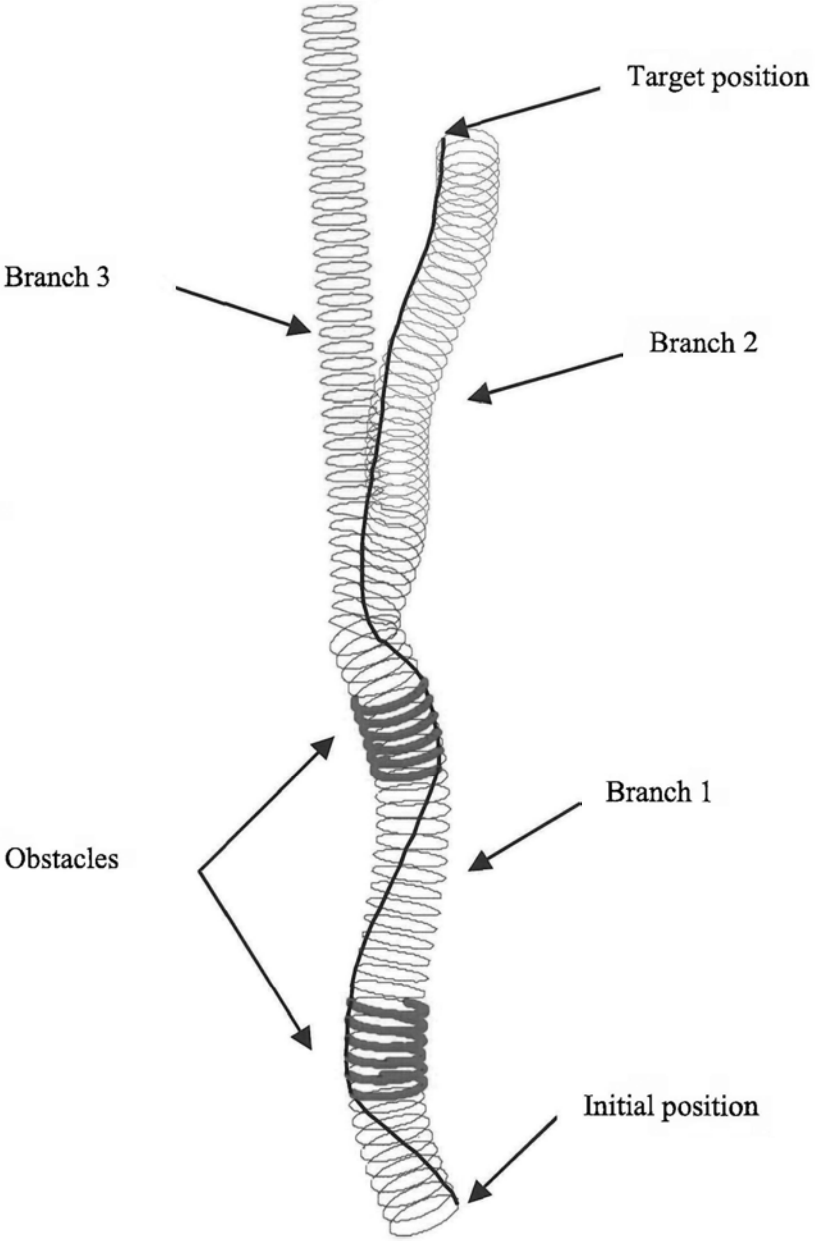


Figure 7.8: Experimental tree model.

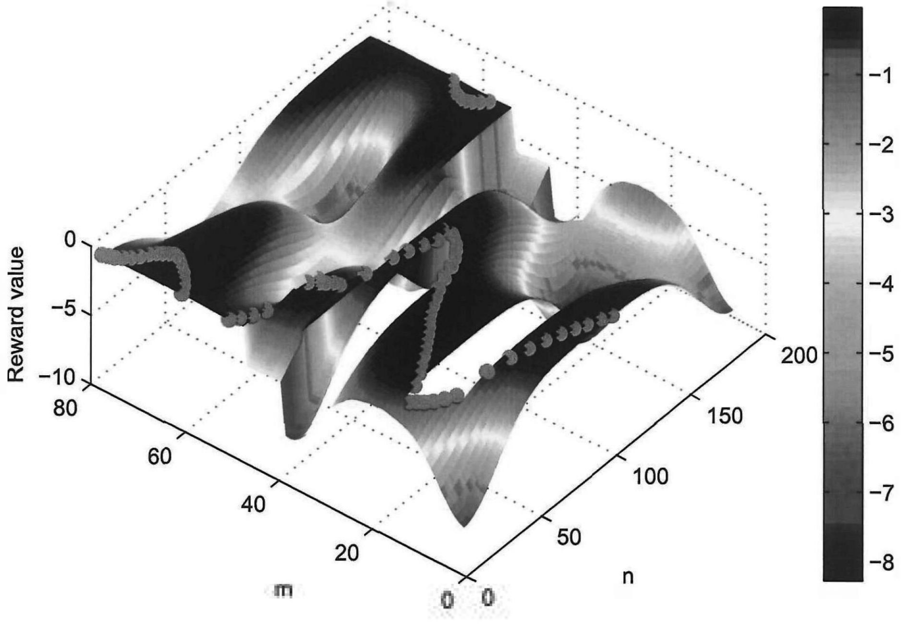


Figure 7.9: Reward value of the selected state space.

7.4.2 Motion Planning

Based on the planned path, the proposed motion planning algorithm has been applied. Fig. 7.10(b) illustrates the motions of Treebot obtained by the proposed motion planning algorithm. To show the motions of Treebot clearly, the obstacles are not displayed in the figure. In the figure, the red line indicates the robot body and the blue and magenta arrows denote the position and direction of the front and rear gripper respectively. The black line denotes the optimal path generated by the path planning algorithm. It can be observed that the motions of Treebot occur close to the planned path, which demostrates the effectiveness of the motion planning algorithm to track the planned path.

A simple alternative motion planning method is also applied for comparison, as illustrated in Fig. 7.10(a). In the simple motion planning algorithm,

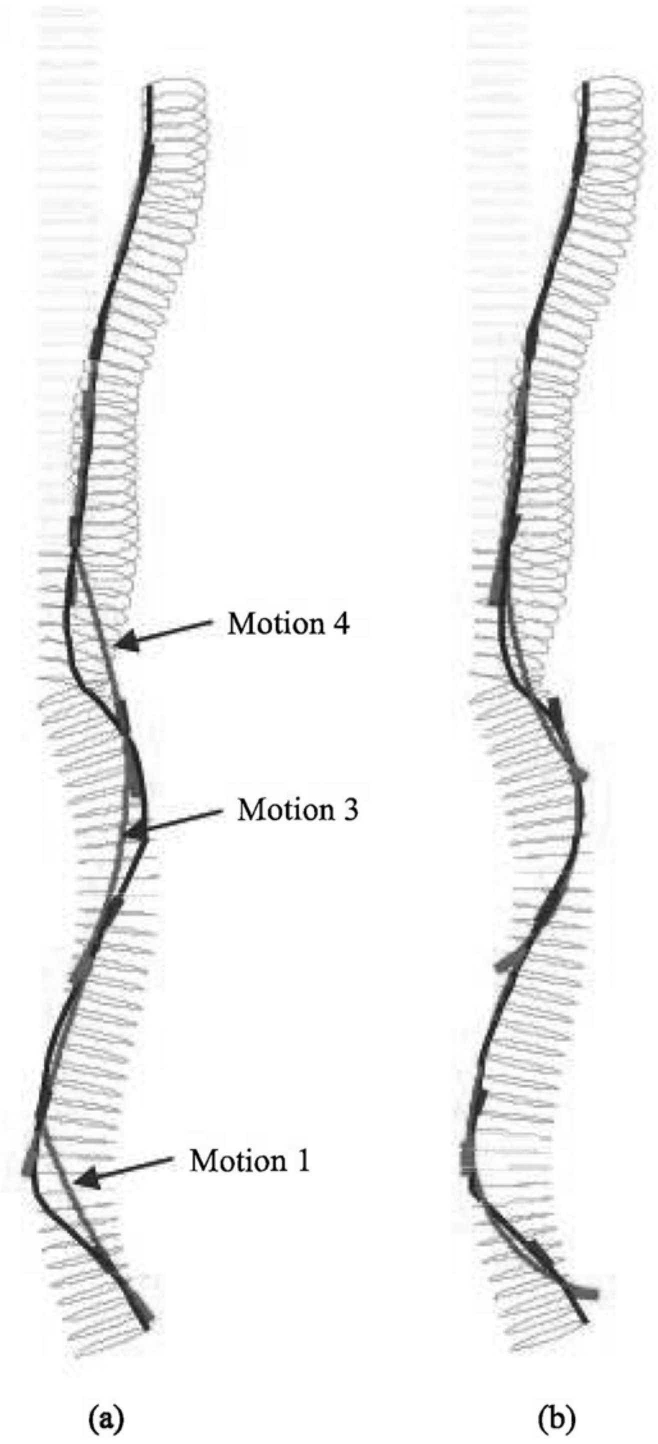


Figure 7.10: Motion planning results. (a) Simple method; (b) Proposed method.

the target direction of the rear gripper is simply equal to the direction of the front gripper. It can be noticed that by using the simple motion planning method, several motions (motion 1, 3, and 4) of Treebot occur far from the planned path. In that, motions 1 and 3 pass through the obstacles, which results in climbing failure. Comparison of the two motion planning results reveals the significance of the proposed motion planning algorithm.

7.5 Summary

In summary, this chapter presents a global path and motion planning algorithm for a tree climbing problem that can be solved in linear time. An intuitive method is proposed to represent the climbing space to simplify the complexity of the problem. A dynamic programming (DP) algorithm is adopted to find thc optimal climbing path that minimizes the climbing effort and avoids obstacles. A computationally efficient motion planning algorithm for Treebot is also proposed to guide Treebot to follow the planned path. The proposed motion planning algorithm is compared with a simple motion planning method. Result reveals that the significant improvement can be made by using the proposed motion planning algorithm.

□ **End of chapter.**

Chapter 8

Conclusion

To conclude this dissertation, the contributions to the fields of tree climbing robot are outlined. Based on the works we have achieved and the limitation of the method, we look beyond these approaches and propose some future research issues associated with these topics.

8.1 Contributions

8.1.1 Develop a Methodology and Design Principle for Tree Climbing Robots

A comprehensive study of the climbing methods in both natural and artificial aspects is done. The major fastening and maneuvering methods for arboreal environment are introduced and discussed. The six major design principles are then proposed for designing a tree climbing robot, which helps designers for making decision. The performances of the natural and artificial climbing principles are ranked based on the design principles. It is a good reference to help with the selection of the most appropriate climbing principles in designing a tree-climbing robot for specific purposes.

8.1.2 Design a Novel Tree Climbing Robot with Distinguish Performance

In this dissertation, a novel tree climbing robot, Treebot, is designed and developed. One of the original contributions of this work is the application of the *extendable continuum mechanism as a maneuvering system for tree climbing*. This opens up a new field of applications for the continuum mechanism. Through studies and experiments, it is discovered that the extendable continuum mechanism is highly suitable for tree climbing application. It gives Treebot has high maneuverability such that the admissible climbing workspace surpasses that of all the state of the art tree climbing robots. The inherent compliance of the continuum mechanism also simplifies the control issues and keeps Treebot lightweight.

Another contribution is the development of a miniature omni-directional tree gripper. This special mechanical design makes the gripper simple to control, and compact. It consumes zero energy in static gripping, which enables Treebot to remain on a tree for a long time. The gripper is also able to adhere to a wide variety of trees with a wide range of gripping curvatures. This allows Treebot to climb between a large tree trunk and small branches without any change in the gripper settings. On top of that, the gripper settings are optimized analytically to generate the maximum gripping force.

8.1.3 Develop an Autonomous Climbing Strategy in an Unknown Environment

An autonomous tree climbing algorithm has been developed that enables Treebot to explore and climb on an unknown shape of tree autonomously. Inspired by arboreal animals, avoid using complex sensing equipment such as cameras, an algorithm is proposed that approximates a shape of tree by using limited tactile sensors only. The results of experiments reveal that the proposed tree shape approximation algorithm can accurately approximate the shape of a tree. Generating an approximated shape of a tree allows Treebot to identify its environment and determine the optimal climbing motion. An associated motion planning algorithm is proposed, and experimental results show that it successfully guides Treebot to follow the optimal tree climbing path. It also reveals how tactile sensors can best be used use to aid autonomous tree climbing.

8.1.4 Develop a Global Path and Motion Planning Algorithm

A global motion-planning strategy for tree climbing is also developed in this dissertation to guide Treebot to climb to a target position based on a given global shape of tree. To the best of the author's knowledge, this is the first study to solve the global motion planning problem for tree climbing. An intuitive method is proposed to represent a climbing space that simplifies the complexity of the problem. A dynamic programming algorithm (DP) is then adopted to find the optimal climbing path that minimizes the climbing effort and avoids obstacles. A path planning solution is thereby obtained in linear-time. A computationally efficient motion planning algorithm is also proposed to guide Treebot to follow the planned path. The performance of the proposed motion planning method is compared with a simple motion planning method. Result reveals that the significant improvement can be made by using the proposed motion planning method.

8.2 Recommendation for Future Research

8.2.1 Compliance Modeling

The structure of the proposed continuum body has a three DOF of compliance. This has little effect when the length of extension is small. However, it will become more obvious when the length of the continuum body is extended, as gravitational force will easily deform the shape and orientation of the body and bend it out of an arc shape. This will affect the tree shape approximation algorithm and motion planning method, as the compliance is not considered in the algorithm. To avoid this problem, future studies should model the compliance of the continuum body and ensure that it is considered in the algorithm.

8.2.2 Dynamic Analysis

The speed of motion of the current Treebot prototype is not fast, and the dynamics can be neglected. With further development, Treebot could be made to climb much faster, but the dynamics would then need to be considered and thus require formulation in the future. In practical applications, Treebot may also need to carry heavy equipment. As the driving force of each tendon is limited, it will be necessary to formulate the force required for each tendon to achieve certain positions to identify the motion limits of Treebot for different loadings.

8.2.3 Hybrid Vision/Tactile-based Sensing

This dissertation proposes the use of tentacles to explore the environment. Although this approach performs well in approximating the local tree shape, the range of the sensors is limited. To obtain farer information to make a better path and motion planning, a combination of visual and tactile information is suggested. For example, a simple camera could be used to explore the rough shape of the tree, with tentacles being employed to explore the local environment. This approach combines the advantages of both sensing techniques to enrich the sensor information and improve the autonomous climbing performance of Treebot.

8.2.4 Global Map Building and Localization

To allow globally autonomous climbing, a global tree shape must be obtained. An appropriate method for obtaining a global tree shape should thus be developed in the future. In addition, a method for locating the robot on the tree is needed so that the proposed global motion-planning algorithm can be fully applied.

□ **End of chapter.**

Appendix A

Derivation of Equations

A.1 Rotation matrix

$$Rot_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{A.1}$$

$$Rot_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{A.2}$$

$$Rot_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

A.2 Inverse kinematics of the continuum manipulator

Initial position of tendon 1: $(-d, 0)$
Initial position of tendon 2: $\left(\frac{1}{2}d, -\frac{\sqrt{3}}{2}d\right)$
Initial position of tendon 3: $\left(\frac{1}{2}d, \frac{\sqrt{3}}{2}d\right)$

After transformation to the new direction of bend coordinate (rotation of ϕ about z-axis), the x coordinate of each spring becomes:

$$T_{1x} = -d\cos\phi \tag{A.4}$$
$$T_{2x} = \frac{1}{2}d\cos\phi - \frac{\sqrt{3}}{2}d\sin\phi = d\sin\left(\frac{\pi}{6} - \phi\right) \tag{A.5}$$
$$T_{3x} = \frac{1}{2}d\cos\phi + \frac{\sqrt{3}}{2}d\sin\phi = d\sin\left(\frac{\pi}{6} + \phi\right) \tag{A.6}$$

Assume constant curvature. Each tendon should have same θ, as a result, by given S, κ and ϕ, length of each tendon can be formulated as:

$$l_i = (r - T_{ix})\,\theta = (r - T_{ix})\,\kappa S \tag{A.7}$$

Then, the inverse kinematics becomes:

$$l_1 = S\,(1 + \kappa d\cos\phi) \tag{A.8}$$
$$l_2 = S\left(1 - \kappa d\sin\left(\frac{\pi}{6} - \phi\right)\right) \tag{A.9}$$
$$l_3 = S\left(1 - \kappa d\sin\left(\frac{\pi}{6} + \phi\right)\right) \tag{A.10}$$

A.3 Forward kinematics of the continuum manipulator

To find the forward kinematics, sub. (A.8) into (A.9):

$$l_2 = \frac{l_1}{(1+\kappa d\cos\phi)}\left(1-\kappa d\sin\left(\frac{\pi}{6}-\phi\right)\right)$$

$$\Rightarrow \kappa = \frac{l_1-l_2}{d\left(l_1\sin\left(\frac{\pi}{6}-\phi\right)+l_2\cos\phi\right)} \tag{A.11}$$

Sub. (A.8) into (A.10):

$$l_3 = \frac{l_1}{(1+\kappa d\cos\phi)}\left(1-\kappa d\sin\left(\frac{\pi}{6}+\phi\right)\right)$$

$$\Rightarrow \kappa = \frac{l_1-l_3}{d\left(l_1\sin\left(\frac{\pi}{6}+\phi\right)+l_3\cos\phi\right)} \tag{A.12}$$

Combine (A.11) and (A.12):

$$\frac{l_1-l_2}{d\left(l_1\sin\left(\frac{\pi}{6}-\phi\right)+l_2\cos\phi\right)} = \frac{l_1-l_3}{d\left(l_1\sin\left(\frac{\pi}{6}+\phi\right)+l_3\cos\phi\right)}$$

$$\begin{aligned}&(l_1-l_2)\left(l_1\sin\left(\frac{\pi}{6}+\phi\right)+l_3\cos\phi\right)\\ &=(l_1-l_3)\left(l_1\sin\left(\frac{\pi}{6}-\phi\right)+l_2\cos\phi\right)\end{aligned}$$

Since $\sin\left(\frac{\pi}{6}\pm\phi\right)=\sin\left(\frac{\pi}{6}\right)\cos\phi\pm\cos\left(\frac{\pi}{6}\right)\sin\phi=\frac{1}{2}\cos\phi\pm\frac{\sqrt{3}}{2}\sin\phi$,

$$(l_1-l_2)\left(\left(\frac{l_1}{2}+l_3\right)\cos\phi+l_1\frac{\sqrt{3}}{2}\sin\phi\right)$$

$$= (l_1 - l_3)\left(\left(\frac{l_1}{2} + l_2\right)\cos\phi - l_1\frac{\sqrt{3}}{2}\sin\phi\right)$$

$$(l_1 - l_2)\left((l_1 + 2l_3)\cos\phi + \sqrt{3}l_1\sin\phi\right)$$

$$= (l_1 - l_3)\left((l_1 + 2l_2)\cos\phi - \sqrt{3}l_1\sin\phi\right)$$

$$\left[\sqrt{3}l_1\,(l_1 - l_2) + \sqrt{3}l_1\,(l_1 - l_3)\right]\sin\phi$$

$$= \left[(l_1 - l_3)\,(l_1 + 2l_2) - (l_1 - l_2)\,(l_1 + 2l_3)\right]\cos\phi$$

$$\sqrt{3}l_1\,(2l_1 - l_2 - l_3)\sin\phi = 3l_1\,(l_2 - l_3)\cos\phi$$

$$\frac{\sin\phi}{\cos\phi} = \frac{3\,(l_2 - l_3)}{\sqrt{3}\,(2l_1 - l_2 - l_3)}$$

$$\tan\phi = \frac{\sqrt{3}\,(l_2 - l_3)}{(2l_1 - l_2 - l_3)} \tag{A.13}$$

$$\Rightarrow \phi = \tan^{-1}\frac{\sqrt{3}\,(l_2 - l_3)}{(2l_1 - l_2 - l_3)} \tag{A.14}$$

Eq. A.11 can be rewritten as:

$$\begin{aligned}
\kappa &= \frac{l_1 - l_2}{d\left(l_1\left(\frac{1}{2}\cos\phi - \frac{\sqrt{3}}{2}\sin\phi\right) + l_2\cos\phi\right)} \\
&= \frac{2\,(l_1 - l_2)}{d\,(l_1\,(\cos\phi - \sqrt{3}\sin\phi) + 2l_2\cos\phi)} \\
&= \frac{2\,(l_1 - l_2)}{d\,((l_1 + 2l_2)\cos\phi - \sqrt{3}l_1\sin\phi)} \\
&= \frac{2\,(l_1 - l_2)}{d\cos\phi\,((l_1 + 2l_2) - \sqrt{3}l_1\tan\phi)} \qquad \text{(A.15)}
\end{aligned}$$

Since $\cos\phi = \frac{1}{\sqrt{1+\tan^2\phi}}$,

$$\kappa = \frac{2\left(l_1 - l_2\right)\sqrt{1+\tan^2\phi}}{d\left(\left(l_1 + 2l_2\right) - \sqrt{3}l_1 \tan\phi\right)} \tag{A.16}$$

Sub. (A.13) into (A.16):

$$\begin{aligned} \kappa &= \frac{2\left(l_1 - l_2\right)\sqrt{1+\left(\frac{\sqrt{3}(l_2-l_3)}{(2l_1-l_2-l_3)}\right)^2}}{d\left(\left(l_1+2l_2\right) - \sqrt{3}l_1\frac{\sqrt{3}(l_2-l_3)}{(2l_1-l_2-l_3)}\right)} \\ &= \frac{2\left(l_1-l_2\right)\sqrt{\left(2l_1-l_2-l_3\right)^2 + \left(\sqrt{3}\left(l_2-l_3\right)\right)^2}}{d\left(\left(2l_1-l_2-l_3\right)\left(l_1+2l_2\right) - 3l_1\left(l_2-l_3\right)\right)} \\ &= \frac{2\sqrt{{l_1}^2+{l_2}^2+{l_3}^2 - l_1l_2 - l_2l_3 - l_1l_3}}{d\left(l_1+l_2+l_3\right)} \end{aligned} \tag{A.17}$$

Sub. (A.16) into (A.8):

$$\begin{aligned} S &= \frac{l_1}{\left(1 + d\frac{2(l_1-l_2)\sqrt{1+\tan^2\phi}}{d\left((l_1+2l_2)-\sqrt{3}l_1\tan\phi\right)}\frac{1}{\sqrt{1+\tan^2\phi}}\right)} \\ &= \frac{l_1}{\left(1+\frac{2(l_1-l_2)}{\left((l_1+2l_2)-\sqrt{3}l_1\tan\phi\right)}\right)} \end{aligned} \tag{A.18}$$

Sub. (A.13) into (A.18):

$$
\begin{aligned}
S &= \frac{l_1}{\left(1+\frac{2(l_1-l_2)}{\left((l_1+2l_2)-\sqrt{3}l_1\frac{\sqrt{3}(l_2-l_3)}{(2l_1-l_2-l_3)}\right)}\right)} \\
&= \frac{l_1}{\left(1+\frac{2(l_1-l_2)(2l_1-l_2-l_3)}{((2l_1-l_2-l_3)(l_1+2l_2)-3l_1(l_2-l_3))}\right)} \\
&= \frac{l_1}{\left(1+(2l_1-l_2-l_3)\frac{2(l_1-l_2)}{((2l_1-l_2-l_3)(l_1+2l_2)-3l_1(l_2-l_3))}\right)} \\
&= \frac{l_1}{\left(1+\frac{(2l_1-l_2-l_3)}{(l_1+l_2+l_3)}\right)} \\
&= \frac{l_1\left(l_1+l_2+l_3\right)}{((l_1+l_2+l_3)+(2l_1-l_2-l_3))} \\
&= \frac{(l_1+l_2+l_3)}{3} \qquad \text{(A.19)}
\end{aligned}
$$

Jones [17] introduces a kinematic model for a general class of continuum robot by different approaches. But it is found that this model is same as the model we have divided. It can bee seen that the equations are similar except the difference of ϕ is $\pi/2$.

A.4 Mapping between the coordinates of end positions and the posture of the continuum manipulator

$(S, \kappa, \phi) \leftarrow f\,(x_t, y_t, z_t)$:

According to Fig. A.1(a), ϕ can be determined by:

$$\phi = \tan^{-1}\frac{y_t}{x_t}$$

To find S and κ, it is first rotate the virtual tendon to x-z plane (refer to Fig. A.1(b)). Then,

$$\theta_1 = \tan^{-1}\frac{z_t}{{x_t}'}$$

where ${x_t}' = x_t \cos\phi + y_t \sin\phi$.

Once θ_1 is obtained, the radius of bend r can be found by:

$$r = \frac{\sqrt{{x_t}'^2 + {z_t}^2}}{2\cos\theta_1}$$

Hence,

$$\kappa = \frac{2\cos\theta_1}{\sqrt{{x_t}'^2 + {z_t}^2}} = \frac{2\cos\left(\tan^{-1}\frac{z_t}{{x_t}'}\right)}{\sqrt{{x_t}'^2 + {z_t}^2}} = \frac{2\frac{1}{\sqrt{1+\left(\frac{z_t}{{x_t}'}\right)^2}}}{\sqrt{{x_t}'^2 + {z_t}^2}} = \frac{2{x_t}'}{{x_t}'^2 + {z_t}^2}$$

as $\cos\left(\tan^{-1}x\right) = \frac{1}{\sqrt{1+x^2}}$. Refer to Fig. A.1(b)),

$$\theta = 2\left(\frac{\pi}{2} - \theta_1\right) = 2\left(\frac{\pi}{2} - \tan^{-1}\frac{z_t}{{x_t}'}\right)$$

$$\tan\left(\frac{\pi}{2}-\frac{\theta}{2}\right)=\frac{z_t}{x_t{}'}$$

$$\cot\frac{\theta}{2}=\frac{z_t}{x_t{}'}$$

$$\tan\frac{\theta}{2}=\frac{x_t{}'}{z_t}$$

$$\theta=2\tan^{-1}\frac{x_t{}'}{z_t}$$

Since $S=\theta/\kappa$,

$$S=\frac{x_t{}'^2+z_t{}^2}{x_t{}'}\tan^{-1}\frac{x_t{}'}{z_t}$$

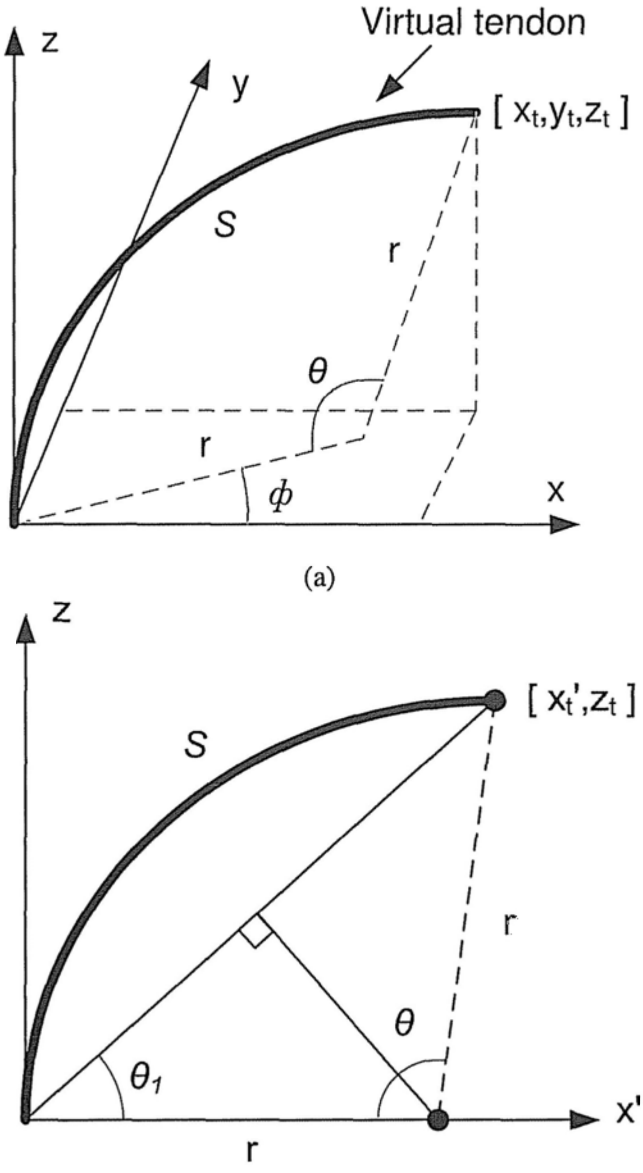
$(x_t, y_t, z_t) \leftarrow f(S, \kappa, \phi)$:

According to Fig. A.1,

$$\begin{bmatrix} x'_t \\ 0 \\ z_t \end{bmatrix} = \begin{bmatrix} r - r\cos\theta \\ 0 \\ r\sin\theta \end{bmatrix} = r\begin{bmatrix} 1-\cos\theta \\ 0 \\ \sin\theta \end{bmatrix}$$

Then,

$$\begin{aligned}\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} &= Rot_z(\phi)\begin{bmatrix} x'_t \\ 0 \\ z_t \end{bmatrix} = Rot_z(\phi)\, r\begin{bmatrix} 1-\cos\theta \\ 0 \\ \sin\theta \end{bmatrix} \\ &= \frac{1}{\kappa}\begin{bmatrix} [1-\cos(\kappa S)]\cos\phi \\ [1-\cos(\kappa S)]\sin\phi \\ \sin(\kappa S) \end{bmatrix}\end{aligned}$$



(a)

(b)

Figure A.1: Notations of the continuum manipulator.

A.5 Mapping between the coordinates of end positions and the posture of the Treebot

A.5.1 Rear gripper frame

$\left({}^{r}x_{f}, {}^{r}y_{f}, {}^{r}z_{f}\right) \leftarrow f\left(S, \kappa, \phi\right)$:

$$\begin{aligned}\begin{bmatrix} {}^{r}x'_{f} \\ 0 \\ {}^{r}z_{f} \end{bmatrix} &= r\begin{bmatrix} 1-\cos\theta \\ 0 \\ \sin\theta \end{bmatrix} + l_{r}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + l_{f}\begin{bmatrix} \sin\theta \\ 0 \\ \cos\theta \end{bmatrix} \\ &= \begin{bmatrix} r\left(1-\cos\theta\right)+l_{f}\sin\theta \\ 0 \\ r\sin\theta + l_{r} + l_{f}\cos\theta \end{bmatrix}\end{aligned} \tag{A.20}$$

$$\begin{aligned}\begin{bmatrix} {}^{r}x_{f} \\ {}^{r}y_{f} \\ {}^{r}z_{f} \end{bmatrix} &= Rot_{z}\left(\phi\right)\begin{bmatrix} {}^{r}x'_{f} \\ 0 \\ {}^{r}z_{f} \end{bmatrix} = Rot_{z}\left(\phi\right)\begin{bmatrix} r\left(1-\cos\theta\right)+l_{f}\sin\theta \\ 0 \\ r\sin\theta + l_{r} + l_{f}\cos\theta \end{bmatrix} \\ &= \begin{bmatrix} \left(\frac{1}{\kappa}\left[1-\cos\left(\kappa S\right)\right]+l_{f}\sin\left(\kappa S\right)\right)\cos\phi \\ \left(\frac{1}{\kappa}\left[1-\cos\left(\kappa S\right)\right]+l_{f}\sin\left(\kappa S\right)\right)\sin\phi \\ \frac{1}{\kappa}\sin\left(\kappa S\right)+l_{f}\cos\left(\kappa S\right)+l_{r} \end{bmatrix}\end{aligned}$$

$(S, \kappa, \phi) \leftarrow f\left({}^{r}x_f, {}^{r}y_f, {}^{r}z_f\right)$:

$$\phi = \tan^{-1}\frac{{}^{r}y_f}{{}^{r}x_f}$$

To find S and κ, it is first rotate the virtual tendon to x-z plane. In addition, transform l_r in z-axis (refer to Fig. A.2), thus, ${}^{r}z'_f = {}^{r}z_f - l_r$ and ${}^{r}x'_f = {}^{r}x_f \cos\phi + {}^{r}y_f \sin\phi$. Then,

$$ {}^{r}x'_f = r\left(1 - \cos\theta\right) + l_f \sin\theta \tag{A.21}$$

$$ {}^{r}z'_f = r\left(\sin\theta\right) + l_f \cos\theta \tag{A.22}$$

Reform (A.22):

$$r = \frac{{}^{r}z'_f - l_f \cos\theta}{\sin\theta} \tag{A.23}$$

Sub. (A.22) into (A.21):

$$ {}^{r}x'_f = \frac{{}^{r}z'_f - l_f \cos\theta}{\sin\theta}\left(1 - \cos\theta\right) + l_f \sin\theta$$

$$\begin{aligned} {}^{r}x'_f \sin\theta &= \left({}^{r}z'_f - l_f \cos\theta\right)\left(1 - \cos\theta\right) + l_f \sin^2\theta \\ &= {}^{r}z'_f - \left({}^{r}z'_f + l_f\right)\cos\theta + l_f \cos^2\theta + l_f \sin^2\theta \end{aligned}$$

$$\frac{\sin\theta}{1 - \cos\theta} = \frac{{}^{r}z'_f + l_f}{{}^{r}x'_f} = u \tag{A.24}$$

Let $\tan\theta = t$, $\sin\theta = \frac{t}{\sqrt{1+t^2}}$ and $\cos\theta = \frac{1}{\sqrt{1+t^2}}$, then,

$$\frac{\sin\theta}{1 - \cos\theta} = \frac{\frac{t}{\sqrt{1+t^2}}}{1 - \frac{1}{\sqrt{1+t^2}}} = \frac{t}{\sqrt{1+t^2} - 1} = \frac{\sqrt{1+t^2} + 1}{t} = u$$

$$\Rightarrow (ut-1)^2 = 1+t^2$$

$$u^2t^2 - 2ut + 1 = 1 + t^2$$

$$t\left[\left(u^2-1\right)t - 2u\right] = 0$$

$$t = \frac{2u}{u^2-1} \tag{A.25}$$

Sub. (A.24) into (A.25):

$$t = \frac{2\left(\frac{{}^r z'_f + l_f}{{}^r x'_f}\right)}{\left(\frac{{}^r z'_f + l_f}{{}^r x'_f}\right)^2 - 1} = \frac{2\,{}^r x'_f\left({}^r z'_f + l_f\right)}{\left({}^r z'_f + l_f\right)^2 - {}^r x'^2_f}$$

$$\Rightarrow \theta = \tan^{-1}\frac{2\,{}^r x'_f\left({}^r z'_f + l_f\right)}{\left({}^r z'_f + l_f\right)^2 - {}^r x'^2_f}$$

From (A.23):

$$r = \frac{{}^r z'_f - l_f \cos\theta}{\sin\theta} = \frac{{}^r z'_f - l_f\frac{1}{\sqrt{1+t^2}}}{\frac{t}{\sqrt{1+t^2}}} = \frac{{}^r z'_f\sqrt{1+t^2} - l_f}{t} \tag{A.26}$$

Sub. (A.25) into (A.26):

$$\begin{aligned} r &= \frac{{}^r z'_f\sqrt{1+\left(\frac{2u}{u^2-1}\right)^2} - l_f}{\frac{2u}{u^2-1}} = \frac{{}^r z'_f\frac{u^2+1}{u^2-1} - l_f}{\frac{2u}{u^2-1}} \\ &= \frac{\left(u^2+1\right){}^r z'_f - \left(u^2-1\right)l_f}{2u} \\ &= \frac{\left({}^r z'_f - l_f\right)u^2 + {}^r z'_f + l_f}{2u} \end{aligned} \tag{A.27}$$

Sub. (A.24) into (A.27):

$$\begin{aligned} r &= \frac{\left({}^{r}z'_{f} - l_{f}\right)\left(\frac{{}^{r}z'_{f}+l_{f}}{{}^{r}x'_{f}}\right)^{2} + \left({}^{r}z'_{f} + l_{f}\right)}{2\frac{{}^{r}z'_{f}+l_{f}}{{}^{r}x'_{f}}} \\ &= \frac{\left({}^{r}z'_{f} - l_{f}\right)\left({}^{r}z'_{f} + l_{f}\right)^{2} + {}^{r}x'^{2}_{f}\left({}^{r}z'_{f} + l_{f}\right)}{2x_{t}'\left({}^{r}z'_{f} + l_{f}\right)} \\ &= \frac{{}^{r}z'^{2}_{f} - {l_{f}}^{2} + {}^{r}x'^{2}_{f}}{2{}^{r}x_{f}} \end{aligned}$$

Hence,

$$\kappa = \frac{1}{r} = \frac{2{}^{r}x'_{f}}{{}^{r}z'^{2}_{f} - {l_{f}}^{2} + {}^{r}x'^{2}_{f}}$$

Finally,

$$S = r\theta = \frac{{}^{r}z'^{2}_{f} - {l_{f}}^{2} + {}^{r}x'^{2}_{f}}{2{}^{r}x'_{f}}\tan^{-1}\frac{2{}^{r}x'_{f}\left({}^{r}z'_{f} + l_{f}\right)}{\left({}^{r}z'^{2}_{f} + l_{f}\right)^{2} - {}^{r}x'^{2}_{f}}$$

A.5.2 Front gripper frame

For the case of the front gripper based, the derivation is similar to the rear gripper based but mirror about z-axis. Hence the difference are: ${}^{r}z_{f} \rightarrow -{}^{f}z_{r}$, $l_{f} \rightarrow l_{r}$ and $l_{r} \rightarrow l_{f}$.

As a result, $\left({}^{f}x_{r}, {}^{f}y_{r}, {}^{f}z_{r}\right) \leftarrow f\left(S, \kappa, \phi\right)$:

$$\begin{bmatrix} {}^{f}x_{r} \\ {}^{f}y_{r} \\ {}^{f}z_{r} \end{bmatrix} = \begin{bmatrix} \left(\frac{1}{\kappa}\left[1 - \cos\left(\kappa S\right)\right] + l_{r}\sin\left(\kappa S\right)\right)\cos\phi \\ \left(\frac{1}{\kappa}\left[1 - \cos\left(\kappa S\right)\right] + l_{r}\sin\left(\kappa S\right)\right)\sin\phi \\ -\left(\frac{1}{\kappa}\sin\left(\kappa S\right) + l_{r}\cos\left(\kappa S\right) + l_{f}\right) \end{bmatrix}$$

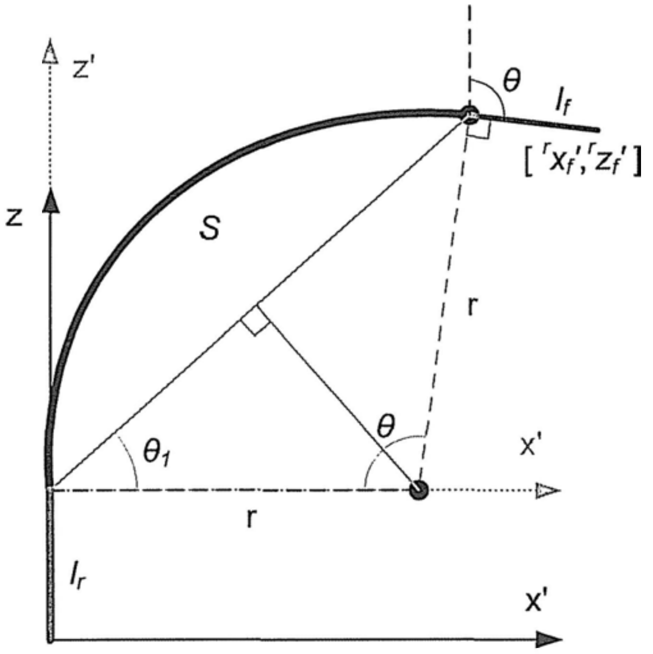


Figure A.2: Notations of Treebot.

$(S, \kappa, \phi) \leftarrow f\left({}^f x_r, {}^f y_r, {}^f z_r\right)$:

$$\begin{bmatrix} S \\ \kappa \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{1}{\kappa}\tan^{-1}\left(\frac{2{}^f\hat{x}_r\left({}^f\hat{z}_r+l_r\right)}{\left({}^f\hat{z}_r+l_r\right)^2-{}^f\hat{x}_r^2}\right) \\ \frac{2{}^f\hat{x}_r}{{}^f\hat{x}_r^2+{}^f\hat{z}_r^2-{l_r}^2} \\ \tan^{-1}\frac{{}^f y_r}{{}^f x_r} \end{bmatrix}$$

where ${}^f\hat{x}_r = {}^f x_r \cos\phi + {}^f y_r \sin\phi$ and ${}^f\hat{z}_r = -{}^f z_r - l_f$.

Appendix B

Author's Publications

1. Tin Lun Lam, Guoqing Xu, Huihuan Qian and Yangsheng Xu, "Linear-time Path and Motion Planning Algorithm for a Tree Climbing Robot - TreeBot", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18-22, 2010.

2. Huihuan Qian, Guoqing Xu, Jingyu Yan, Tin Lun Lam, Yangsheng Xu, and Kun Xu, "Energy Management for Four-Wheel Independent Driving Vehicle", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18-22, 2010.

3. Tin Lun Lam, Huihuan Qian and Yangsheng Xu, "Omnidirectional Steering Interface and Control for a Four-Wheel Independent Steering Vehicle", *IEEE/ASME Transactions on Mechatronics*, pp. 329-338, vol. 152, no. 3, June 2010.

4. Tin Lun Lam, Yangsheng Xu, and Guoqing Xu, "Traction Force Dis-

tribution on Omni-directional Four Wheel Independent Drive Electric Vehicle", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3724-3729, Kobe, Japan, May 12-17, 2009.

5. Tin Lun Lam, Huihuan Qian, Yangsheng Xu, and Guoqing Xu, "Omni-directional Steer-by-Wire Interface for Four Wheel Independent Steering Vehicle", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1383-1388, Kobe, Japan, May 12-17, 2009.

6. Tin Lun Lam, Huihuan Qian, and Yangsheng Xu, "Behavior-based Steering Control for Four Wheel Independent Steering Vehicle", *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 536-541, Bangkok, Thailand, February 21 - 26, 2009.

7. Huihuan QIAN, Tin Lun LAM, Weimin LI, Chenggang XIA, and Yangsheng XU, "System and Design of an Omni-directional Vehicle", *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 389-394, Bangkok, Thailand, February 21 - 26, 2009.

8. Hung-kwan Chan, Weizhong Ye, Tin-lun Lam, Yongsheng Ou, and Yangsheng Xu, "Sensor System for A Human-Following Robot", *Proceedings of the International Conference on Automation, Control, and Applications*, pp. 350-355, Novosibirsk, Russia, 2005.

□ **End of chapter.**

Bibliography

[1] S. Kim, M. Spenko, S. Trujillo, B. Heyneman, D. Santos and M. R. Cutkosky, "Smooth Vertical Surface Climbing With Directional Adhesion", *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 65-74, February 2008.

[2] H. Prahlad, R. Pelrine, S. Stanford, J. Marlow, and R. Kornbluh, "Electroadhesive Robots-Wall Climbing Robots Enabled by a Novel, Robust, and Electrically Controllable Adhesion Technology", *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, pp. 3028-3033, May 19-23, 2008.

[3] B. Aksak, M.P. Murphy, and M. Sitti, "Gecko inspired micro-fibrillar adhesives for wall climbing robots on micro/nanoscale rough surfaces", *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, pp. 3058-3063, May 19-23, 2008.

[4] D. Xu, X. Gao, X. Wu, N. Fan, K. Li, K. Kikuchi, "Suction Ability Analyses of a Novel Wall Climbing Robot", *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Kunming, China, pp. 1506-1511, December 17-20, 2006.

[5] W. Shen, J. Gu and Y. Shen "Permanent Magnetic System Design for the Wall-climbing Robot", *Proceedings of the IEEE International Conference on Mechatronics and Automation Niagara Falls*, Canada, vol.4, pp. 2078-2083, July 2005.

[6] S.J. Segal, S. Virost and W.R. Provancher, "ROCR: Dynamic Vertical Wall Climbing with a Pendular Two-Link Mass-Shifting Robot", *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, pp. 3040-3045, May 19-23, 2008.

[7] M. Murphy, M. Sitti, "Waalbot: An Agile Small-Scale Wall Climbing Robot Utilizing Dry Elastomer Adhesives", *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, June 2007.

[8] R. Aracil, R.J. Saltarn, and O. Reinoso, "A climbing parallel robot: a robot to climb along tubular and metallic structures", *IEEE Robotics and Automation Magazine*, vol. 13, no.1, pp. 16-22, March 2006.

[9] C. Balaguer, A. Gimenez, J.M. Pastor, V.M. Padron and M. Abderrahim "A climbing autonomous robot for inspection applications in 3D complex environments", *Robotica*, Cambridge University Press, vol. 18, no. 3, pp. 287-297, May 2000.

[10] Y. Yoon and D. Rus, "Shady3D: A Robot that Climbs 3D Trusses", *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, Italy, pp. 4071-4076, April 10-14, 2007.

[11] T. Mahmoud, M. Ali, M. Lino, and A.T. de Anibal, "3DCLIMBER: A climbing robot for inspection of 3D human made structures", *Proceedings*

of the IEEE International Conference on Intelligent Robots and Systems, Nice, France, pp. 4130-4135, September 22-26, 2008.

[12] A. Baghani, M. Ahmadabadi, and A. Harati, "Kinematics Modelling of a Wheel-Based Pole Climbing Robot (UT-PCR)", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2099-2104, April 18-22, 2005.

[13] Y. Kushihashi, et al., "Development of Tree Climbing and Pruning Robot, Woody-1-Simplification of Control using adjust Function of Grasping Power" (in japanese). *Proceedings of JSME Conference on Robotics and Mechatronics*, pp. 1A1-E08, 2006.

[14] H. Kawasaki, et al., "Novel climbing method of pruning robot", *Proceedings of the SICE Annual Conference*, Tokyo, pp. 160-163, 2008.

[15] M. J. Spenko, G. C. Haynes, J. A. Saunders, M. R. Cutkosky, A. A. Rizzi, "Biologically Inspired Climbing with a Hexapedal Robot", *Journal of Field Robotics*, vol. 25, no. 4-5, pp. 223-242, 2008.

[16] G. Haynes, et al., "Rapid Pole Climbing with a Quadrupedal Robot", *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, pp. 2767-2772, May 12-17, 2009.

[17] B.A. Jones, I.D. Walker, "Kinematics for Multisection Continuum Robots", *IEEE Transaction on Robotics*, vol. 22, no. 1, pp. 43-55, February 2006.

[18] K. Kotay, D. Rus, "The Inchworm Robot: A Multi-Functional System", *Autonomous Robots*, vol. 8, no. 1, pp. 53-69(17), January 2000.

[19] I. A. Gravagne and I. D. Walker, "Manipulability, Force, and Compliance Analysis for Planar Continuum Manipulators", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, June 2002.

[20] W.R. Provancher, J.E. Clark, B. Geisler, M.R. Cutkosky, "Towards penetration-based clawed climbing", *7th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Madrid, Spain, September 22-24, 2004.

[21] G. Immega, K. Antonelli, "The KSI tentacle manipulator", *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, vol. 3, pp. 3149-3154, 21-27 May 1995.

[22] W. McMahan, et al, "Field Trials and Testing of the OctArm Continuum Manipulator", *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, Florida, pp. 2336-2341, May 15-19, 2006.

[23] G. Chen, M.T. Pham, T. Redarce, "Development and kinematic analysis of a siliconerubber bending tip for colonoscopy", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, pp. 168-173, October 9-15, 2006.

[24] G. Robinson, J.B.C. Davies, "Continuum Robots - A State of the Art", *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, Michigan, vol.4, pp. 2849-2854, May 1999.

[25] K. Xu, N. Simaan, "Actuation Compensation for Flexible Surgical Snake-like Robots with Redundant Remote Actuation", *Proceedings of*

the IEEE International Conference on Robotics and Automation, Orlando, Florida, pp. 4148-4154, May 15-19, 2006.

[26] D.B. Camarillo, C.F. Milne, C.R. Carlson, M.R. Zinn, J.K. Salisbury, "Mechanics Modeling of Tendon-Driven Continuum Manipulators", *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1262-1273, December 2008.

[27] I.D. Walker, C. Carreras, "Extension versus Bending for Continuum Robots", *International Journal of Advanced Robotic Systems*, vol. 3, no. 2, pp. 171-178, 2006.

[28] Z. Li, J. Canny, "Nonholonomic Motion Planning", *Kluwer Academic Publishers*, 1993.

[29] Z. Fu, Y. Zhao, Z. Qian and Q. Cao, "Wall-climbing Robot Path Planning for Testing Cylindrical Oilcan Weld Based on Voronoi Diagram", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, pp. 2749 - 2753, October 9-15, 2006.

[30] S.M. LaValle, "Planning Algorithms", *Cambridge University Press*, 2006.

[31] T. Lozano-Perez. "Spatial planning: A configuration space approach", *IEEE Transactions on Computing*, vol. C-32, no. 2, pp. 108-120, February 1983.

[32] D. H́ahnel, W. Burgard, S. Thrun, "Learning compact 3D models of indoor and outdoor environments with a mobile robot", *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 15-27, 2003.

[33] D. Monnin, A.L. Schneider, F. Christnacher and Y. Lutz, "A 3D Outdoor Scene Scanner Based on a Night-Vision Range-Gated Active Imaging System", *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, Chapel Hill, North Carolina, USA, pp. 938-945, June 14-16, 2006.

[34] S.E. Dreyfus, A.M. Law, "The art and theory of dynamic programming", *Academic Press*, 1977.

[35] S. Hirose and H. Tsutsumitake, "Disk Rover: A Wall-Climbing Robot using Permanent Magnet Disks", *lEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, North Carolina, USA, vol. 3, pp. 2074-2079, July 7-10, 1992,

[36] D. Longo and G. Muscato, "SCID - A non-actuated robot for walls exploration", *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Como, Italy, vol. 2, pp. 874-879, 2001.

[37] Y. Fu, Z. Li, H. Yang and S. Wang, "Development of a wall climbing robot with wheel-leg hybrid locomotion mechanism", *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Sanya, China, pp. 1876-1881. December 15-18, 2007.

[38] R.T. Pack, J.L. Christopher and K. Kawamura, "A Rubbertuator-based structure-climbing inspection robot", *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, USA, vol. 3, pp. 1869-1874, April 20-25, 1997.

[39] H. Zhang, J. Zhang and G. Zong, "Effective pneumatic scheme and control strategy of a climbing robot for class wall cleaning on high-rise buildings", *International Journal of Advanced Robotic Systems*, vol. 3, no. 2, pp. 183-190, 2006.

[40] T.P. Sattar, H.L. Rodriguez and B. Bridge, "Climbing ring robot for inspection of offshore wind turbines", *International Journal of Industrial Robot*, vol. 36, no.4, pp. 326-330, 2009.

[41] O. Unver, A. Uneri, A. Aydemir and M. Sitti, "Geckobot: A gecko inspired climbing robot using elastomer adhesives", *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, pp. 2329-2335, May 15-19, 2006.

[42] G.D. Wile, et al., "Screenbot: walking inverted using distributed inward gripping", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, pp. 1513-18, September 22-26, 2008.

[43] Y. Jia and J. Tian, "Surface Patch Reconstruction From "One-Dimensional" Tactile Data", *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 400-407, April 2010.

[44] A.M. Okamura, M.R. Curkosky, "Feature-guided exploration with a robotic finger" *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 589-596, 2001.

[45] M. Schopfer, H. Ritter and G. Heidemann, "Acquisition and Application of a Tactile Database", *Proceedings of the IEEE International Conference*

on Robotics and Automation, Roma, Italy, pp. 1517-1522, April 10-14, 2007.

[46] H.C. Astley and B.C. Jayne, “Effects of perch diameter and incline on the kinematics, performance and modes of arboreal locomotion of corn snakes (Elaphe guttata)”, *The Journal of Experimental Biology 210*, pp. 3862-3872, 2007.

[47] M. Cartmill, “Pads and claws in arboreal locomotion”, *Primate Locomotion (ed. F. A. Jenkins Jr)*, New York: Academic Press, pp. 45-83, 1974.

[48] M. Cartmill, “The volar skin of primates: its frictional characteristics and their functional significance”, *Am. J. Phys. Anthropol.* vol. 50, no. 4, pp. 497-510, May 1979.

[49] M. Cartmill, “Climbing”, *In Functional Vertebrate Morphology (ed. M. Hildebrand, D. M. Bramble, K. F. Liem and D. B. Wake)*, Cambridge: Belknap Press, pp. 73-88, 1985.

[50] K.D. Hunt, J.G.H. Cant, D.L. Gebo, M.D. Rose, S.E. Walker and D. Youlatos, “Standardized descriptions of primate locomotor and postural modes”, *Primates*, vol. 37, pp. 363-387, 1996.

[51] Daniel Saenz, Christopher S. Collins, and Richard N. Conner, ”A bark-shaving technique to deter rat snakes from climbing red-cockaded woodpecker cavity trees”, *Wildlife Society Bulletin*, vol. 27, no. 4, Winter 1999.

[52] G.A. Pavlova, "Effects of Serotonin, Dopamine and Ergometrine on Locomotion in the Pulmonate Mollusc Helix Lucorum", *The Journal of Experimental Biology*, vol. 204, pp. 1625-1633, 2001.

[53] W. Federle, E. L. Brainerd, T. A. McMahon, B. Holldobler, "Biomechanics of the movable pretarsal adhesive organ in ants and bees", *Proceedings of the National Academy of Sciences*, USA, vol. 98, no.11, pp. 6215-6220, 2001.

[54] J. Gray, "The Mechanism of Locomotion in Snakes", *Journal of Experimental Biology*, vol.23, no.2, pp. 101-120, 1946.

[55] G. Goldman and D. W. Hong, "Considerations for Finding the Optimal Design Parameters for a Novel Pole Climbing Robot", *Proceedings of the ASME Mechanisms and Robotics Conference*, August 2008.

[56] M.P. Murphy, S. Kim and M. Sitti, "Enhanced Adhesion by Gecko-Inspired Hierarchical Fibrillar Adhesives", *ACS Appl. Mater. Interfaces* , Vol. 1, No. 4, pp. 849-855, 2009.

[57] X.J. Zhang, Y. Liu, Y.H. Liu, S.I.-U. Ahmed, "Controllable and switchable capillary adhesion mechanism for bio-adhesive pads: Effect of micro patterns", *Chinese Science Bulletin*, Vol. 54, No. 10, pp. 1648-1654, 2009.

[58] B. Aksak, M.P. Murphy, M. Sitti, "Gecko Inspired Micro-Fibrillar Adhesives for Wall Climbing Robots on Micro/Nanoscale Rough Surfaces", *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, pp. 3058-3063, May 19-23, 2008.

[59] M.F. Silva, J. Machado, J.K. Tar, "A Survey of Technologies for Climbing Robots Adhesion to Surface", *Proceedings of the IEEE International Conference on Computational Cybernetics*, Location, Stara Lesna, Slovakia, pp. 127 - 132, November 27-29, 2008.

[60] M. Scherge. and S. N. Gorb, "Biological Micro-and Nanotribology", *Springer-Verlag Berlin Heidelberg*, New York, pp. 107-110, 2001.

[61] K. Kim, Y. Kim and D. Kim, "Adhesion characteristics of the snail foot under various surface conditions", *International Journal of Precision Engineering and Manufacturing*, Vol. 11, No. 4, pp. 623-628, May 17, 2010.

[62] Y. Jiao, S. Gorb and M. Scherge, "Adhesion Measured on the Attachment Pads of Tettigonia Viridissima (Orthoptera, Insecta)", *The Journal of Experimental Biology*, Vol. 203, No. 12, pp. 1887-1895, 2000.