Climbing Strategy for a Flexible Tree Climbing Robot—Treebot

Tin Lun Lam, Member, IEEE, and Yangsheng Xu, Fellow, IEEE

Abstract—In this paper, we propose an autonomous tree climbing strategy for a novel tree climbing robot that is named Treebot. The proposed algorithm aims to guide Treebot in climbing along an optimal path by the use of minimal sensing resources. Inspired by inchworms, the algorithm reconstructs the shape of a tree simply by the use of tactile sensors. It reveals how the realization of an environment can be achieved with limited tactile information. An efficient nonholonomic motion planning strategy is also proposed to make Treebot climb on an optimal path. This is accomplished by the prediction of the future shape of the tree. The study that is presented in this paper also includes the formulation of Treebot kinematics and an analysis of the workspace of Treebot on different shapes of a tree. Numerous experiments have been conducted to evaluate the proposed autonomous climbing algorithm and to unveil the ability of Treebot.

Index Terms—Biologically-inspired robots, robotics in agriculture and forestry.

I. INTRODUCTION

CLIMBING robots constitute a challenging research topic that has gained much attention from researchers. Most of the climbing robots that are reported in the literature are designed to work on man-made structures, such as vertical walls and glass windows [1]–[6]. Few climbing robots have been designed to work on natural structures, such as trees.

WOODY [7] is one of the climbing robots that are designed to replace human workers in the removal of branches from trees. The robot fastens onto a tree by encircling an entire tree trunk and climbs up by extending and contracting its body. Kawasaki [8] developed a climbing robot for tree pruning. It uses a gripping mechanism that was inspired by lumberjacks and uses a wheel-based driving system for vertical climbing. Aracil [9] proposed a climbing robot, i.e., climbing parallel robot (CPR), which uses a Stewart–Gough platform to maneuver. RisE V2 [10] is a wall climbing robot that imitates the movement of an insect, using six legs to maneuver. This robot has also been demonstrated to be able to climb trees vertically.

For the aforementioned tree climbing robots, the workspaces are restricted to tree trunks only. Trees with branches and irregular shapes are not considered. Climbing on a tree with an irregular shape is a challenging task because the climbing motion and the adhesion method need to adapt to the complex and irregular surface. To achieve climbing on irregularly shaped trees, we developed a novel tree climbing robot that is named Treebot. The prototype of Treebot is shown in Fig. 1. Treebot is the first robot that uses a continuum mechanism [11] for climbing. It opens a new field of applications for continuum mechanisms. Treebot is able to climb on irregularly shaped trees with a high degree of maneuverability.

Treebot has the potential to be applied to various pursuits, such as harvesting, tree maintenance, and observation of tree-dwelling animals. A certain level of autonomous climbing ability of Treebot helps reduce the complexity of manipulation required for operation 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, and this can 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 that is 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, we developed a novel methodology to estimate the shape of trees by the usage of tactile sensors only. In addition, an inchworm-like motion planning strategy is developed to make Treebot climb up trees by the use of an optimal path. To the best of our knowledge, this is the first paper that focuses on the motion planning problems that are associated with climbing irregularly shaped trees. Kotay [17] also developed an inchworm-like robot and proposed a motion planning scheme. However, the nature of its maneuvering mechanism and climbing scenario are different to ours and inapplicable on Treebot.

The remainder of this paper is organized as follows. In Section II, we give an introduction to the mechanical design and sensing equipment of Treebot. In Section III, the kinematics of Treebot is presented. The workspace of Treebot on different shapes of trees is discussed in Section IV. In Section V, we discuss the proposed autonomous climbing algorithm and present a tree-shape approximation method and motion planning strategy. The experiment results are presented in Section VI. Finally, the conclusion and suggestions for future work are summarized in Section VII.

II. DESIGN OF TREEBOT

The design of Treebot is aimed to maximize the maneuverability and at the same time to minimize the weight as it is one of the critical concerns in a climbing robot. It is achieved by the minimization of the number of usage of actuators and adoption of the lightweight mechanism. Fig. 2 shows the structure of Treebot. Treebot is composed of three main elements: gripper, continuum body, and semipassive joint. Two grippers are connected to the ends of the continuum body, respectively, and the semipassive joint is installed between the body and the front gripper.

The proposed gripper is designed to fasten onto a wide variety of trees with a wide range of sizes. The gripper is composed of four claws and tactile sensors. The gripper fastens onto a tree by means of claw penetration. The gripper 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 gripper has one linear motor that actuates all four claws.

The continuum body is a type of single-section continuum manipulator [11] with a novel mechanism. It has high degrees of freedom (DOF) and a superior ability to extend the weight as it is one of the critical concerns in a climbing robot. It is achieved by the minimization of the number of usage of actuators and adoption of the lightweight mechanism. Fig. 2 shows the structure of Treebot. Treebot is composed of three main elements: gripper, continuum body, and semipassive joint. Two grippers are connected to the ends of the continuum body, respectively, and the semipassive joint is installed between the body and the front gripper.

The proposed gripper is designed to fasten onto a wide variety of trees with a wide range of sizes. The gripper is composed of four claws and tactile sensors. The gripper fastens onto a tree by means of claw penetration. The gripper 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 gripper has one linear motor that actuates all four claws.

To appress the gripper to a gripping surface, the gripper should have a certain turning ability about the y- and z-axes. 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. As a result, a 2-DOF semipassive revolute joint is installed between the front gripper and the continuum body. The joint can be locked and unlocked actively. To reduce the number of actuators, 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, which will be discussed in Section V.

To realize the motions of Treebot and 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 are installed on each gripper to detect the interaction between the gripper and the climbing surface. A three-axis tilting sensor is also attached to the front gripper to measure the direction of gravity.

Consequently, only five actuators are used in Treebot, two for the motion of the grippers and three for the motion of the continuum body. The weight of Treebot including the battery is only 650 g, which is very light compared with other types of tree climbing robots. The climbing speed of Treebot is 73.3 cm/min with a maximum 1.75-kg payload, which is nearly three times its own weight.

### III. KINEMATICS ANALYSIS

Fig. 3 shows the configuration of Treebot. In the notations, the superscript $r$ and $f$ denote the rear- and the front-gripper frame, respectively. $l_f$ and $l_r$ represent the distance from the end of the continuum body to the center of the front and the rear gripper, respectively. $h_g$ denotes the distance between the base of the gripper and the continuum body. The reference frames of the front and rear grippers are also illustrated in the figure. The direction of a gripper denotes the direction along the positive $z$-axis, where a normal direction denotes the direction toward the positive $x$-axis.

Jones [11] develops a kinematic model of a continuum-type manipulator. It is also applicable to the proposed continuum body. It formulates the relationship between the posture of virtual tendon and the length of each tendon. Fig. 4 shows the notation used to represent the parameters of each tendon and the posture of virtual tendon. $S$, $\kappa$, and $\phi$ denote the length, curvature, and the direction of bend of the virtual tendon, respectively. $s_i$ denotes the length of each tendon, while $d$ is the distance between tendons and virtual tendon. According to Jones [11], the forward and inverse kinematics are defined as follows.

#### Inverse kinematics

\[
\begin{bmatrix}
S \\
\kappa \\
\phi
\end{bmatrix}
\leftarrow
f
\begin{bmatrix}
s_1 \\
s_2 \\
s_3
\end{bmatrix}
\]

\[
\begin{bmatrix}
S \\
\kappa \\
\phi
\end{bmatrix}
= S
\begin{bmatrix}
1 - d\kappa \sin (\phi - \pi/2) \\
1 + d\kappa \sin (\phi - \pi/6) \\
1 - d\kappa \cos (\phi - \pi/3)
\end{bmatrix}
\]

(1)

#### Forward kinematics

\[
S
= r\theta.
\]

\[
\kappa = 1/r.
\]

(3)

(4)

In addition

To formulate the kinematics of Treebot, $l_f$ and $l_r$ must be considered. In addition, the mapping between the positions coordinates and the posture of Treebot should also be formulated in our application. As a result, the kinematics of Treebot is developed by extending (1) and (2).

In view of the rear-gripper frame as shown in Fig. 5(a), the mapping between the end point of the robot $(x_r^f, y_r^f, z_r^f)$ is formulated as
(x_f^j, y_f^j, z_f^j) \leftarrow f (S, \kappa, \phi);

\begin{align*}
\begin{bmatrix}
x_f^j \\
y_f^j \\
z_f^j 
\end{bmatrix} &= \begin{bmatrix}
\frac{1}{\kappa} [1 - \cos (\kappa S)] + l_f \sin (\kappa S) \cos \phi \\
\frac{1}{\kappa} [1 - \cos (\kappa S)] + l_f \sin (\kappa S) \sin \phi \\
\sin (\kappa S) + l_f \cos (\kappa S) + l_r
\end{bmatrix}
\end{align*}

(S, \kappa, \phi) \leftarrow f (x_f^j, y_f^j, z_f^j):

\begin{align*}
\begin{bmatrix}
S \\
\kappa \\
\phi
\end{bmatrix} &= \begin{bmatrix}
\tan^{-1} \left( \frac{2 \dot{x}_f^j (z_f^j - l_r + l_f)}{(z_f^j - l_r + l_f)^2 - \dot{x}_f^j 2}ight) \\
\frac{2 \dot{y}_f^j}{\dot{x}_f^j 2} + \left( z_f^j - l_r \right)^2 - l_r 2 \\
\tan^{-1} \frac{y_f^j}{x_f^j}
\end{bmatrix}
\end{align*}

where \( \dot{x}_f^j = x_f^j \cos \phi + y_f^j \sin \phi \).

In view of the front-gripper frame as shown in Fig. 5(b), the mapping between the end point of the robot \((x_f^j, y_f^j, z_f^j)\) is formulated as

\begin{align*}
(x_f^j, y_f^j, z_f^j) & \leftarrow f (S, \kappa, \phi):
\end{align*}

\begin{align*}
\begin{bmatrix}
x_f^j \\
y_f^j \\
z_f^j 
\end{bmatrix} &= \begin{bmatrix}
\frac{1}{\kappa} [1 - \cos (\kappa S)] + l_r \sin (\kappa S) \cos \phi \\
\frac{1}{\kappa} [1 - \cos (\kappa S)] + l_r \sin (\kappa S) \sin \phi \\
- \left( \frac{1}{\kappa} \sin (\kappa S) + l_r \cos (\kappa S) + l_f \right)
\end{bmatrix}
\end{align*}

(A. Required Angle of Twist)

By providing the shape of the tree and the position and orientation of the rear gripper, the angle of twist that is required to place the front gripper appressed to the target surface can be determined. It is assumed that the geometry of a segment of the tree can roughly be approximated as a straight or curved cylinder. The radius of the tube \( R_{tube} \), length of the tube segment \( S_{tube} \), bending direction \( \phi_{tube} \), and bending curvature \( \kappa_{tube} \) represent the shapes of the centerline of the tree model, which is a concept similar to the description of a virtual tendon (see Fig. 4). The target position of the front gripper is defined by the angle of change \( \theta_t \), and the length of centerline \( S_t \), for the target position that is illustrated in Fig. 6(a). The distance between the continuum body and the tree surface is defined as \( h_y \). The center of the rear gripper is located at the origin of the reference frame of the tree \((x^T - y^T - z^T)\). By using (5) and (10), the coordinate \( \vec{P}_t \) and the normal vector \( \vec{n}_t \) of the target position can be obtained by

\begin{align*}
\vec{P}_t^T &= -\text{Rot}_z (\phi_{tube}) \text{Rot}_y (\theta_{tube}) \text{Rot}_z (\phi_{tube}) \begin{bmatrix}
\cos \theta_t \\
\sin \theta_t \\
0
\end{bmatrix} \\
\vec{n}_t^T &= \begin{bmatrix}
\cos \theta_t \\
\sin \theta_t \\
0
\end{bmatrix}
\end{align*}
\[
\begin{align*}
\vec{P}^T_i &= \frac{1}{\kappa_{\text{tree}}} \begin{bmatrix}
1 - \cos \theta_{\text{tree}} & \cos \phi_{\text{tree}} \\
1 - \cos \theta_{\text{tree}} & \sin \phi_{\text{tree}} \\
\sin \theta_{\text{tree}} & 0
\end{bmatrix} \\
&+ (b_y + R_{\text{tree}}) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \vec{n}_i 
\end{align*}
\]

where \( \theta_{\text{tree}} = \kappa_{\text{tree}} S_i \).

The target position and normal vector can be transformed to the rear-gripper frame \((x' - y' - z')\) by

\[
\begin{align*}
\vec{P}'_i &= \text{Rot}_y (-\theta_{xy}) \text{Rot}_x (-\theta_{xz}) \vec{P}^T_i \\
\vec{n}'_i &= \text{Rot}_y (-\theta_{xy}) \text{Rot}_x (-\theta_{xz}) \vec{n}^T_i
\end{align*}
\]

where \( \theta_{xz} \) and \( \theta_{xy} \) denote the angles between the tree and the rear-gripper frame, as illustrated in Fig. 6.

According to (9), the normal vector in the front-gripper frame can be determined by

\[
\vec{n}'_i = \text{Rot}_z (\phi_B) \text{Rot}_y (-\kappa_B S_B) \text{Rot}_x (-\phi_B) \vec{n}^T_i
\]

where \( S_B, \kappa_B, \) and \( \phi_B \) are the length, bending curvature, and bending direction of the continuum body.

Finally, the angle of twist about \( y' \)- and \( z' \)-axes to appress the front gripper to the target surface can be determined by

\[
\begin{align*}
\theta_{\text{twist},y} &= \tan^{-1} \left( \frac{n'_i}{n'_i} \right) \\
\theta_{\text{twist},z} &= \tan^{-1} \left( \frac{n'_i}{n'_i} \right)
\end{align*}
\]

where \( \vec{n}'_i = [n'_{ix} \ n'_{iy} \ n'_{iz}] \). The subscript “\( i \)” \( (i \in \{x, y, z\}) \) denotes the \( i \) component value in a vector.

\section*{B. Admissible Target Position}

The admissible gripping positions can be determined by the consideration of all the necessary constraints. Fig. 7 illustrates the admissible gripping positions of the front gripper on a straight tree for different directions of the rear gripper. In the figure, the long arrow at the bottom denotes the direction of the rear gripper. The inner circle illustrates the circumference of the tree. The dots are the admissible positions of the front gripper with the small arrow that denotes the direction of the front gripper. This information is useful to determine the motion of Treebot. It can be observed in the figure that when \( \theta_{xz} \) increases, the admissible angle of change increases accordingly.

\section*{V. 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 the gravitational force acts on the robot to direct it to the centerline of the tree [13]. 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. 8. 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. To complete the main loop of the procedure once is termed as a complete stride. By the repetition of the stride, Treebot can climb a tree along the optimal path. The following sections discuss this procedure in detail.

\subsection*{A. Tree-Shape Approximation}

The concept of tree-shape modeling is mentioned in Section IV. This section discusses the method used to approximate the values of the model parameters from information that is provided by the tactile sensors. Treebot explores the shape of a tree by tactile sensors that are 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 exploring strategy. Approximation of 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 by the usage of information from tactile sensors. Okamura and Cutkosky [14] proposed a method to extract the local features of a surface. Jia and Tian [15] reconstructed the unknown local curved surface by the usage of 1-D tactile data. Schopfer [16] used a 2-D 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 that is based on a known geometric model to speed up the exploration and reconstruction processes.

\subsubsection*{1) Exploring Strategy}

The proposed exploring strategy aims to trace a growing path of a tree by the use of the front gripper, which is similar to the feature-tracing method presented in [14]. The trajectory of the front gripper can then be used to...
reconstruct the shape of the tree. The top-left and top-right tactile sensors that are attached to the front gripper are used for exploring. The state and action pairs for the exploring motion are listed in Table I. A tactile sensor acts in a similar fashion to a mechanical switch. It is triggered when a force acts on the bottom part of the tactile sensor 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 tactile sensor is triggered frequently. The front gripper, then, moves to eliminate this unbalanced triggering between tactile sensors to keep the front gripper follow the growing path of the tree.

Once a tactile sensor is triggered, the Cartesian coordinates of the front gripper are recorded, which can be found by (5). During the exploring motion, the semipassive joint is locked to make (5) applicable. As only one tactile sensor is installed on each tactile sensor, there is no way to determine where force is exactly applied along the tactile sensors? As a result, to trigger a tactile sensor 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 both the left and right tactile sensors are triggered at the same time or where the average position of the points at which the left and right tactile sensors are triggered alternatively.

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 3-D 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 (rotation about the $z$-axis of $-\theta_z$ and then rotation about the $y$-axis of $-\theta_y$).

To simplify the 3-D arc fitting problem into 2-D, the data are fitted onto a plane, as illustrated in Fig. 9(a). This is accomplished by the determination of the optimal angle of rotation about the $z$-axis $\theta_{plane}$ to minimize the $x$-component value of the transformed data:

$$\theta_{plane} = \tan^{-1}\left(\frac{-\hat{a} \pm \sqrt{\hat{a}^2 + 4}}{2}\right)$$  \hspace{1cm} (18)

where $\hat{a} = \sum y_i^2 - \sum z_i^2$. When the fitted plane is obtained, the data are converted into 2-D by projection onto the plane. Then, from Fig. 9(b), the center of the approximated arc $(y_c, z_c)$, the curvature of bend $\kappa_{arc}$, and the angle of the arc $\theta_{arc}$ can be found by a 2-D arc fitting method:

$$(y_c, z_c) = \left(-\sum m_i / \sum n_i, a - b y_c\right)$$  \hspace{1cm} (19)

$$\kappa_{arc} = 1 / \sqrt{y_c^2 + z_c^2}$$  \hspace{1cm} (20)

$$\theta_{arc} = 2\cos^{-1}(y_c \kappa_{arc})$$  \hspace{1cm} (21)
where \( m_i = -\left(y_i^2 + z_i^2\right) - 2az_i, \) \( n_i = 2bz_i - 2y_i, \) \( a = \frac{y_m^2 + z_m^2}{2x_m}, \) \( b = y_m / z_m, \) \( (y_i, z_i) \) represents the transformed data points that \( i \in \{1, m\}. \)

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

\[
\vec{v}_S = \text{Rot}_z (\theta_z) \text{Rot}_y (\theta_y) \text{Rot}_z (-\theta_{\text{plane}}) \begin{bmatrix} 0 \\ -\cos \theta_{2-D} \\ \sin \theta_{2-D} \end{bmatrix}
\]

\[
\vec{v}_{\text{bend}} = \text{Rot}_z (\theta_z) \text{Rot}_y (\theta_y) \text{Rot}_z (-\theta_{\text{plane}}) \begin{bmatrix} 0 \\ -\sin \theta_{2-D} \\ \cos \theta_{2-D} \end{bmatrix}
\]  

(22)

(23)

where \( \theta_{2-D} = \tan^{-1} \frac{\dot{z}}{y_r} + \text{sign} (y_r) \frac{\pi}{2}. \)

3) Tree-Shape Reconstruction: To approximate the parameters of the tree model, the fitted arc is transformed into the tree frame, i.e., to transform the tangent vector of the arc on the \( z^T \)-axis:

\[
\vec{v}_S^T = \text{Rot}_z (\theta_{rx}) \text{Rot}_y (\theta_{ry}) \vec{v}_S
\]

\[
\vec{v}_{\text{bend}}^T = \text{Rot}_z (\theta_{rx}) \text{Rot}_y (\theta_{ry}) \vec{v}_{\text{bend}}
\]  

(24)

(25)

where \( \theta_{ry} = \sin^{-1} (v_{\text{arc}}^z), \) \( \theta_{rx} = -\tan^{-1} \frac{v_{\text{arc}}^x}{v_{\text{arc}}^y}, \) \( \vec{v}_S = [v_{S,x}^T \ v_{S,y}^T \ v_{S,z}^T], \) and \( \vec{v}_{\text{bend}} = [v_{\text{bend,x}}^T \ v_{\text{bend,y}}^T \ v_{\text{bend,z}}^T]. \)

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

\[
\phi_{\text{arc}} = \tan^{-1} \frac{v_{\text{bend,y}}}{v_{\text{bend,x}}}
\]  

(26)

According to Fig. 10, by giving the radius of a tree \( R_{\text{tree}} \), the values of the parameters of the tree model, i.e., \( \phi_{\text{tree}}, \kappa_{\text{tree}}, \) and \( S_{\text{tree}} \), can be determined as

\[
\phi_{\text{tree}} = \phi_{\text{arc}}
\]  

(27)

\[
\kappa_{\text{tree}} = \frac{1}{1/\kappa_{\text{arc}} - (h_y + R_{\text{tree}}) \cos \phi_{\text{tree}}}
\]  

(28)

\[
S_{\text{tree}} = \theta_{\text{arc}} / \kappa_{\text{tree}}.
\]  

(29)

4) Tree Radius Approximation: The data from the exploring motion can be used to approximate the shape of a tree but not the radius of a tree. Hence, two methods are proposed to approximate a 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. Once the front gripper fails to appress to the target position, 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 appressed to the target position.

The second method for the approximation of the radius of a tree involves the comparison of the angle of change to the upper apex \( \theta_{\text{optimal}} \) for different positions of the front gripper.

The details of the procedure to find the angle of change to the upper apex of the tree will be discussed in the next section. As to obtain \( \theta_{\text{optimal}} \) does not require the radius of the tree to be known, the new approximated tree radius \( R'_{\text{tree}} \) can be obtained as follows:

\[
R'_{\text{tree}} = R_{\text{tree}} \left| \theta_{\text{optimal}} / \theta'_{\text{optimal}} \right|
\]  

(30)

where \( \theta'_{\text{optimal}} \) is the difference between \( \theta_{\text{optimal}} \) obtained by two different positions of the front gripper, and \( R_{\text{tree}} \) 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 stride. However, the application of this method is not feasible when the inclined angle of the tree is \( \pi/2 \), because in this state, the slope in any position is minimal. In such a case, the radius of the tree can be approximated by using the first method.

B. 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 by the tilting sensor that is attached to the front gripper. As the tilting sensor is fixed on the front gripper, the coordinate with respect to the rear-gripper frame can be determined by the posture of Treebot as discussed in Section III. Once the tree shape has been approximated, the transformation relationship between the rear-gripper frame and the tree frame, i.e., \( \theta_{rx} \) and \( \theta_{ry} \), 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. 11. The angle of change that is required to reach the upper apex \( \theta_{\text{optimal}} \) is equivalent to the angle of rotation about \( z^T \)-axis that is required to make the gravity vector \( \vec{v}_{\text{gravity}} \) lie on the \( z^T \times x^T \) plane with a positive \( x \).
\( \theta_{optimal} = \tan^{-1}\left(\frac{v_{gravity_x}}{v_{gravity_z}}\right) \)  \( (31) \)

where \( v_{gravity} = [v_{gravity_x}, v_{gravity_y}, v_{gravity_z}] \).

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

\[ \varphi_{incline} = \sin^{-1}\left(\left|v_{gravity_z}\right|\right) \in [0, \pi/2]. \]  \( (32) \)

C. 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 tactile sensor signals can be used to detect whether the gripper is appressed to the surface of the tree. The gripper is regarded as appressed when any two tactile sensors are triggered diagonally.

When the gripper is on the target position, Treebot will try to appress the gripper to the tree surface. The semipassive joint will first unlock so that it can be rotated freely. The gripper then, pushes forward onto the tree surface to 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 will be recalculated. The process is then repeated until an appressed placement is achieved.

D. Motion Planning

The optimal solution to make 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 [19], in which the direction of the gripper and the position of the gripper are coupled, it is impossible to achieve this solution in one stride. However, it can be achieved in two strides. In the first stride, the direction of the rear gripper is adjusted, and in the second stride, the front gripper is set on the optimal path with target direction.

Fig. 12 illustrates the concept to achieve the target position and direction in two strides. In the figure, the circle denotes the target position and the arrow represents the target direction. Rectangles colored in white and gray represent the attached and detached grippers, respectively. After (a) exploration, Treebot acquires the optimal position and direction for the front gripper. The continuum body then (b) contracts and (c) adjusts the direction of the rear gripper. Finally, the front gripper moves to the target position in (d) in the appropriate direction along the path. The forward motion is completed when (e) the continuum body contracts to pull up the rear gripper.

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. 13. After (a) exploration, the front gripper moves directly to the target position by neglecting (b) the target direction. 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) in the next stride; see Fig. 12(c)–(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 two times faster. However, the drawback of this method is that it may not go exactly to the target position and direction due to inaccurate estimation of the future target position and direction.

1) Target Position of the Front Gripper: Basically, an optimal solution 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 that is generated by gravity. As a result, to avoid Treebot to make a large change in angle to reduce much of the pull-out force that is generated by gravity. The 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 two times faster. However, the drawback of this method is that it may not go exactly to the target position and direction due to inaccurate estimation of the future target position and direction.

2) 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 \( S_{explore} = S_{tree} - l - l_r \). The future optimal position can then be obtained by the same method to find 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 at the tree frame can be determined by

\[
\vec{p}_{Tb}^{T} = \vec{p}_{Tt}^{T} - \vec{v}_{Tf}^{T} f
\]

where \( \vec{v}_{Tf}^{T} \) 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 first transforms \( \vec{p}_{Tb}^{T} \) to the future front-gripper frame \( \vec{p}_{Tff}^{T} \) and, then, find out the posture of the continuum body, i.e., \( \theta_f \) and \( \phi_f \), to place its rear part to \( \vec{p}_{Tff}^{T} \) by (8).

The direction of the rear gripper in future front-gripper frame \( \vec{v}_{Tff}^{r} \) can be found by

\[
\vec{v}_{Tff}^{r} = \text{Rot}_z(\phi_f) \text{Rot}_y(-\theta_f) \text{Rot}_z(-\phi_f) [0 0 1]^T.
\]

The angle between the direction of the rear gripper and the growing direction of the tree in the target position \( \gamma_r \) can then be determined by transforming \( \vec{v}_{Tff}^{r} \) to the coordinate frame of the target position. To make the exploring motion easy to implement, \( \gamma_r \) should be as small as possible. As a result, \( \gamma_r \) 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

\[
\begin{align*}
\theta_r &= |\gamma_f - \gamma_r| \\
\phi_r &= \gamma_f - \gamma_r - \frac{\pi}{2}
\end{align*}
\]

where \( \gamma_f \) 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 to place the rear gripper \( S_r \) is defined by avoiding the tendons with a negative length. Equation (1) can be reformulated as

\[
\begin{bmatrix}
\hat{s}_1 \\
\hat{s}_2 \\
\hat{s}_3
\end{bmatrix} =
\begin{bmatrix}
S_r - d\theta_r \sin (\phi_r - \pi/2) \\
S_r + d\theta_r \sin (\phi_r - \pi/6) \\
S_r - d\theta_r \cos (\phi_r - \pi/3)
\end{bmatrix} = S_r -
\begin{bmatrix}
\hat{s}_1 \\
\hat{s}_2 \\
\hat{s}_3
\end{bmatrix}.
\]

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

\[
S_r = \max (\hat{s}_1, \hat{s}_2, \hat{s}_3).
\]

VI. EXPERIMENTS AND RESULTS

Numerous experiments have been carried out to evaluate the proposed autonomous climbing algorithm in terms of 1) tree-shape approximation, 2) optimal path following, and 3) climbing a tree with branches.
A. Tree-Shape Approximation

The results of three experiments to test tree-shape approximation, i.e., Test 1, Test 2, and Test 3, are shown in Figs. 14, 15, and 16, respectively. Subfigures (a) show the approximation target and the final exploring posture of Treebot, and subfigures (b) illustrate the approximation result. In subfigures (b), the solid arc denotes the posture of Treebot, the dots are the exploration data, the dotted arc represents the fitted arc, and the large circles represent the circumference of the ends of the segment of the tree. The parameters of the approximated shape of the tree are listed on the top of the figures. The dashed lines in subfigures (a) denote the centerline of the shape of 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 the tree is almost straight, which approximate the actual shape of the 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 to tackle 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 3-D.

B. Optimal Path Following

The proposed autonomous climbing algorithm can guide Treebot to climb along 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° inclined angle. Fig. 17 shows the exploring and climbing motions. The optimal path is marked as a dashed line in the figures. Initially, Treebot was not on the optimal path and climbed up according to the approximated shape of the tree only. After two climbing strides, the whole body of Treebot was successfully placed on the optimal path with motions similar to those proposed motion planning strategy that is illustrated in Fig. 13. This result indicates that the proposed autonomous climbing algorithm can successfully guide Treebot to follow the optimal climbing path.

C. Climbing on 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. 18(a). Finally, Treebot selected branch A to climb as shown in Fig. 18(b). In the second test, the initial position of Treebot was shifted a little bit to left, as shown in Fig. 18(c). This time, branch B was selected by the exploring motion [see Fig. 18(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.

VII. CONCLUSION AND FUTURE WORK

In this paper, the main contribution is the development of an autonomous tree climbing algorithm that enables Treebot to explore and climb autonomously on an irregularly shaped tree. This study includes the formulation of the kinematics of Treebot, an analysis of the workspace of Treebot on different shapes of trees, the tree-shape approximation method, and a motion planning strategy. Numerous experiments have been carried out on a tree to evaluate the proposed algorithm. Results reveal that the proposed tree-shape approximation algorithm can probably approximate the shape of the tree. The proposed motion planning strategy is also able to guide Treebot to climb on the optimal path of 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. In the future, we will propose the development of a branch-selection function that can guide Treebot to climb autonomously on a desired branch to further simplify control of it.

REFERENCES


