Robotica (2025), **43**, pp. 997–1026
doi:10.1017/S0263574725000062

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Motion planning and searching strategy of a transverse ledge climbing robot based on force feedback

Reno Pangestu, Guan Xian Yu and Chi-Ying Lin

Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
Corresponding author: Chi-Ying Lin; Email: chiying@mail.ntust.edu.tw

Received: 6 May 2024; **Revised:** 14 October 2024; **Accepted:** 6 January 2025; **First published online:** 3 February 2025

Keywords: transverse ledge climbing robot; motion planning; searching strategy; contact force estimation

Abstract

A transverse ledge climbing robot inspired by athletic locomotion is a customized robot aiming to travel through horizontal ledges in vertical walls. Due to the safety issue and complex configurations in graspable ledges such as horizontal, inclined ledges, and gaps between ledges, existing well-known vision-based navigation methods suffering from occlusion problems may not be applicable to this special kind of application. This study develops a force feedback-based motion planning strategy for the robot to explore and make feasible grasping actions as it continuously travels through reachable ledges. A contact force detection algorithm developed using a momentum observer approach is implemented to estimate the contact force between the robot's exploring hand and the ledge. Then, to minimize the detection errors due to dynamic model uncertainties and noises, a time-varying threshold is integrated. When the estimated contact force exceeds the threshold value, the robot control system feeds the estimated force into the admittance controller to revise the joint motion trajectories for a smooth transition. To handle the scenario of gaps between ledges, several ledge-searching algorithms are developed to allow the robot to grasp the next target ledge and safely overcome the gap transition. The effectiveness of the proposed motion planning and searching strategy has been justified by simulation, where the four-link transverse climbing robot successfully navigates through a set of obstacle scenarios modeled to approximate the actual environment. The performance of the developed grasping ledge searching methods for various obstacle characteristics has been evaluated.

1. Introduction

A transverse climbing robot refers to a specialized climbing robot that moves transversely along ledges [1], which are commonly part of the architectural features of high-rise residential buildings on the exterior of walls [2]. Based on the movement mechanism inspired by climbing athletes [3], such a robot is developed to perform rhythmic transverse movements by alternately grasping and releasing its grip on ledge features [4]. As improper worker safety in high-rise buildings may increase the number of fatalities [5], this robot can replace humans to perform important and potentially hazardous tasks such as monitoring [6], cleaning [7], maintenance, inspection [8], and rescue [9].

Feature ledges in buildings have various shapes that are often found, such as rain shelters, outside window trims, roofs, balconies, etc. [2]. Some climbing robots treat them as obstacles to be avoided, so they try to find an alternative path to move [10]. Instead of treating ledges as obstacles, transverse climbing robots treat them as a gripping medium and move transversely through the complex layout of features along the building. To accomplish autonomous and safe mobility functions, the robot's climbing strategy requires the ability to recognize and adapt to ledge feature changes, such as horizontal position, slope, and gaps at different elevations. To facilitate comprehension of building ledge features, an example of a high-rise building suitable for transverse climbing robot grasping characteristics can be found in

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

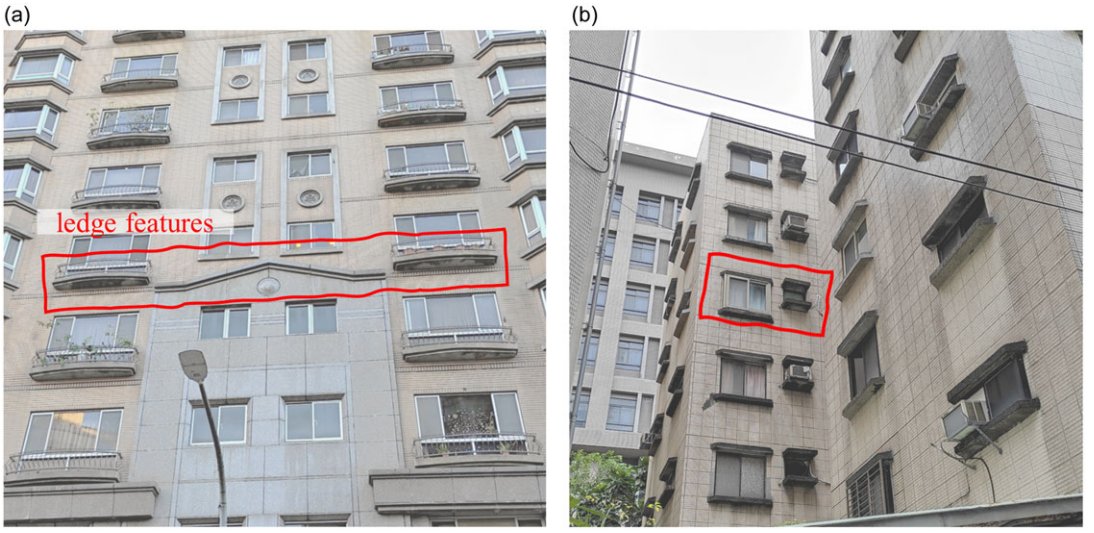


Figure 1. *Ledge features as an architectural part on high-rise buildings; (a) ledges slope change. (b) gap with the elevation difference.*

Fig. 1. The robot sensing system recognizing unknown environments consists of two types, i.e., contact-based (e.g., sensing) and non-contact-based (e.g., vision) [11]. One possible navigation solution is to install vision sensors and develop an algorithm to identify target grasping objects while the robot is moving [12].

Lin et al. [2] developed a transverse ledge climbing robot using a four-link planar mechanism. They integrated two infrared sensors for the navigation system at the top and bottom of the gripper. These infrared sensors measured the distance to the targeted ledge surface in real time. The robot can detect a change in slope between the two connected ledges when there is a difference in the distance measured by both sensors. Subsequently, the robot adjusted its posture to the slope of the ledge using a motion control algorithm based on the Central Pattern Generator. However, the static viewing angle of an infrared sensor limits its sensing area, making it difficult for the robot to recognize more complex ledge features, such as gaps with different heights or slopes. Additionally, the installation of these sensors increases the robot's volume, reducing its agility and movement space. Surface conditions such as scattering, reflection, and infrared (IR) light absorption can affect the effectiveness of infrared sensors [13]. As a result, the robot could not operate in an unknown environment without first identifying the surface properties of the ledge to ensure accurate measurements [14]. Furthermore, sensors that rely on emitted light waves suffer from decreased performance in smoky [15] and foggy [16] conditions.

In another robot application, Kim et al. [17] installed a camera sensor and a laser emitter on a cylinder grasp climbing robot. However, there were several failed attempts caused by the inability of the camera to capture objects with poor surface reflection. The experiment results of Lu et al. [18] indicate that light intensity also affects the error of the camera on the robot sensor in identifying objects. Similarly, Veiga et al. [19] had a blind spot problem with the LIDAR sensor on their climbing robot, so they installed multiple LIDAR sensors to perform navigation. Almost all navigation methods developed using visual sensors suffer from various problems, which makes it difficult to obtain reliable information from the working environment and derive movement decisions.

This study adopted a navigation method based on contact force feedback as an effective solution to overcome the limitations of visual sensor-based methods, which was inspired by a blind person who uses his cane to detect obstacles in front of him. This method is effective in recognizing environments that are partially unknown to the robot in advance, such as certain kinds of obstacles or sudden environment changes [20]. Some researchers have proven its effectiveness, such as Saund et al. [21], who developed a motion planning strategy on a robot manipulator by implementing a contact force

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

feedback method to relocate the object by avoiding various obstacles without utilizing a visual sensor. Similarly, Khedekar et al. [22] also justified the effectiveness of the contact force navigation method on aerial robots.

Kuffner et al. [23] implemented footstep locomotion planning, which is integrated into an algorithm to enable autonomous navigation through complex obstacles. Multi-contact interactions with the environment can improve the robot's ability to identify and adapt to various environment conditions while guided by a motion planning strategy [24]. However, to avoid severe failures during navigation, it is crucial to use a compliance controller to limit the torque load on the robot's joints caused by disturbances [25]. Ding et al. [26] introduced admittance control as a method for motion/force compliance, utilizing a general momentum observer to estimate the contact interaction with the environment. When the robot's dynamic model is available, the contact force can be estimated with the momentum observer without a force sensor device [27]. A time-varying threshold (TVT) can be set to manage the joint torque limits due to disturbances [28]. The combination of TVT and momentum observers provides a robust solution for compensating uncertainties in dynamic disturbances [29].

This paper proposes motion planning and environment-searching strategies for transverse climbing robots. Initially, the robot performs exploratory movements with one hand extended to the target position while the other hand grasps the ledge. When reaching the target position, the exploring hand will grasp the ledge, and the other hand will release the grip to return to the same initial position (hand-returning). When contact with the environment occurs, the robot must be able to adapt to changes in the environment. For this reason, the technique of a momentum observer is implemented to obtain contact force information. TVT is integrated into the robot sensing system to avoid detection errors due to dynamic uncertainties [30] and noises [31]. In addition, it can minimize collision effects [28]. When the estimated contact force exceeds the tolerance threshold, the robot will detect it as a collision, and then an admittance control will revise the motion planning to achieve a smooth and safe transition [32]. When the robot encounters a change in partially unknown environments (i.e., through a gap with different elevations), a couple of environment-searching methods adopted from search and rescue applications [33] will be implemented to investigate the navigation performance [34]. In numerical simulations, the robot has demonstrated its effectiveness in implementing this proposed contact force-based motion planning scheme.

The remainder of this paper is organized as follows: Section 2 presents the problem statement. Section 3 details the robot model, including inverse kinematic and dynamic equations. Section 4 details the contact force estimator using a momentum observer and the detection system based on TVT. Section 5 introduces motion planning and strategy. Section 6 details locomotion control, including computed torque control (CTC), environment impedance, and admittance control. Section 7 discusses the simulation results of the momentum observer, the effectiveness of the proposed motion planning and searching methods within the obstacle model, and a comparison with the vision-based method. Section 8 concludes the paper with remarks on the research contributions.

2. Problem statement

Figure 2 illustrates the problem formulation under investigation in this research. Basically, the climbing robot is designed to grasp and move along the transverse ledges. The main concern is how the robot can perform navigation and adaptation through a wide variety of ledge positions, which creates challenges in its path. With the proposed contact force-based method, the robot can recognize more complex ledge variations, overcoming the limitations of the visual-based method. The momentum observer approach is employed to estimate the contact force between the robot and the environment, utilizing the robot's dynamic information to calculate external disturbances without a force sensor. Instead, the momentum observer relies on motor encoders, which must be installed on each robot joint actuator [35].

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

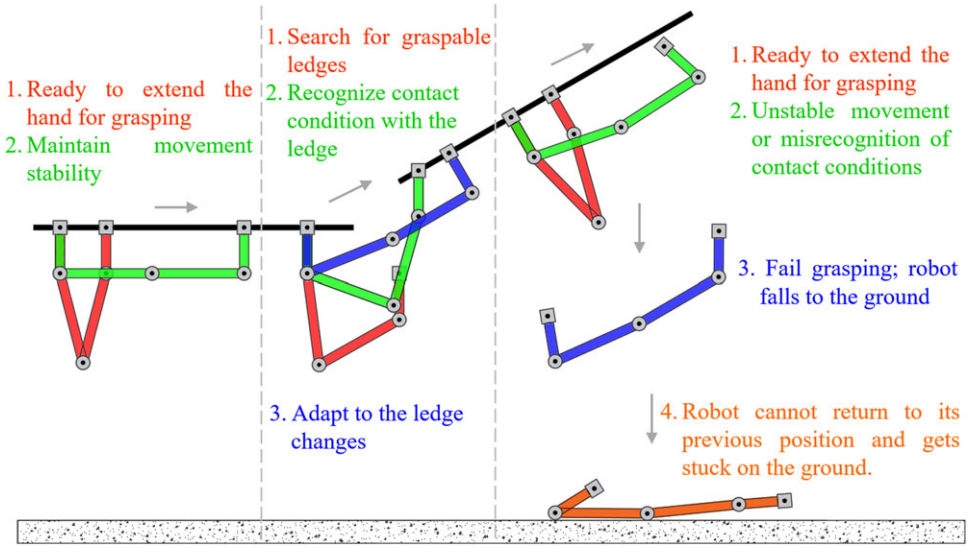


Figure 2. *Conceptual illustration of a transverse climbing robot when exploring the environment.*

When the robot detects contact with an unknown environment during exploration, admittance control is implemented to generate a new trajectory reference for the robot's posture adaptation based on the contact force information obtained from the momentum observer.

In a dynamic state, the robot generates several internal disturbance forces due to dynamic model uncertainty or sensor noise [36]. In certain situations, the robot mistakenly identifies internal disturbance forces as external disturbances, even when no contact with the environment has occurred. To overcome the problem of contact force misdetection, which may cause navigation or grasping failures, we developed a contact detection system by integrating the TVT with the momentum observer. Indeed, climbing robots cannot return to their previous position and continue moving if they fall due to improper movement or grasping failure.

To coordinate each step of the robot's movement and ensure successful environment exploration, we developed several locomotion templates integrated into a motion planning and search strategy. This strategy enables the robot to independently make proper locomotion decisions when encountering environment change. Since the robot's dynamic model is nonlinear, one of the most suitable approaches for maintaining the robot's dynamic stability is CTC [37].

As a mobile robot that relies on collision force feedback, we developed a locomotion template specifically designed for graspable ledge searching, inspired by rescue and maritime searching operations [33]. We analyzed several search methods, aiming to get a reasonable balance between energy efficiency and effective navigation.

To address the problem statements, this paper focuses on the following components: Momentum Observer, TVT, Motion Planning Strategy, Computed Torque Control, Admittance Control, and Position Graspable Ledge Searching.

3. Robot model

In this section, we introduce the robot design concept, dynamic model, and kinematic analysis which will be applied to the robot control system.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

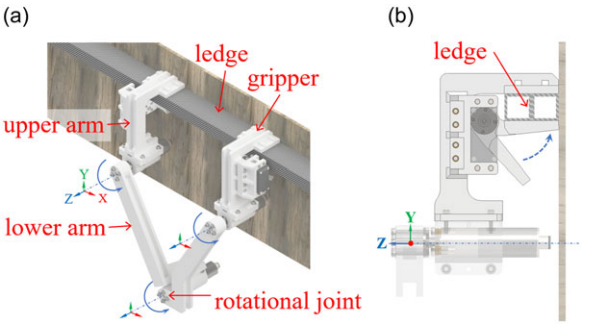


Figure 3. *Conceptual design model of a transverse ledge climbing robot; (a) isometric view (b) side view.*

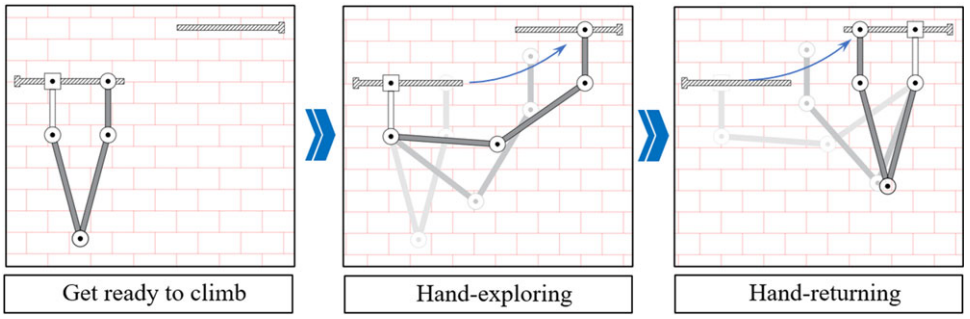


Figure 4. *The locomotion of a transverse climbing robot.*

3.1 Robot design concept

The concept of robot design and motion planning mechanisms is inspired by the actions of wall-climbing athletes [38]. While athletes perform a climbing movement, the limbs that contribute include the joints of the arms, shoulders, and palms. Based on these considerations, we designed a transverse climbing robot with a four-link planar mechanism that offers three degrees of freedom (3 DOF) motion while holding onto the ledge with a single hand. In this study, the conceptual design model of the transverse climbing robot is shown in Fig. 3. The main components of this robot are two upper arms, two lower arms, and two grippers at each end of the upper arms. Both grippers are actuated by a motor servo. The other rotational joints are actuated by brushed DC motors to perform motion control.

Previous generations of transverse ledge brachiation robots developed by Lin et al. [1, 3, 39] performed movements by swinging the lower limb for energy accumulation. Different from this type of locomotion, our robot mechanism performs an exploratory movement by extending one of its hands towards the target position while the other hand grasps tightly to hold the robot on the ledge. Figure 4 illustrates a four-link robot moving through a gap transition between two ledges at different elevations. Compared to robots using a swing-like movement style, our robot does not need to go through the energy storage phase to initiate the whole motion and thus can directly extend the exploring arm for ledge grasping. However, while extending the arm, our robot requires higher joint torques and gripping forces to hold it on the ledge.

The robot model used in this study was specifically designed for movement within a single plane aligned with the orientation and movement along a transverse ledge. We optimized the navigation to traverse horizontal ledges and climb within the plane of a high-rise building's outer wall. Consequently, the robot is limited in its ability to adapt to changes in ledge orientation that are perpendicular to its plane of motion.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

Table I. *DH parameter of the transverse ledge climbing robot.*

Link	l_i	α_i	d_i	q_i
0	l_0	0	0	−90°
1	l_1	0	0	q_1
2	l_2	0	0	q_2
3	l_3	0	0	q_3

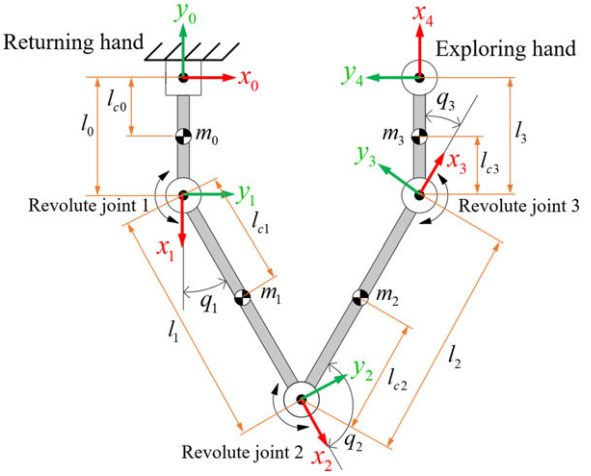


Figure 5. *Schematic diagram of the transverse ledge climbing robot.*

3.2. Robot dynamics

3.2.1. Lagrange-based dynamic model

The robot dynamic model is essential for joint motion control [40] and motion planning analysis. According to the Lagrange method [41], a dynamic model of the transverse climbing robot could be written in the following standard form:

$$M(q)\ddot{q} + C\,(q, \dot{q}) + G(q) + F_f(\dot{q}) = \tau + \tau_{ext} \tag{1}$$

Where $q, \dot{q}, \ddot{q} \in R^3$ represent the angular position, angular velocity, and angular acceleration, respectively; $M(q)\ddot{q} \in R^{3x3}$ is the moment of inertia, $C(q, \dot{q}) \in R^{3x3}$ is the centrifugal or Coriolis force; $G(q) \in R^{3xl}$ is the gravity force, $F_f(\dot{q}) \in R^{3xl}$ is the joint friction force, $\tau \in R^{3xl}$ is the joint torque; and $\tau_{ext} \in R^{3xl}$ is the external torque caused by external disturbances, especially the contact force between the robot end-effector and environment.

Figure 5 shows the schematic diagram of the four-link climbing robot, which has three degrees of freedom. Each link motion is joined by a revolving joint. The robot's hands are each a returning hand in the grasp of a ledge (fixed joint) and an exploring hand in the release of its grasp (end-effector). To identify the robot's kinematic link, we provide the Denavit–Hartenberg (DH) parameters in Table I [42], where l is the link length, α is the twist link, d is the link offset, and q is the joint angle. The robot's desired movement is in the right direction, according to a single plane. Table II summarizes the robot parameters' symbols and approximated values obtained from the CAD software.

3.2.2. Uncertain dynamic model

In practice, there are several changes in the parameters of the dynamic system when a robot performs the desired movements. This issue arises because of changes in the robot's physical uncertainty, such as the center of gravity, moment of inertia, centrifugal or Coriolis, and joint friction [30]. Besides that,

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

Table II. *Robot model parameter.*

Symbol	Meaning	Design value	Unit
m_0	Mass of returning-hand arm	0.471	kg
m_1	Mass of arm 1	0.377	kg
m_2	Mass of arm 2	0.14	kg
m_3	Mass of arm 3	0.471	kg
l_0	Length of returning-hand arm	100	mm
l_1	Length of arm 1	200	mm
l_2	Length of arm 2	200	mm
l_3	Length of arm 3	100	mm
l_{c0}	Gravity center distance of returning hand arm	61.058	mm
l_{c1}	Gravity center distance of arm 1	160.9	mm
l_{c2}	Gravity center distance of arm 2	94.8	mm
l_{c3}	Gravity center distance of arm 3	37.528	mm
θ_1	Joint angle 1 (returning hand arm to arm 1)	0∼360	degree
θ_2	Joint angle 2 (arm 1 to arm 2)	30∼270	degree
θ_3	Joint angle 3 (arm 2 to arm 3)	0∼360	degree

several other reasons may also cause dynamic uncertainty, such as noise in the sensor [31], measurement inaccuracies, and a slight fault in calculating the dynamic system when using model-based control methods. The existence of dynamic uncertainty is the primary issue with robot control stability [43, 44]. Even a slight consideration of this uncertainty is important to improving the stability and accuracy of robot movement. However, this uncertainty is difficult to predict and model, but it can still be modeled as an additional deviation from each parameter of the dynamic model, defined as follows [28]:

$$\begin{cases} M(q)\ddot{q} = M_0(q)\ddot{q} + \Delta M(q)\ddot{q} \\ C(q,\dot{q}) = C_0(q,\dot{q}) + \Delta C(q,\dot{q}) \\ G(q) = G_0(q) + \Delta G(q) \\ F_f(\dot{q}) = F_{f0}(\dot{q}) + \Delta F_f(\dot{q}) \end{cases} \tag{2}$$

where $M_0(q)\ddot{q}$; $C_0(q,\dot{q})$; $G_0(q)$ and $F_{f0}(q)$ are dynamic terms that can be modeled; while $\Delta M(q)\ddot{q}$; $\Delta C(q,\dot{q})$; $\Delta G(q)$ and $\Delta F_f(q)$ represent the unmodeled dynamics. The uncertain dynamic deviation ρ can be written as follows:

$$\rho = \Delta M(q) + \Delta C(q,\dot{q}) + \Delta G(q) + \Delta F_f(\dot{q}) \tag{3}$$

With the uncertain dynamic deviations, the robot model can be rewritten as follows:

$$M_0(q)\ddot{q} + C_0(q,\dot{q}) + G_0(q) + F_{f0}(\dot{q}) = \tau + f(\tau_{ext}, \rho, t) \tag{4}$$

where $f(\tau_{ext}, \rho, t)$ represent the lumped disturbance to the robot dynamic system.

3.3. Robot inverse kinematic

In a transverse climbing robot, the trajectory reference is based on the end-effector's desired position. However, the robot's motion is controlled by joint control. The inverse kinematics method [42] is used to determine the angular position of each robot joint based on the position of the grasping and extended hands.

To facilitate the discussion, in Fig. 6, we present a simple illustrative diagram of the four-link transverse climbing robot for kinematic analysis. When the robot grasps the ledge, the grasping hand position causes the upper arm position to become perpendicular to the grasped ledge. Therefore, if there is a

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

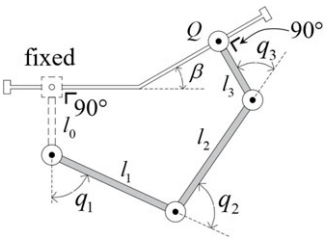


Figure 6. *Kinematic diagram of a four-link climbing robot with a slope environment.*

known inclination difference between the ledge grasped by each gripper, the robot's geometry could be identified as follows:

$$q_1 + q_2 + q_3 = 180^\circ + \beta \tag{5}$$

Using the concept of forward kinematic analysis of the robot, the position of the gripper (Q) at the slope angle can be written as follows:

$$\begin{cases} Q_x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) + l_3 \cos(180^\circ + \beta) \\ Q_y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) + l_3 \sin(180^\circ + \beta) \end{cases} \tag{6}$$

The angular position at joint 2 is written as follows:

$$q_2 = \cos^{-1}\left(\frac{(Q_x + l_3 \cos(\beta))^2 + \left(Q_y + l_3 \sin(\beta)\right)^2 - {l_1}^2 - {l_2}^2}{2 l_1 l_2}\right) \tag{7}$$

The angular position at joint 1 is written as follows:

$$A = \frac{-(Q_{x3} \cos\beta)\,(l_2 \sin q_2) + \left(Q_y + l_3 \sin\beta\right)(l_1 + l_2 \cos q_2)}{{l_1}^2 + {l_2}^2 + 2 l_1 l_2 \cos q_2} \tag{8}$$

$$B = \frac{(Q_{x3} \cos\beta)\,(l_1 + l_2 \cos q_2) + \left(Q_y + l_3 \sin\beta\right)(l_2 \sin q_2)}{{l_1}^2 + {l_2}^2 + 2 l_1 l_2 \cos q_2} \tag{9}$$

$$q_1 = \tan^{-1}\left(\frac{A}{B}\right) \tag{10}$$

If the angular position at joints 2 and 1 is known, the angular position at joint 3 can be obtained from Eq. (5).

4. A contact force estimator between robot and environment

Depending on the observer state level, the momentum observer response has different characteristics [28, 29]. In this section, we discuss the first-order (FOMO) and second-order (SOMO) momentum observers. FOMO uses a first-order transfer function that considers only changes in contact velocity, while SOMO uses an extended state with a second-order transfer function that considers both acceleration and velocity. Each state results in different output performance and computational load. Therefore, further investigation through simulation results (in Section 7.3) is necessary to determine which characteristics are most suitable for implementation in the climbing robot. Additionally, we discuss the use of a TVT as a contact force detector.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

4.1. First-order momentum observer (FOMO)

According to Leunberger's observer definition [45], the observer consists of a state variable vector $\hat{x}$ and an output vector $\hat{y}$. The state of the system to be estimated is momentum P, where the equation can be expressed as follows:

$$P(t) = M_0(q)\dot{q}(t) \tag{11}$$

Then, the change in momentum per unit time $\dot{P}$ is defined as follow:

$$\dot{P} = \dot{M}_0(q)\dot{q}(t) + M_0(q)\ddot{q}(t) \tag{12}$$

Substitute the equations (4) and (12) as follows:

$$\dot{P}(t) = \left(\dot{M}_0(q) - C_0(q,\dot{q})\right)\dot{q}(t) - \left(G_0(q) + F_{f0}(\dot{q})\right) + (\tau + f\,(\tau_{ext}, \rho, t)) \tag{13}$$

The matrix $(\dot{M}(q) - 2C(q,\dot{q}))$ is skew-symmetric [46], and the differentiation of the inertial matrix $M(q)$ could be written as:

$$\dot{M}(q) = C\,(q,\dot{q}) + C^T\,(q,\dot{q}) \tag{14}$$

Substitute equations (1) and (14) into Eq. (11), and the following is obtained:

$$\dot{P}(t) = C^T\,(q,\dot{q})\,\dot{q}(t) + \tau + \tau_{ext} - G(q) - \tau_f \tag{15}$$

The equation of FOMO can be written as follows:

$$\begin{cases} x_1(t) = P = M_0(q)\dot{q}(t) \\ \dot{x}_1(t) = \dot{P} = \tau_p^*(t) + f\,(\rho, \tau_{ext}, t) = \left(\tau - G_0(q) - F_{f0}(\dot{q})\right) + (\tau_{ext} - \rho) \\ \hat{x}_1(t) = \hat{P} \\ \dot{\hat{x}}_1(t) = \dot{\hat{P}} = \tau + K_I e - G_0(q) - F_{f0}(\dot{q}) \\ \hat{y}(t) = K_I\left(P - \hat{P}\right) \end{cases} \tag{16}$$

where $\tau_p^*(t)$ is the internal disturbance force of the system, $\hat{y}(t)$ is the observer output as an estimator of the external disturbance force on the system, $e{=}(P - \dot{P})$ is the state estimation error, and K_I is the observer gain constant for tuning the observer convergence. If the K_I is set too high, it can amplify the noise, leading to increase in the estimation errors. Therefore, adjusting the observer gain requires consideration of observer's bandwidth and noise level [58].

4.2. 4.2. Second-order momentum observer (SOMO)

Because of the sensor limitations in the system, when a robot makes contact with its environment, the physical state cannot be measured directly, making it difficult to model accurately. Therefore, the extended state observer (ESO) is a solution to estimate the unmeasured physical state of a dynamic system [47]. We extend a state observer to SOMO by introducing additional state variables in FOMO. Making it could enhance the response's performance in estimating the contact force. In this system, the equation for ESO variables is expressed as follows:

$$\begin{cases} x_1(t) = P(t) = M_0(q)\dot{q}(t) \\ \dot{x}_1(t) = \tau * (t) + x_2(t) \\ \dot{x}_2(t) = \phi(t) \\ y(t) = x_1(t) \end{cases} \tag{17}$$

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

Where x_2 is the new variable state of the SOMO, and ϕ is the unknown (unmeasured) disturbance. Thus, the SOMO equation can be written as follows:

$$\begin{cases} \hat{x}_1(t) = \hat{P}(t) \\ \dot{\hat{x}}_1(t) = \tau(t) + \tau_{ext}(t) - \tau_{f0}(\ddot{q}) - G_0(q) + \hat{x}_2(t) + \beta_1 e_1 \\ \hat{x}_2(t) = \tau_{ext}(t) - \rho(t) = \hat{f}(\rho, \tau_{ext}, t) \\ \dot{\hat{x}}_2(t) = \beta_2 e_1 \\ y(t) = \hat{x}_2(t) \\ e_1 = \hat{x}_1 - x_1 \end{cases} \tag{18}$$

Where $e_1 = \hat{x}_1 - x_1$ is the estimation error, β_1 and β_2 are the matrix observer gains for tuning the estimator convergence. The transfer function of the Laplace transforms from Eq. (18) can be written as follows:

$$\hat{x}_1 = \frac{\hat{x}_2 + \beta_1 p + \tau *}{\mathrm{s} + \beta_1} \tag{19}$$

$$\hat{x}_2 = \frac{\beta_2 p\,(\mathrm{s} + \beta_1 - \beta_1) - \beta_2 \tau_p^*}{\mathrm{s}^2 + \beta_1 s + \beta_2} \tag{20}$$

where $\hat{x}_1$ and $\hat{x}_2$ represent lumped disturbance estimators. The Eq. (20) follows the characteristic of the second-order standard equation $s^2 + 2\xi\omega_n s + \omega_n{}^2 = 0$. So, β_1 and β_2 can be obtained as follows:

$$\begin{cases} \beta_2 = \omega_0{}^2 \\ \beta_1 = 2\xi\omega_0 \end{cases} \tag{21}$$

where ω_0 and ξ represent the natural frequency and damping ratio of the system respectively, which determines the performance of the momentum observer.

4.3. Time-varying threshold (TVT)

The principle of TVT is to determine the tolerable force limit of an internal disturbance to avoid the possibility of misdetection [28]. According to Eq. (3), the bounded model uncertainty can be expressed as a polynomial equation consisting of angular acceleration, angular velocity, angular position, and other disturbances that cannot be modeled.

$$\left|\hat{\rho}_i\right| \le \delta_i = b_0 \left|\ddot{q}_i\right| + b_1 \left|\dot{q}_i\right| + b_2 \left|q_i\right| + b_3 \mathrm{sgn}\,(\dot{q}_i) \tag{22}$$

Where $\hat{\rho}$ is the model uncertainty that could be estimated while no external disturbance force is encountered by the robot, δ_i $(1 \le 1 \le n)$ is a model uncertainty bound, and b_0, b_1, b_2, and b_3 is a polynomial coefficient. To determine the polynomial coefficient, the parameters of TVT can be identified by expressing the equation of TVT in the linear regression model as follows:

$$\hat{\delta} = W(q_d, \dot{q}_d, \ddot{q}_d) \cdot \hat{\Theta} \tag{23}$$

where W is the regression matrix, while $\hat{\Theta}$ is the matrix coefficient. Each is defined as follows:

$$W(q, \dot{q}, \ddot{q}) = \left[\, \left|\ddot{q}_{di}\right| \; \left|\dot{q}_{di}\right| \; \left|q_{di}\right| \; \mathrm{sgn}\,(\dot{q}_{di}) \,\right] \tag{24}$$

$$\hat{\Theta} = \left[\, b_0 \quad b_1 \quad b_2 \quad b_3 \,\right]^T \tag{25}$$

When the robot moves along its trajectory, it must reidentify the $\hat{\Theta}$ to minimize the error in model uncertainty estimation by the parameter composition that consists of desired angular acceleration,

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

desired angular velocity, desired angular position, and estimated model uncertainty. Thus, it can be written as follows:

$$\hat{\Theta} = \left(W^T \cdot W\right)^{-1} W^T \cdot \hat{\delta} \tag{26}$$

According to the TVT model, the upper threshold δ_{Ui} and lower threshold δ_{Li} can be expressed as follows:

$$\begin{cases} \delta_{Ui} = W\left(q_d, \dot{q}_d, \ddot{q}_d\right) \cdot \hat{\Theta} + \lambda_1 \\ \delta_{Li} = W\left(q_d, \dot{q}_d, \ddot{q}_d\right) \cdot \hat{\Theta} - \lambda_2 \end{cases} \tag{27}$$

where λ_1 and λ_2 are phase offset coefficients that can be adjusted depending on the intensity of model uncertainty estimation. The principle of the TVT is that when the value of the force disturbance estimation received by the robot exceeds the thresholds, the system will detect that force disturbance estimation comes from external sources or contact with the environment.

5. Motion planning and searching strategy

Figure 7 provides the locomotion templates for the transverse climbing robot, which consist of four types that will be implemented in the motion planning and searching strategies. Type A motion planning is responsible for the hand-exploring task. Type B is responsible for hand-returning once the hand-exploring action has been done. Furthermore, Type C also does hand-returning, but with a different grasping hand position. Type C is responsible for achieving the proper position for executing the grasping position search procedure and stopping the robot when it finishes exploring the environment. Type D is tasked with executing the grasping position search operation. In type D, its searching path is designed inside the workspace. When the robot detects contact with the ledge, the admittance control will revise motion planning to adjust to the environment's direction change.

In this study, a motion planning and searching strategy algorithm is developed for the robot's navigation capabilities when exploring the environment, as the details of the pseudocodes are illustrated in Figure 8 below. This algorithm applies the four types of locomotion templates declared in Fig. 7 and utilizes the contact force feedback approach, which is then built into a single entity system that will be controlled when the robot encounters an environment change. Figure 8(a) demonstrates that this strategy is primarily designed to keep the robot moving continuously. When the robot recognizes an environment change, such as a slope change, a slope recognition algorithm is constructed in Fig. 8(c) with the aim of establishing a smooth transition motion for the robot. Then, when the robot needs to execute an attempt to transition through the gap between graspable ledges, a grasping position search algorithm is constructed in Fig. 8(b) with the aim of finding the graspable ledge with the robot's maximum reachable range capability.

Figure 9 demonstrates the algorithms for motion planning and searching strategies in a flow chart diagram designed for simplicity of comprehension. It is illustrated based on the three pseudocode algorithms in Fig. 8. This algorithm is designed to allow the robot to keep moving forward continuously, no matter the condition of the environment. As shown in Fig. 9, the robot is instructed to execute every step of exploration in the environment, then continue to begin a new step of exploration that may be different from the previous step, which it conducts repeatedly. The steps of exploration are variably dependent on the environment that the robot is confronting, such as ledge slope change, gap between the ledges, or no environment change.

The algorithmic strategy for grasping position search is designed to address two potential unintended situations that could lead to failure or limitations in environment exploration.

1. The first situation occurs when the robot grasps a ledge, but the position is identical to a previously grasped ledge. This can be detected by checking the tracking trajectory. The solution is to command the robot to search for a new graspable ledge position.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

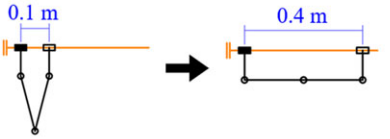
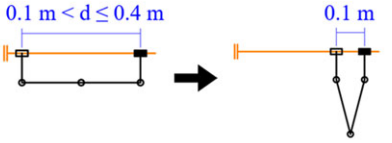
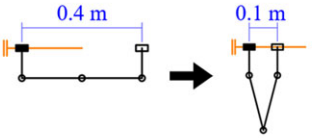
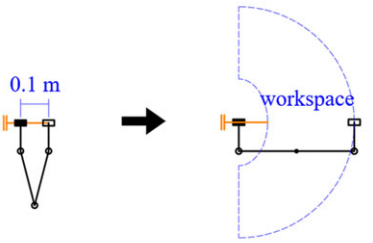
Type	Locomotion Initial → Final	Command	Task
A		if $\lvert\tau_{ext}\rvert > \lvert\delta_i\rvert$ motion= "admittance" else motion= to "final" end	Hand-exploring
B		-	Hand-returning
C		if $\lvert\tau_{ext}\rvert > \lvert\delta_i\rvert$ motion= "admittance" else motion= to "final" end	1. Get proper position for searching procedure 2. finish the exploration
D		if $\lvert\tau_{ext}\rvert > \lvert\delta_i\rvert$ motion= "admittance" else motion= to "final" end	Grasping position search procedure

Figure 7. *Locomotion templates of transverse climbing robot.*

2. The second situation occurs when the front hand successfully grasps the ledge (by motion Type D), but the ledge is too short to support the next step (by motion Type B). As a result, the rear hand would be unable to grasp the ledge. The solution involves two steps: first, abort motion Type B, and second, command the robot to search for another graspable ledge. Exploration should end if no new ledge position can be reached during the grasping position search process.

6. Locomotion control

The transverse climbing robot is designed to move continuously by using the contact force sense to recognize the environment's conditions. So, in this study, we need to develop a locomotion control strategy, as illustrated in Fig. 10, which aims to identify the environment and control the movement of the robot in each step of exploration.

The control process begins with the motion planning block, where the input reference is the desired end-effector trajectory x_d. The robot system is designed to control each joint angle between arms. Using inverse kinematics, x_d is converted to the desired angle trajectory q_d. Computed torque control computes the input torque τ_d based on the difference between q_d and the actual output of joint angle q. In addition, we include the disturbance torque feedback τ_d, which we modeled based on the environment's impedance, in order to maintain the robot's stability. The robot dynamics compute q based on the total joint torque τ. After obtaining the q, the following stage is essential to get the current end-effector location x using forward kinematics. The momentum observer is tasked with estimating the contact force F_{ext}, which then integrates with TVT to identify the contact force between the robot and the environment. If $|\hat{\tau}_{ext}|$ is still below the tolerance threshold $|\delta_i|$, which indicates that the robot does

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

(a)

Algorithm: Continuous transverse climbing

```
for exploration loop  % exploration started
    motion reference: type A
    if front-hand contact doesn't detected ⟶ |τ_ext| < |δ_i|
        if grasping position found ⟶ q_gripper = 90°
            motion reference: type B
        elseif grasping position doesn't found ⟶ q_gripper < 90°
            Algorithm: grasping position search
        endif
    elseif front-hand detects the contact (slope change) ⟶ |τ_ext| > |δ_i|
            Algorithm: Slope change transitioning
    endif
endfor
```

(b)

Algorithm: Grasping position search

```
grasping position does not found ⟶ q_gripper < 90°
motion reference: type C
get admittance adjustment reference ⟶ x_r
motion reference: type B
for searching loop  % grasping position search started
    motion reference: type D
    if front-hand detects the contact (grasping position found) ⟶ |τ_ext| > |δ_i|
        if found a new grasping ledge position
            get admittance adjustment reference ⟶ x_r
            motion reference: type B
            if The rear hand is unable to grasping the ledge  ⟶ q_gripper < 90°
                Abort the motion: Type B
            elseif The rear-hand is capable grasping the ledge  ⟶ q_gripper = 90°
                    breakfor % grasping position search ends
            endif
        elseif found the same ledge as the previous position
        endif
    elseif grasping position doesn't found  ⟶ |τ_ext| < |δ_i|
            motion reference: type C
            breakfor % grasping position search ends
            breakfor % exploration ends
    endif
endfor
```

(c)

Algorithm: Slope change transitioning

```
contact detected (slope change)⟶ |τ_ext| > |δ_i|
get admittance adjustment reference⟶ x_r
motion reference: type B
```

Figure 8. *Pseudocode algorithm of motion planning and searching strategy; (a) Continuous transverse climbing; (b) grasping position search; (c) slope change transitioning.*

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

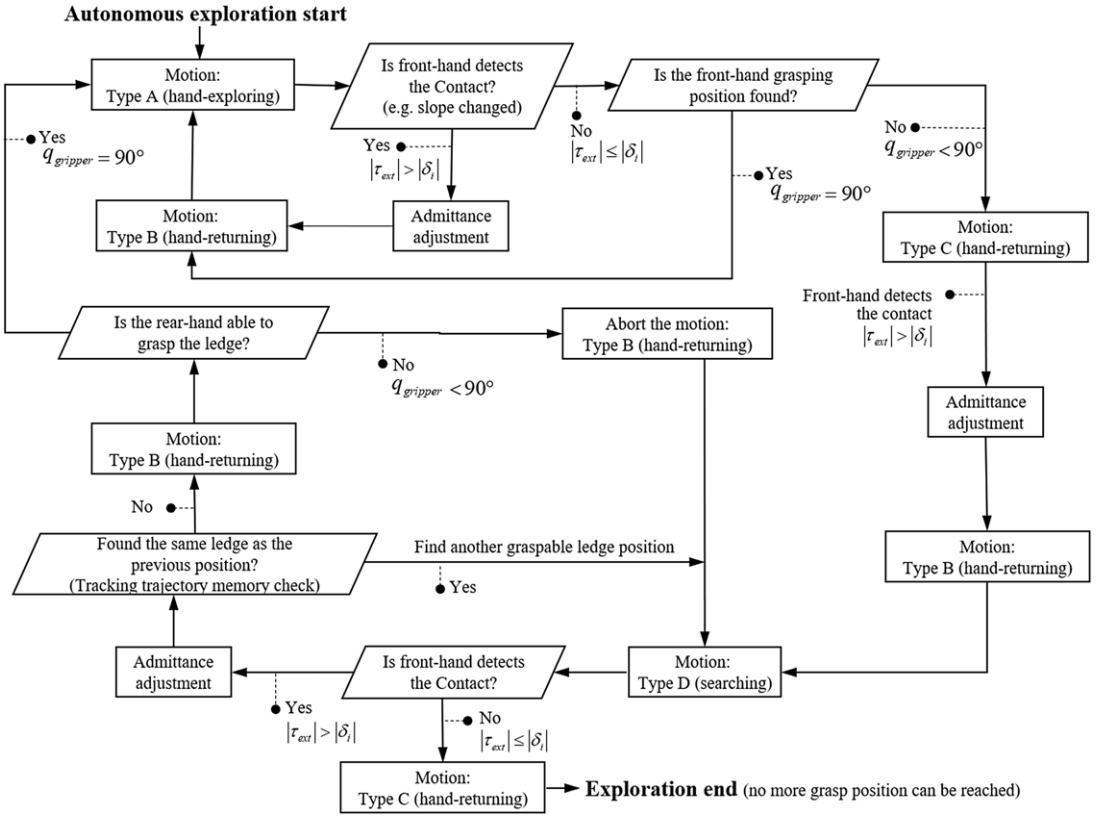


Figure 9. *Flow chart of motion planning and searching strategy.*

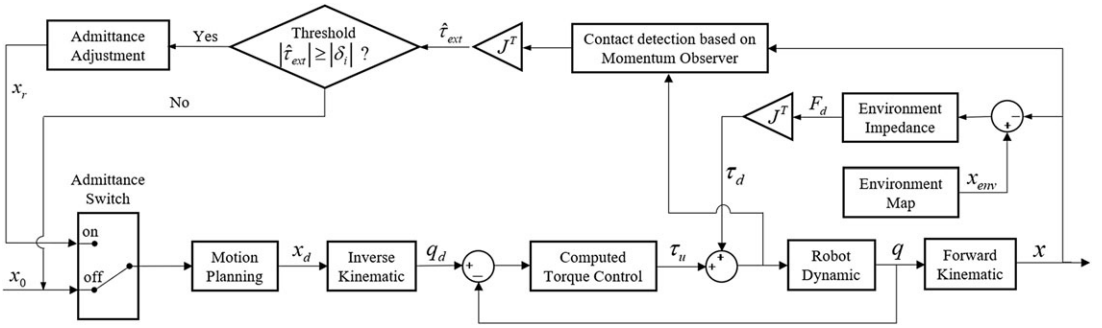


Figure 10. *Control block diagram for transverse climbing robot.*

not detect a contact, when no contact is detected, the admittance switch still preserves the motion planning reference x_0. Therefore, if $|\hat{\tau}_{ext}| > |\delta_i|$, which indicates that the robot has detected a contact, then the admittance adjustment will compute the robot trajectory change reference x_r based on information of F_{ext} and x. Immediately after a contact is detected, the admittance switch will spontaneously switch the reference from x_0 to x_r, allowing the robot's motion planning to adjust to direction changes in the environment.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

6.1. Computed torque control (CTC)

Computed torque control is able to compensate for some limited disturbance forces to be able to maintain robot stability [48]. The fundament architecture of CTC involves an outer and an inner loop control system [49]. The outer loop system employs a proportional derivative to control the desired trajectory parameters, which include the desired joint position, desired velocity, and desired acceleration; and its feedback signal is the actual output trajectory. In the inner loop system, the real-time precise trajectory output is achieved by computing it based on the actual joint torque information (including disturbance torque) and the robot dynamic parameters.

According to Eq. (1), the dynamic model equation without lumped disturbance can be expressed as follows:

$$M(q)\ddot{q} + C\left(q, \dot{q}\right) + G(q) = \tau \tag{28}$$

The robot error dynamic can be expressed as follows:

$$\left(\ddot{q}_d - \ddot{q}\right) + K_d\left(\dot{q} - \dot{q}_d\right) + K_p\left(q - q_d\right) = 0 \tag{29}$$

Then Eq. (29) can be substituted into Eq. (28) to get the joint torque input τ_u:

$$M(q)\left[\ddot{q}_d - K_d\left(\dot{q} - \dot{q}_d\right) - K_p\left(q - q_d\right)\right] + C\left(q, \dot{q}\right) + G(q) = \tau_u \tag{30}$$

If $e = q - q_d$ can be defined as the system error, Eq. (30) can be written as:

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \tag{31}$$

To stabilize the system, the proportional gain K_p and derivative gain K_d need to be determined. The results show that the system output can converge following the reference input. The characteristic Eq. (31) can be written as follows:

$$|sI - A| = s^2 + K_d s + K_p = 0 \tag{32}$$

The characteristics of Eq. (32) are similar to those of the standard second-order system. So, K_p and K_d can be written as follows:

$$\begin{cases} K_p = {\omega_n}^2 \\ K_d = 2\xi\omega_n \end{cases} \tag{33}$$

where ξ is the damping ratio, and ω_n is the natural frequency of the system.

6.2. Environment impedance

An essential element of robot navigation in this research is direct physical contact with its environment. We apply the mechanical impedance principle [50] to calculate the contact force interaction based on the impedance coefficient parameter and the tracking position error. Figure 11 illustrates the contact force interaction between the climbing robot and the environment, based on the mechanical impedance principle [51]. The impedance force function equation of the robot system is then expressed as follows:

$$F_d = m_e\left(\ddot{x}_r - \ddot{x}_e\right) + b_e\left(\dot{x}_r - \dot{x}_e\right) + k_e\left(x_r - x_e\right) \tag{34}$$

where m_e, b_e, and k_e represent the impedance parameters (mass, damping, and stiffness) that interacted with the ledge environments; x_e is the environment contact position (ledge), x_r is the reference position (end-effector), and F_d is the impedance interaction force between the robot and the environment. Figure 11 shows the position of the ledge environment is modeled by a mass-spring-damper system. x_e is obtained based on the x_r when contact occurred, which can be written as the following:

$$\left.\begin{aligned} x_e &= \tfrac{y_r - y_{e1}}{h} + x_{e1} \\ y_e &= h\left(x_r - x_{e1}\right) + y_{e1} \end{aligned}\right\} \tag{35}$$

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

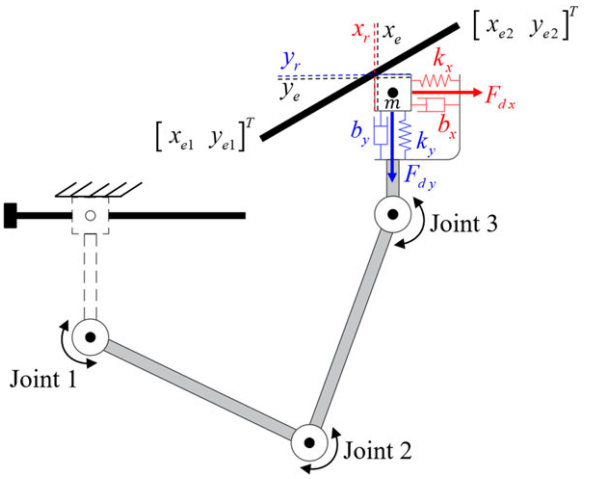


Figure 11. *Illustration of the environment impedance with the three joint climbing robots.*

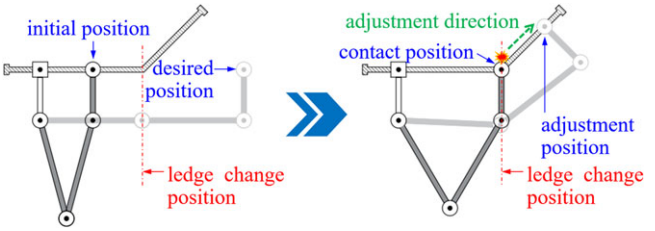


Figure 12. *Illustration of a robot's trajectory adjustment by admittance control in response to the environment change.*

where $[x_{e1}\ y_{e1}]^T$ and $[x_{e2}\ y_{e2}]^T$ are the end tip positions of the ledge, while h is the slope of the ledges and can be written as follows:

$$h = \frac{(y_{e2} - y_{e1})}{(x_{e2} - x_{e1})} \tag{36}$$

6.3. Admittance control

Admittance control is a type of position-based impedance control that aims to establish a dynamic relationship between an external force and the robot's position [52]. The admittance control as illustrated in Fig. 12, allows the robot to make adjustments to changes in its environment.

When the robot receives an external disturbance force $\tau_d \neq 0$, the dynamic equation can be expressed as follows:

$$M_0(q)\ddot{q} + C_0(q, \dot{q})\dot{q} + G_0(q) = \tau + f\,(\tau_{ext}, \rho, t) \tag{37}$$

The admittance control mechanism relates the force feedback to the system's position control [32]. As a result, the system with input $F_{ext}\,(s)$ and output $x_r\,(s)$ can be described as follows:

$$\left(M_a s^2 + B_a s + K_a\right) x_r\,(s) = F_{ext}\,(s) \tag{38}$$

where $x_r\,(s)$ is the revised trajectory of the robot after the system obtains the contact force feedback information with the environment $F_{ext}\,(s)$; therefore M_a, B_a, and K_a are the parameters of inertia, damping, and stiffness between the robot and the environment, respectively. In Eq. (38), the dynamic model is expressed in a joint space. So, to apply the admittance control function, it must be expressed in Cartesian space, like in Eq. (39). So, it can be written as follows:

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

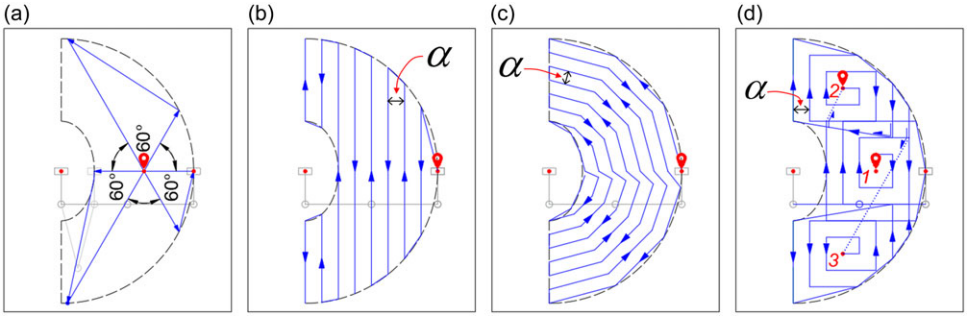


Figure 13. *The grasping position search method for a transverse climbing robot: (a) sector search; (b); creeping line; (c) A∗ algorithm; (d) expanded square.*

$$M(x)\ddot{x} + C(x,\dot{x})\dot{x} + G(x) = F - F_{ext} \tag{39}$$

To obtain a position revision based on contact force feedback information, we need to design the controller input F using equations (39) and (40), which are expressed as follows:

$$F = C(x,\dot{x})\dot{x} + G(x) + M(x)\left\{\ddot{x}_d - M_a^{-1}\left[B_a(\dot{x} - \dot{x}_d) + K_a(x - x_d) + F_{ext}\right]\right\} \tag{40}$$

By applying Eq. (41), the admittance control approach can be achieved, allowing the robot to navigate through a direction change of graspable ledge in a safe and smooth transition.

7. Simulation results and discussion

The effectiveness of transverse climbing robots is evaluated using numerical simulation methods. The robots conducted performance assessments on an obstacle model using four proposed grasp position search methods. The simulation results included evaluating the momentum observer's ability to detect and estimate contact force, evaluating the robot's capability to autonomously perform an exploration through the obstacle model using proposed motion planning and search strategies and then evaluating the effectiveness of the proposed search methods.

7.1. Searching method

The effectiveness of the searching method inspired by search and maritime rescue has been proven in cases where information is provided only in the search zone [33]. We adopt searching methods such as sector search, creeping line, A∗ algorithm, and expanded square. These searching methods are proposed to be modeled in the robot's workspace, as seen in Fig. 13.

The proposed methods have been studied for several reasons, such as being simple in the searching region confined to the robot's workspace [34], having a low computation cost [53], and being acceptable for the specific position of the graspable ledge. Since d is the minimum distance of both hands and α is the span of the searching path line, to preserve the robot's chance of finding a ledge, the α can be expressed as:

$$2\alpha \leq d \tag{41}$$

In this simulation, we systematically assess the performance of each grasping position search method. The assessment criteria are based on the robot's hand-exploration distance [54], the total energy consumption [55], and the time duration necessary for executing the grasping position search operation. The total energy consumption of the transverse climbing robot is expressed as follows [3].

$$E = \int |\tau|\, d\theta \tag{42}$$

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

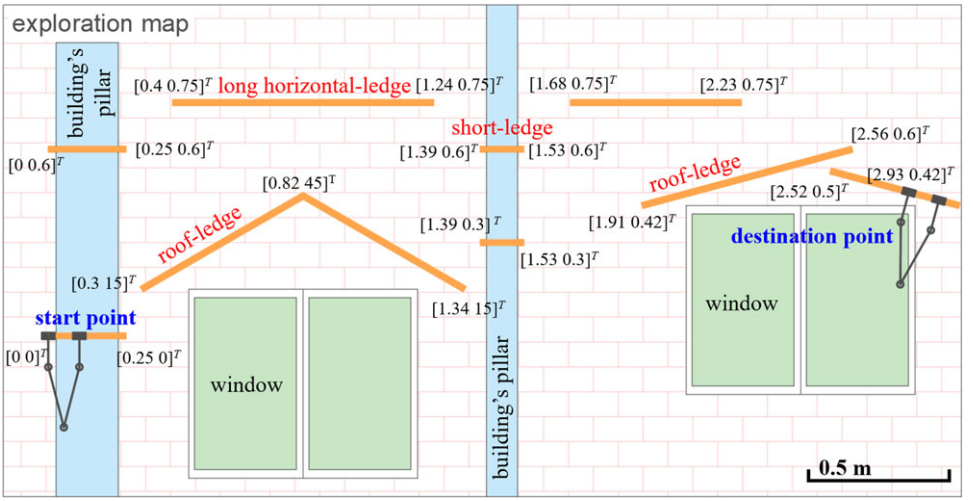


Figure 14. *Obstacle model (environment map), including position coordinates for environment impedance function.*

where E is the total energy consumption, which depends on the joint torque and the angular angle displacement of the robot arm.

7.2. Obstacle model

Figure 14 illustrates an obstacle model that closely represents the actual environment conditions found on a building's exterior. For the challenge, this obstacle model has a complex arrangement of graspable ledge features. In this testing scenario, the robot begins at a designated start point, after which the motion planning and searching methods are activated to explore and navigate toward the endpoint. The ledge coordinate written in the obstacle model is used by the environment impedance function to calculate the contact force. It is important to note that the robot does not have information about the obstacle path during exploration. Instead, using its own navigation strategies, the robot can identify the environment information at each step of its journey.

This scenario in Figure 14 is designed to evaluate the robot's capability to travel by performing transverse climbing, slope change transitioning, and grasping position search procedures with a variety of obstacles. While the robot has traveled through every position on this obstacle, it is important for the robot to estimate the robot's cost of transport (COT) [56]. Each of the robot's searching strategies will result in unique COT results. While COT is defined as the cost of energy consumption necessary per given unit of weight and unit of moving distance [57], it is expressed as follows [3].

$$COT = \frac{E}{mgd} \tag{43}$$

where m is the robot mass, g is the gravity acceleration, and d is the robot work-done distance.

7.3. Investigation of the momentum observer

Figure 15(a) shows the two observers deployed at the robot joint to detect changes in external torque disturbances. In this scenario, the robot does not generate any dynamic force (no internal force disturbance), but an external torque disturbance is applied with a sinusoidal wave reference. The comparison in Fig. 15(b) shows that SOMO is faster to estimate the actual disturbance response than FOMO. Furthermore, Fig. 15(c) shows that SOMO achieves higher estimation accuracy.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

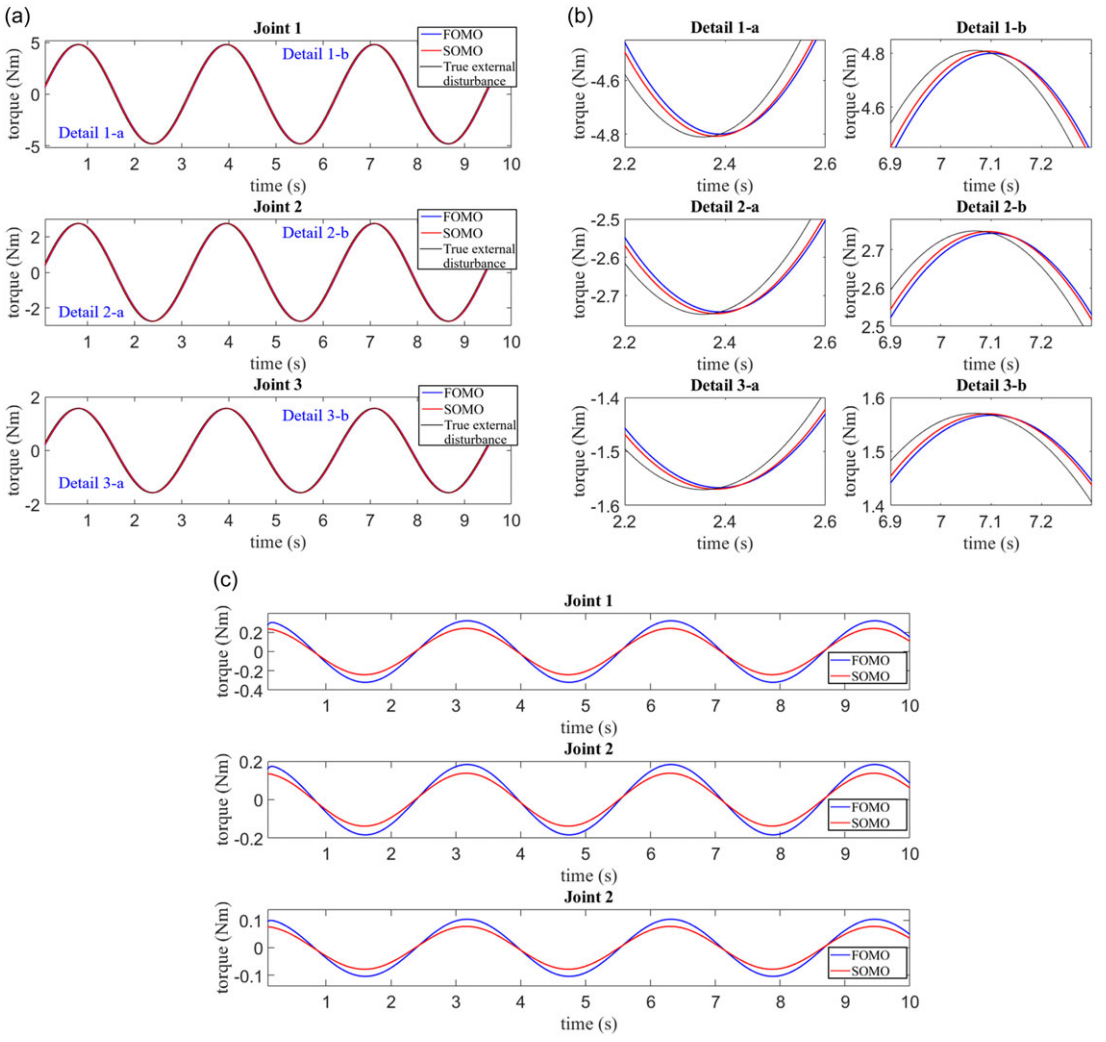


Figure 15. *The performance of first-order momentum observer and second-order momentum observer with sinusoidal wave reference; (a) estimation of external torque; (b) details of the graph; (c) external torque estimation error.*

When using the force feedback approach, dynamic uncertainty is one of the internal disturbances that cause the robot to misdetect contact with the environment during navigation. Since the dynamic uncertainty value cannot be predicted or measured [43], without loss of generality, we assume the dynamic uncertainty deviation is 20% of the modeled dynamic value. Figure 16 demonstrates the robot's dynamic simulation results, where each joint torque observer is integrated using TVT. In this scenario, the robot generates dynamic forces and uncertainty. The sudden external disturbances simultaneously apply to its dynamic system from 2 s to 2.5 s and 6.5 s to 7 s, indicating collision forces. We set the phase shift coefficient and TVT's upper and lower bounds at 1 Nm and −1 Nm, respectively. These findings indicate that when no external disturbance is applied, the robot can tolerate the influence of the uncertainty of the dynamic model. Therefore, the system does not detect it as an external disturbance.

When the estimated external torque applied exceeds the TVT's upper or lower tolerance bounds, the robot detects an external disturbance. In contrast, Fig. 16(b) shows that SOMO detects sudden external torque with a faster response time and faster convergence than FOMO. Another feature noted in Fig. 17 is

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

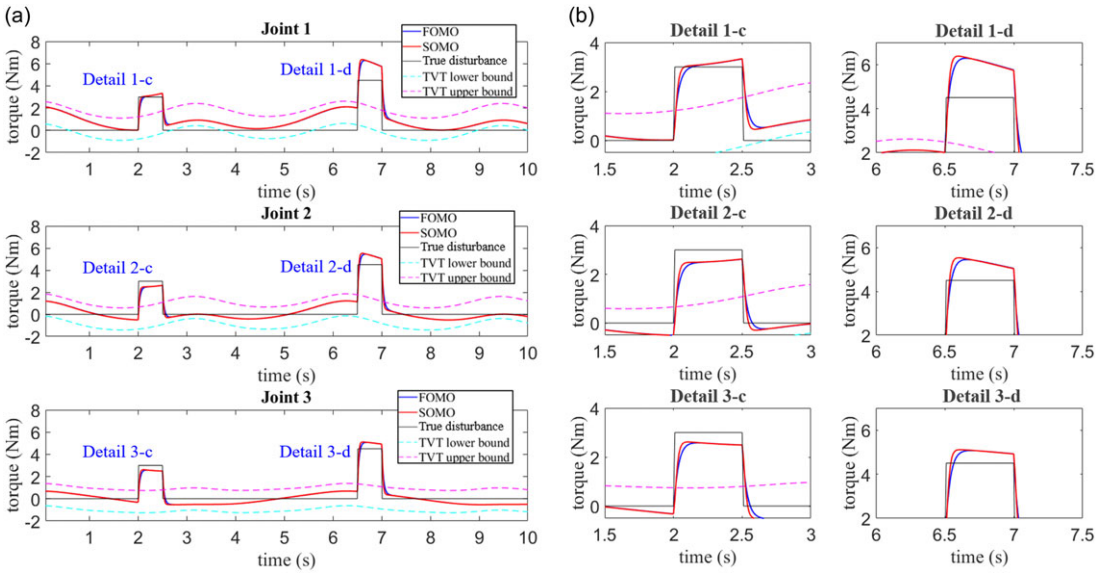


Figure 16. *The performance of first-order momentum observer and second-order momentum observer with step function reference; (a) estimation of external torque integrated with TVT; (b) details of the graph.*

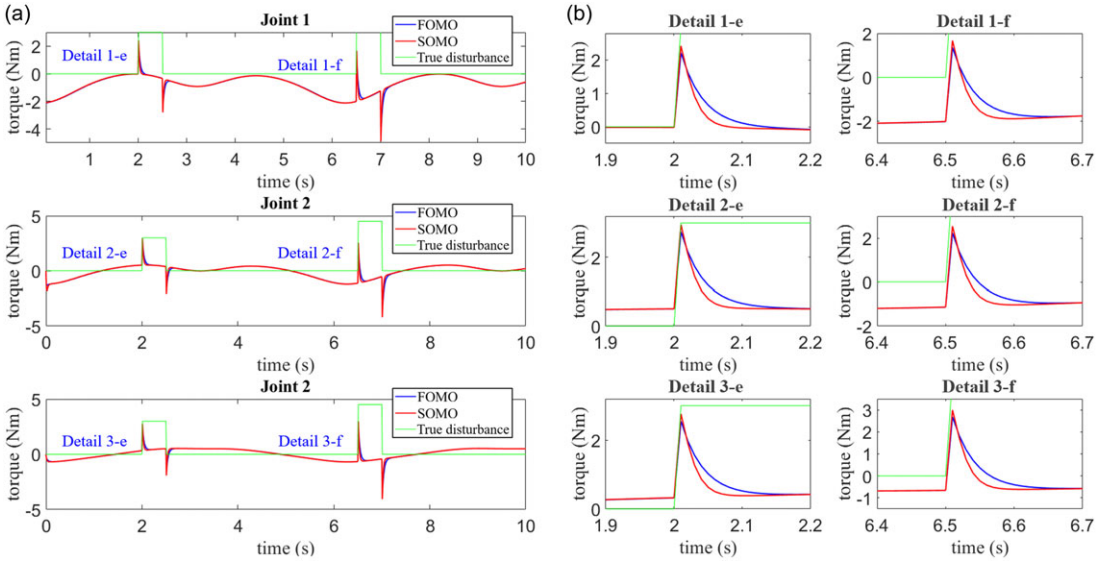


Figure 17. *The performance of first-order momentum observer and second-order momentum observer with step function reference; (a) external torque estimation error; (b) details of the graph.*

that SOMO's estimation error for external joint torque is smaller. Figures 15 and 16 demonstrate that the performance of SOMO is consistently outstanding. A faster and more accurate response for disturbance estimation may increase the robot's safety during navigation [28]. Therefore, based on this evaluation, SOMO is an appropriate alternative to be implemented for the robot as a detector and estimator of the contact force with the environment. However, SOMO is an extended state of FOMO, which requires more computational power.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

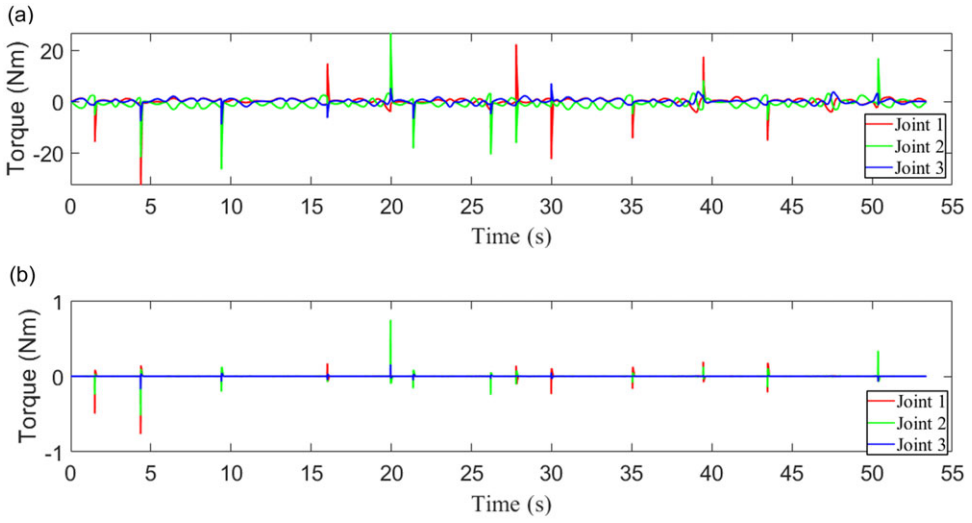


Figure 18. *The performance of second-order momentum observer using a sector search method for exploration. (a) external torque estimation; (b) the estimation errors.*

7.4. Simulation in the model obstacle

In this simulation, motion planning and grasping position search strategies are assessed by allowing a transverse climbing robot to perform exploration through an obstacle model. In a preliminary discussion, we analyze the robot's performance using the sector search approach. Figure 18(a) shows the estimated external joint torque using SOMO, indicating that the robot completed its exploration of the obstacle model in 53.39 s. These spikes indicate a disturbance received by the robot, which should be integrated with the TVT to establish the categorization of the external or internal disturbance. Figure 18(b) shows that the estimated error is obtained based on the difference between the external joint torque estimation and the actual disturbance force. When the robot is in a condition without disturbance, the error is relatively low (nearly 0 Nm), but when the robot gets disturbed, the system suffers a slight inaccuracy when estimating the disturbance force. This suggests that SOMO is operating without difficulties. The simulation video demo can be obtained by visiting this web link: https://youtu.be/9jyP6XNLWps (Accessed on January 12, 2025).

A technique for detecting an external disturbance or contact force between the robot and the environment is developed by integrating the SOMO and TVT, as presented in Fig. 19. When the joint torque estimation exceeds the TVT tolerance threshold, the system will detect it as an external disturbance. The results show that the robot has identified a number of contact forces during the obstacle model exploration, each of which is designated by the symbols "a" to "m," and its explanation of denotation is provided in Table III.

Figure 20 illustrates the movement and locomotion trace of a transverse climbing robot during exploration through the obstacle model. The robot moved forward from the beginning point to the end destination point with a work-done distance of 3.451 m by navigating autonomously without detecting the current condition of the environment in advance. According to Table III, the robot has completed the environment transition through a six-time grasping position search and a one-time ledge slope change. A proposed motion planning and searching strategy has worked properly according to its desired function. As proven during the searching procedure, the robot detects two contact forces: firstly, to determine the ideal position for executing the searching procedure, and secondly, when it finds the unknown graspable position during the searching procedure.

Following Fig. 20, the robot completed a search procedure across three different ledge gap characteristics. The first characteristic is the upper-ledge gap position, as demonstrated by the 1st and 2nd searches.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

Table III. *Identification of recognized external disturbances.*

Notation	Time (s)	Description	Notation	Time (s)	Description
a	1.52	Get 1st searching position	h	27.81	Get 4th searching position
b	4.38	1st grasping ledge found	i	30.00	4th grasping ledge found
c	9.42	slope change detected	j	35.07	Get 5th searching position
d	16.03	Get 2nd searching position	k	39.50	5th grasping ledge found
e	19.97	2nd grasping ledge found	l	43.49	Get 6th searching position
f	21.39	Get 3rd searching position	m	50.39	6th grasping ledge found
g	26.21	3rd grasping ledge found			

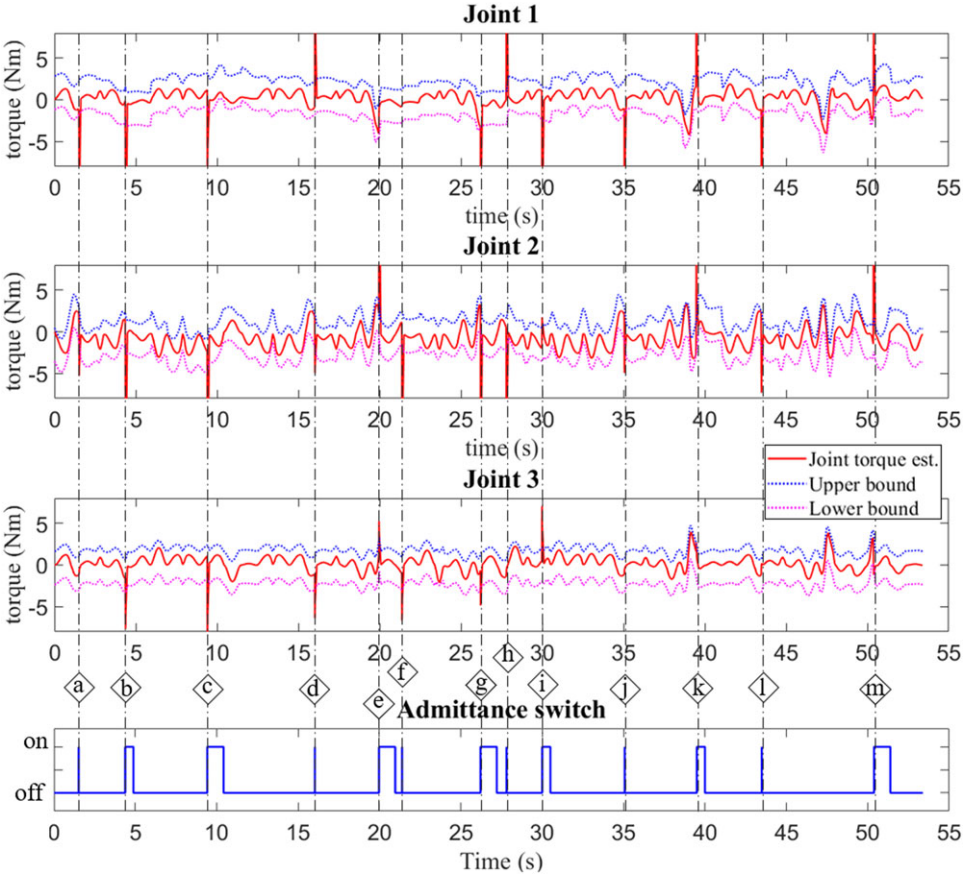


Figure 19. *TVT integrated with second-order momentum observer external joint torque estimation and admittance switch.*

If it's linked to the actual environment, the robot can climb upward to a horizontal ledge where the position is higher than the robot. The second one is the short-ledge feature gap position characteristics, as demonstrated by the 3rd and 4th searches. The last one is the lower-ledge gap position characteristics, as demonstrated in the 5th and 6th searches. Where the robot may climb downwards to the ledge where its position is lower than the robot. In addition to executing these searching procedures, the robot is capable of making transitions across a slope change on the connected ledge. For example, a ledge is designed like a roof.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

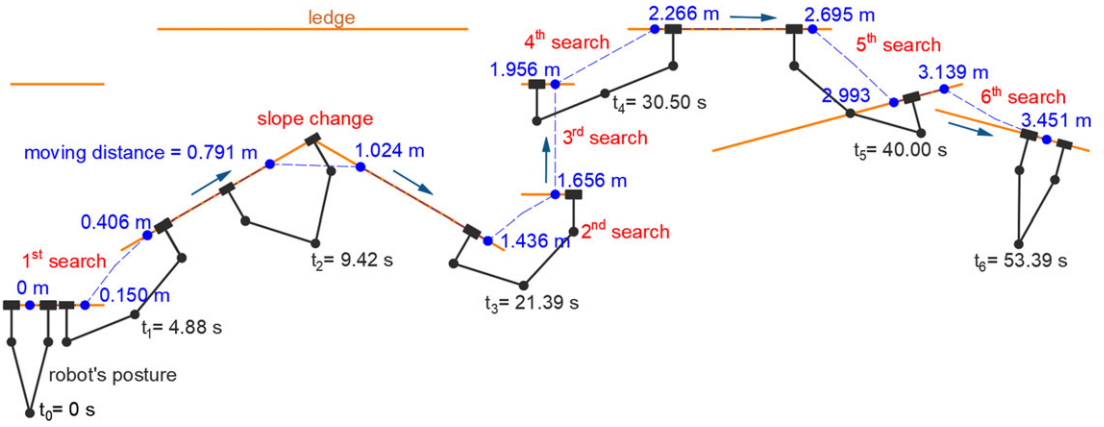


Figure 20. *A transverse movement trace illustrations of a transverse climbing robot's, including work–done distance, slope change, and grasping position search. It can be demonstrated in the simulation video by accessing this web link: https://youtu.be/9jyP6XNLWps (Accessed on January 12, 2025).*

Immediately after the robot detects the contact force with the environment, an automatic admittance switch will respond spontaneously to switch the trajectory reference to the admittance control. Admittance control computes a revised trajectory due to the ledge direction shift based on the estimated contact force given by the observer, resulting in a new motion planning reference. In Fig. 21, it can be noted that when the system detects a contact force (for example, when the grasping ledge was recognized during the searching procedure), the admittance control revises the robot joint angle to adjust it to the change in environment. As a result, the spike in robot joint torque caused by contact spontaneously declines within the TVT tolerance limit. Returning to Fig. 19, it is suggested that an automated admittance system has an essential contribution in assisting in minimizing the amount of collision force acquired by the robot, which may reduce the risk of serious damage to the robot. Based on the algorithm of motion planning and searching strategy, an admittance switch will revert to a disabled state once the admittance adjustment action has been completed.

7.5. Performance of grasping position search method

The present study assesses four different methods of grasping position search. Figure 22 shows the transverse climbing robot's hand-exploration and hand-returning trace using a variety of searching methods. All of the methods have successfully made the robot travel through the obstacle model for the identical total work-done distance of 3.45 m, but each of them has various performance characteristics. Figure 23(a) illustrates the performance of the exploration time relative to the total work-done distance, where the sector search method completes the task in the fastest time, which is 53.39 s. All of the other methods have significant delays in the 2nd and 3rd searches (short-ledge feature positions), whereas the sector search method only has slight delays in the 5th and 6th searches (lower-ledge positions).

Figure 23(b) illustrates the hand-exploration distance relative to the work-done distance, revealing that the sector search method requires the shortest exploration path distance, which is 10.13 m. All of the other methods require a considerably increased exploration path distance for the 2nd and 3rd searches, which causes the exploring time to get longer. If we observe the exploration path of the sector search method in Fig. 22(a), its searching path is not simply focused on one specified location that must be finished early on. As a result, the sector search method is getting faster to reach the workspace's center region, which makes it easier to execute searching procedures in the short-ledge feature position.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

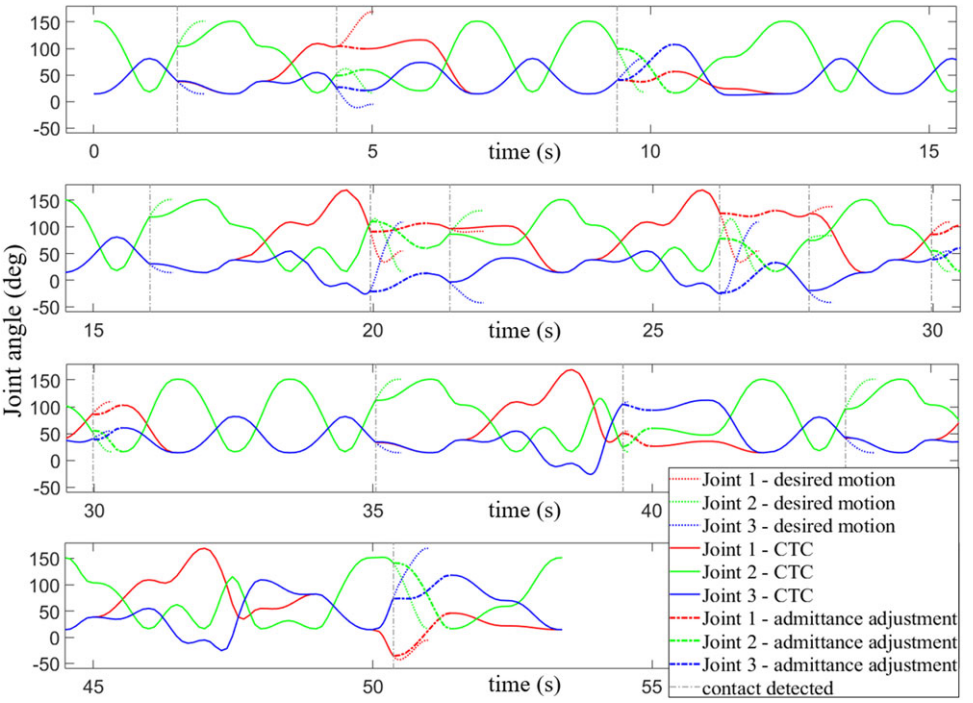


Figure 21. *Robot joint control results.*

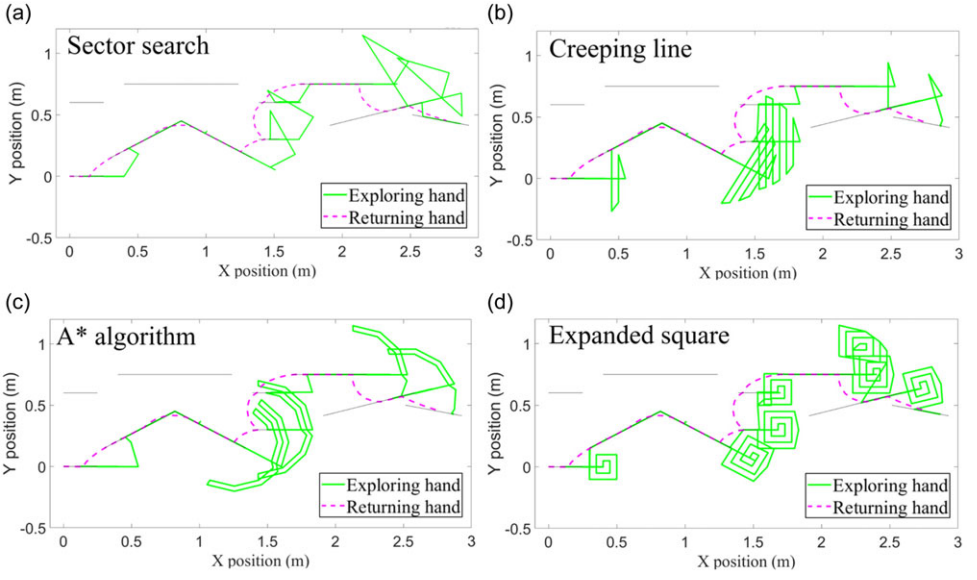


Figure 22. *An exploration trace of a transverse climbing robot by using its grasping position search method; (a) sector search; (b) creeping line; (c) A∗ algorithm; (d) expanded square.*

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

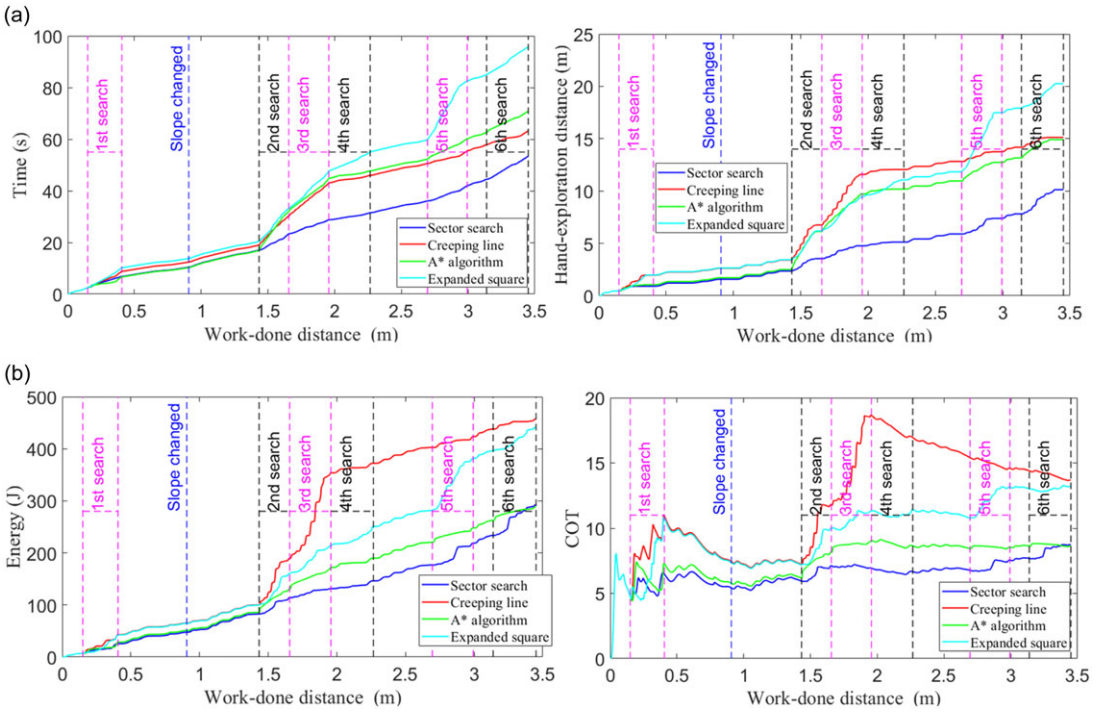


Figure 23. *Performance of grasping position search method; (a) exploration time; (b) hand-exploring distance; (c) energy consumption; (d) cost of transport.*

Figure 23(d) illustrates that the A∗ algorithm consumes the least amount of total energy. Although the A∗ algorithm also has difficulties in the 2nd and 3rd searches, it just spends the least amount of energy compared to the creeping line and extended square. This is because the searching path of the A∗ algorithm focuses on tracking the workspace boundary line. Therefore, after reaching the upper boundary location, it will track to the lower boundary, or vice versa. This is distinct from the creeping line method, which has spent the largest amount of energy in the 2nd and 3rd searches due to its searching path line focusing on tracing the higher and lower parts of the workspace practically at the same time by following the horizontal centerline of the workspace. However, because of its searching path characteristics, the creeping line method is the most successful for finding the lower ledge position. Figure 23(c) illustrates the cost of transport (COT), with the A∗ algorithm being the least costly compared to the other methods. The cost of transport is related to the amount of energy consumed per unit of work-done distance.

The next assessment is to identify the most effective grasping position search method for each gap characteristic between graspable ledges. Table IV(a) presents the searching performance on the upper-ledge position (at the 1st and 4th searches), where the sector search method is the most effective in terms of searching time, hand-exploration distance, and energy consumption. Next is Table IV(b), which presents its searching performance on the short-edge features position (at the 2nd and 3rd searches), where the sector search method still has outstanding performance in the same criteria. Finally, Table IV(c) presents the search performance on the lower-ledge position (at the 5th and 6th searches), where the creeping line method is the most effective in terms of searching time, hand-exploration distance, and energy consumption. Meanwhile, sector search is ranked third.

The four types of grasping position search methods that have been assessed in this study have varying performances in each aspect of the gaps between grasping ledges. So, in practice, it is difficult to integrate more than one grasping position search method during a single environment exploration.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

Table IV. *Performance of the grasping position search method based on characteristics of the obstacle; (a) upper-ledge position; (b) short-ledge features; (c) lower-ledge position.*

(a)

		1st Search			4th Search		
Rank	**Searching method**	**Time (s)**	**Searching dist. (m)**	**Energy (J)**	**Time (s)**	**Searching dist. (m)**	**Energy (J)**
1st	Sector search	2.36	0.46	8.02	2.69	0.36	14.62
2nd	A∗ algorithm	2.21	0.61	11.18	2.85	0.47	17.83
3rd	Creeping line	4.34	1.52	25.61	2.92	0.47	18.44
4th	Expanded square	5.74	1.54	25.02	7.50	1.62	35.57

(b)

		1st Search			4th Search		
Rank	**Searching method**	**Time (s)**	**Searching dist. (m)**	**Energy (J)**	**Time (s)**	**Searching dist. (m)**	**Energy (J)**
1st	Sector search	2.36	0.46	8.02	2.69	0.36	14.62
2nd	A∗ algorithm	2.21	0.61	11.18	2.85	0.47	17.83
3rd	Expanded square	4.34	1.52	25.61	2.92	0.47	18.44
4th	Creeping line	5.74	1.54	25.02	7.50	1.62	35.57

(c)

		5th Search			6th Search		
Rank	**Searching method**	**Time (s)**	**Searching dist. (m)**	**Energy (J)**	**Time (s)**	**Searching dist. (m)**	**Energy (J)**
1st	Creeping line	2.85	0.93	14.52	3.29	0.94	14.25
2nd	A∗ algorithm	5.87	1.78	21.31	6.19	1.75	21.14
3rd	Sector search	3.93	1.49	35.73	6.90	2.31	53.85
4th	Expanded square	20.7	5.64	93.89	8.7	2.34	39.97

Because, in fact, the robot does not have any information about what's occurring in front of it when exploring the environment. A future study focusing on developing an optimization of its searching method could be a solution to this challenge. However, based on the discussion of this present study, we make a decision on how to determine the compatibility of the grasping position search method by considering the following desires: First, for faster exploration time, we recommend using the sector search method. This can be examined in Fig. 21(a), where the rise in exploration time is substantially more stable during the searching and travel climbing procedures. The sector search method is also recommended for a shorter exploration path length, as can be noticed in Fig. 21(b). Furthermore, to reduce energy consumption and COT, we recommend using the A∗ algorithm. This can be observed in Fig. 21(d), where the increase in consumption of energy seems remarkably consistent. In addition, if assessed on the searching performance of all gap characteristics between the grasping ledges, which is presented in Table IV(a), (b), and (c), the A∗ algorithm is consistently in the second rank in terms of energy consumption.

7.6. Comparative review of visual-based and contact force-based navigation

This Section presents a comparative review of the four-link transverse ledge climbing robot, comparing the contact force-based navigation method developed in this research with the visual-based navigation method introduced by Lin et al. [2]. This comparison highlights how the contact-force-based approach addresses the limitations of the visual-based method. The results of this comparison analysis are summarized in Table V.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

Table V. *Comparison of visual-based and force-based navigation methods for a four-link ledge transverse climbing robot.*

Description	Visual sensor-based	Contact force-based
Range of sensing	Limited due to the static view angle	Greater range with continuous dynamic contact sensing
Obstacle capability (ledges arrangement)	Connected slope	Connected slope; gap with: same altitude, different altitude, and different slope
Sensitivity of sensing	Affected by ledge surface properties [13]	Influenced by the properties of environment impedance
Environment conditions	Smoky [14], foggy [15], and lighting intensity [18] affect an accuracy	No affects are reported in this finding
Agility	Disrupted by the installation of sensors	No robot's agility disruption reported in this finding

The climbing robot developed in this research can perform exploratory movements across its workspace by dynamically and repeatedly attempting contact with the environment, resulting in a wider sensing range for a variety of obstacles. If the robot interacts with a rigid environment, it can acquire an optimal contact impedance force for admittance control references to accurately calculate the robot's position as the environment changes. So, the environment's surface reflection does not affect the contact impedance force. By utilizing momentum observers, the robot's space requirements for the installation of additional sensor devices that may limit the robot's movement can be eliminated, allowing for more agile movement and enhanced adaptation to complex environments.

8. Conclusion

This paper presents the development of an autonomous navigation system that utilizes contact force feedback to realize the proposed motion planning and searching strategies for a transverse ledge climbing robot. Through simulation verification, our approach has successfully enabled the robot to perform real-time posture adjustments based on force interactions with the environment. This capability facilitates smooth transitions and stability, ensuring effective and safe navigation across various environment conditions. The main contributions from this research are as follows:

1. The motion planning strategy coordinates the locomotion templates to provide trajectory references based on the robot's current posture and environment situation.
2. Admittance control calculates trajectory corrections in response to environment changes. The automatic admittance switch provides a spontaneous response to trigger the control admittance when the SOMO exceeds the TVT limit. This combination contributes to the robot's safety.
3. We suggest using the sector search method to minimize exploration time and search distance. However, to minimize energy consumption and COT, we suggest switching to the A$*$ algorithm method.

Additionally, several approaches have been integrated to support the robot's function and avoid potential failures. The CTC aligns the output trajectory with the input reference, ensuring stability against uncertain dynamics and limited external disturbances. As a contact force estimator, SOMO improves response time, convergence, and accuracy, making it ideal for robust control systems in climbing robots. The integration of TVT into SOMO creates an effective contact detection system that characterizes disturbances caused by both internal and external forces.

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

By combining these contributions and approaches, the system enhances the robot's performance in complex environments, as validated by our results compared to previous methods using visual sensors [9]. Based on these findings, future research could focus on further optimizing the searching methods.

Author contributions. Reno Pangestu and Chi-Ying Lin conceived and designed the study. Reno Pangestu wrote the article, established the methodology, conducted a numerical simulation, and analyzed the data. Guan Xian Yu compiled the literature and derived several equations. Chi-Ying Lin supervised this study.

Financial support. MOST 111-2628-E-011-011

Competing interests. The authors declare no competing interests exist.

Ethical approval. Not applicable.

References

[1] C.-Y. Lin and Y.-J. Tian, "Design of transverse brachiation robot and motion control system for locomotion between ledges at different elevations," *Sensors* **8**(11), 343 (2021).
[2] C.-Y. Lin and T.-C. Hu, "Locomotion control of ledge climbing robot using CPG control and infrared sensors," *Int. J. iRobot.* **4**(4), 40–49 (2022).
[3] C.-Y. Lin and J.-M. Lee, "Multi-locomotion design and implementation of transverse ledge brachiation robot inspired by sport climbing," *Biomimetics* **8**(2), 204 (2023).
[4] Y. Wang, H. Jiang, T. M. Huh, D. Sun, W. Ruotolo, M. W. Miller, W. R. T. Roderick, H. Stuart and M. R. Cutkosky, "SpinyHand: Contact load sharing for a human-scale climbing robot, ASME," *J. Mech. Robot.* **11**(3), 031009 (2019).
[5] F. Y. Y. Ling, M. Liu and Y. C. Woo, "Construction fatalities in Singapore," *Int. J. Project Manag.* **27**(7), 717–726 (2009).
[6] L. Patané, "Bio-inspired robotic solutions for landslide monitoring," *Energies* **12**(7), 1256 (2020).
[7] T. Seo, Y. Jeon, C. Park and J.-W. Kim, "Survey on glass and façade-cleaning robots: Climbing mechanisms, cleaning methods, and applications," *Int. J. Precis. Eng. Manuf.-Green Tech.* **6**(2), 367–376 (2019).
[8] Z. Zheng, N. Ding, H. Chen, X. Hu, Z. Zhu, X. Fu, W. Zhang, L. Zhang, S. Hazken, Z. Wang and M. Zhao "CCRobot-V: A Silkworm-Like Cooperative Cable-Climbing Robotic System for Cable Inspection and Maintenance," **In:** *2022 International Conference on Robotics and Automation (ICRA)* (2022) pp. 167–170.
[9] D. Wei and W. Ge, "Research on one bio-inspired jumping locomotion robot for search and rescue," *Int. J. Adv. Robot. Syst.* **11**(10), (2014).
[10] M. P. Austin, M. Y. Harper, J. M. Brown, E. G. Collins and J. E. Clark, "Navigation for legged mobility: Dynamic climbing," *IEEE Trans. Robot.* **36**(2), 537–544 (2020).
[11] K. S. Fu, R. C. Gonzalez and C. S. G. Lee, "*Robotics: Control, Sensing, Vision, and Intelligence*. vol. **1** (McGraw-Hill, New York, 1987).
[12] Z. Ming, Y. Ma, L. Zhan, J. He and Y. Liu, "Façade protrusion recognition and operation-effect inspection methods based on binocular vision for wall-climbing robots," *Appl. Sci.* **13**(9), 5721 (2023).
[13] P. M. Novotny and N. J. Ferrier, "Using Infrared Sensors and the Phong Illumination Model to Measure Distances," **In:** *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, **2**, (1999) pp. 1644–1649.
[14] G. Benet, F. Blanes, J. E. Simó and P. Pérez, "Using infrared sensors for distance measurement in mobile robots," *Robot. Auton. Syst.* **40**(4), 255–266 (2002).
[15] J. W. Starr and B. Y. Lattimer, "A comparison of IR stereo vision and LIDAR for use in fire environments," *Sensors*, 1–4 (2012).
[16] J. W. Giles, I. N. Bankman, R. M. Sova, W. J. Green, T. R. M. King, F. J. Marcotte, D. D. Duncan and J. A. Millard. "General-purpose lidar system model with experiment validation in fog, oil, and smoke conditions," *Proc. SPIE Int. Soc. Optical Eng. Proc. SPIE*, **4484**, 178–185 (2002).
[17] J.-H. Kim, J.-C. Lee and Y.-R. Choi, "PiROB: Vision-based pipe-climbing robot for spray-pipe inspection in nuclear plants," *Int. J. Adv. Robot. Syst.* **15**(6), (2018).
[18] H. Lu, H. Zhang, S. Yang and Z. Zheng, "Camera Parameters Auto-adjusting Technique for Robust Robot Vision," **In:** *2010 IEEE International Conference on Robotics and Automation* (2010) pp. 1518–1523.
[19] R. S. Da Veiga, A. S. De Oliveira, L. V. R. De Arruda and F. Neves, "Localization and Navigation of a Climbing Robot Inside a LPG Spherical Tank Based on Dual-LIDAR Scanning of Weld Beads," **In:** A. Koubaa (ed.), *Robot Operating System (ROS). Studies in Computational Intelligence, 625* (2016).
[20] B. Saund and D. Berenson, "Motion planning for manipulators in unknown environments with contact sensing uncertainty," *Int. Conf. Robot. Autom. (ICRA)* **11**, 461–474 (2020).
[21] B. Saund, S. Choudhury, S. Srinivasa and D. Berenson, "The blindfolded robot: A Bayesian approach to planning with contact feedback," *Springer Proc. Adv. Robot.* **20**, 443–459 (2022).

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

[22] N. Khedekar, F. Mascarich, C. Papachristos, T. Dang and K. Alexis, "Contact-based Navigation Path Planning for Aerial Robots," **In:** *2019 International Conference on Robotics and Automation (ICRA*, (2019) pp. 4161–4167.

[23] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba and H. Inoue, "Motion planning for humanoid robots," *Springer Tracts Adv. Robot.* **15**, 365–374 (2005).

[24] P. Kanajar, D. G. Caldwell and P. Kormushev, "Climbing Over Large Obstacles with a Humanoid Robot via Multi-contact Motion Planning," **In:** *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (2017) pp. 1202–1209.

[25] J. Luo, Y. Zhang, K. Hauser, H. A. Park, M. Paldhe, C. S. G. lee, M. Grey, M. Stilman, O. Jun Ho, J. Lee, I. Kim and P. Oh "Robust Ladder-climbing with a Humanoid Robot with Application to the DARPA Robotics Challenge," **In:** *2014 IEEE International Conference on Robotics and Automation (ICRA)*, (2014) pp. 2792–2798.

[26] L. Ding, H. Xing, H. Gao, A. Torabi, W. Li and M. Tavakoli, "VDC-based admittance control of multi-DOF manipulators considering joint flexibility via hierarchical control framework," *Control Eng. Practice* **124**, 105186 (2022).

[27] J. C. Martínez-Rosas and M. A. Arteaga, "Force and velocity observers for the control of cooperative robots," *Robotica* **26**(1), 85–92 (2008).

[28] Y. Li, Y. Li, M. Zhu, Z. Xu and D. Mu, "A nonlinear momentum observer for sensorless robot collision detection under model uncertainties," *Mechatronics* **78**, 102603 (2021).

[29] T. Li, H. Xing, E. Hashemi, H. D. Taghirad and M. Tavakoli, "A brief survey of observers for disturbance estimation and compensation," *Robotica* **41**(12), 3818–3845 (2023).

[30] P. Cao, Y. Gan and X. Dai, "Model-based sensorless robot collision detection under model uncertainties with a fast dynamics identification," *Int. J. Adv. Robot. Syst.* **16**(3), 1–15 (2019).

[31] S. Long, X. Dang, S. Sun, Y. Wang and M. Gui, "A novel sliding mode momentum observer for collaborative robot collision detection," *Machines* **10**(9), 818 (2022).

[32] T. Fujiki and K. Tahara, "Series admittance-impedance controller for more robust and stable extension of force control," *ROBOMECH J.* **9**(23), (2022).

[33] D. A. Otote, B. Li, B. Ai, S. Gao, J. Xu, X. Chen and L. G., "A decision-making algorithm for maritime search and rescue plan," *Sustainability-Basel* **11**(7), 2084 (2019).

[34] J. Jiang, L. Yao, Z. Huang, G. Yu, L. Wang and Z. Bi, "The state of the art of search strategies in robotic assembly," *J. Ind. Inf. Integr.* **26**, 100259 (2022).

[35] Y. Joo, H. Kim, J. Park, C. Shin, H. Kwak, J. Song and C. Shin "Sensorless Force Control Algorithm Based on Momentum Observer Technique," **In:** *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 1747–1751 (2021).

[36] W. Li, Y. Han, J. Wu and Z. Xiong, "Collision detection of robots based on a force/torque sensor at the bedplate," *IEEE/ASME Trans. Mechatron.* **25**(5), 2565–2573 (2020).

[37] B. Varga, J. K. Tar and R. Horváth, "Fractional order inspired iterative adaptive control," *Robotica* **42**(2), 482–509 (2024).

[38] Official Website of the International Federation of Sport Climbing. Available online: https://www.ifsc-climbing.org/. Accessed January 23, 2025.

[39] C.-Y. Lin. and Z.-H. Yang, "TRBR: Flight body posture compensation for transverse ricochetal brachiation robot," *Mechatronics* **65**, 102307 (2020).

[40] M. M. Olsen and H. G. Petersen, "A new method for estimating parameters of a dynamic robot model," *IEEE Trans. Robot. Autom.* **17**(1), 95–100 (2001).

[41] M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control* (John Wiley & Sons, United States, 2020).

[42] L.-W. Tsai. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators* (John Wiley & Sons, Canada, 1999).

[43] Z. Song, J. Yi, D. Zhao and X. Li, "A computed torque controller for uncertain robotic manipulator systems: Fuzzy approach," *Fuzzy Sets Syst.* **154**(2), 208–226 (2005).

[44] T. D. Le, H.-J. Kang, Y.-S. Suh and Y.-S. Ro, "An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator," *Neurocomputing* **116**, 53–61 (2013).

[45] J. H. Davis, "Luenberger observers," *Found. Determin. Stochast. Control*, 245–254 (2002).

[46] P. Johan, J. T. Gravdahl and K. Y. Pettersen*Vehicle-Manipulator Systems. Advances in Industrial Control* (Springer-Verlag, United Kingdom, 2014).

[47] T. Ren, Y. Dong, D. Wu and K. Chen, "Collision detection and identification for robot manipulators based on extended state observer," *Control Eng. Pract.* **17**, 144–153 (2018).

[48] M. A. Llama, R. Kelly and V. Santibanez, "Stable computed-torque control of robot manipulators via fuzzy self-tuning," *IEEE Trans. Syst. Man Cybernet. Part B (Cybernetics)* **30**(1), 143–150 (2000).

[49] W. Shang and S. Cong, "Nonlinear computed torque control for a high-speed planar parallel manipulator," *Mechatronics* **19**(6), 987–992 (2009).

[50] H. F. N. Al-Shuka, S. Leonhardt, W.-H. Zhu, R. Song, C. Ding and Y. Li, "Active impedance control of bioinspired motion robotic manipulators: An overview," *Appl. Bionics Biomech.* **2018**, 1–19 (2018).

[51] T. Yoshikawa. *Foundations of Robotics: Analysis and Control* (MIT Press, 1990).

[52] C. Liu and Z. Li, "Force tracking smooth adaptive admittance control in unknown environment," *Robotica* **41**(7), 1991–2011 (2023).

[53] A. Majeed and S. O. Hwang, "A multi-objective coverage path planning algorithm for UAVs to cover spatially distributed regions in urban environments," *Aerospace* **22**(11), 4031 (2021).

[54] F. Niroui, K. Zhang, Z. Kashino and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robot. Autom. Lett.* **4**(2), 610–617 (2019).

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press

[55] Y. Mei, Y.-H. Lu, Y. C. Hu and C. S. G. Lee, "Deployment of mobile robots with energy and timing constraints," *IEEE Trans. Robot. Autom. Robot. (ICCAR)* **22**(3), 507–522 (2006).
[56] O. Wongwirat and A. Anuntachai, "Searching Energy-efficient Route for Mobile Robot with Ant Algorithm," **In:** *IEEE 2011 11th International Conference on Control, Automation and Systems, Gyeonggi-do, Korea (South)* (2006) pp. 507–522.
[57] J. Nishii, "An analytical estimation of the energy cost for legged locomotion," *J. Theor. Biol.* **238**(3), 636–645 (2006).
[58] G. Garofalo, N. Mansfeld, J. Jankowski and C. Ott, "Sliding Mode Momentum Observers for Estimation of External Torques and Joint Acceleration," **In:** *2019 International Conference on Robotics and Automation (ICRA)* (2019) pp. 6117–6123.

Cite this article: R. Pangestu, G. X. Yu and C.-Y. Lin (2025). "Motion planning and searching strategy of a transverse ledge climbing robot based on force feedback", Robotica **43**, 997–1026. https://doi.org/10.1017/S0263574725000062

https://doi.org/10.1017/S0263574725000062 Published online by Cambridge University Press