Journal of Robotics, Networking and Artificial Life
Vol. 11(2); December (2025), pp. 160-169
ISSN (Online): 2352-6386; ISSN (Print): 2405-9021
https://alife-robotics.org/jrnal.html

Research Article

Flipper Control for Crawler Robots on Unstructured Terrain Using 2.5D Terrain Maps

Kotaro Kanazawa[1*], Noritaka Sato[1], Yoshifumi Morita[1]
[1] *Faculty of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan*
Email: *k.kanazawa.837@nitech.jp*, *sato.noritaka@nitech.ac.jp*, *morita@nitech.ac.jp*
[*] Corresponding Author

ARTICLE INFO

Article History

Received 25 November 2024
Accepted 20 October 2025

Keywords

Flipper control
Rescue robot
Crawler robot

ABSTRACT

This study presents a flipper control method for crawler robots based on terrain geometry, without requiring sequential optimization. A filtered 2.5D terrain map was used to determine the target surfaces and automatically switch between the driving and traversing modes according to the terrain, extending the step-climbing sequences to 3D unstructured environments. By predicting the contact between the robot and terrain, all four flippers were independently controlled to achieve collision avoidance and posture stabilization during arbitrary movements, including turning. Simulations of step climbing, stair climbing, and oblique-step traversal demonstrated that the method suppressed excessive pitch and roll, avoided collisions, eliminated manual flipper operation, and expanded the operational range of crawler robots.

1. Introduction

Remotely operated rescue robots are indispensable for effective information gathering in disaster environments. Disaster sites are typically cluttered with rubble and characterized by uneven terrain, requiring locomotion mechanisms that can adapt to irregular surfaces. To meet this demand, many rescue robots employ crawler-based platforms equipped with four articulated sub-crawlers, commonly referred to as flippers. Figure 1 shows a crawler robot developed by authors. The crawler robot, which is equipped with crawler belts on both the main body and the flippers, and whose belts on each side are driven synchronously, can be considered a nonholonomic system. The flippers can be independently controlled to change their angles, allowing the robot to overcome large obstacles, maintain stability through postural adjustment, and reduce impact forces during descent, thereby mitigating the risk of structural damage. However, predicting environmental conditions in disaster scenarios is inherently difficult, making flexible adaptation to unforeseen situations essential. Despite ongoing advances, most systems continue to rely heavily on manual teleoperation. In such cases, operators must simultaneously manage locomotion, manipulator operation, and the independent motion of the four flipper joints based on camera and sensor feedback, imposing a substantial cognitive load.

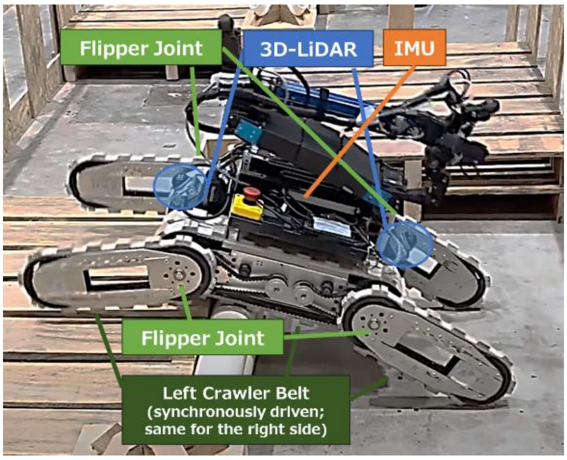

Figure 1 Conceptual hardware configuration of the crawler robot. The robot is equipped with crawler belts, four flipper joints, a 3D LiDAR, and an IMU.

Crawler robots must maintain an appropriate body posture while traversing irregular terrain. Many studies have proposed methods that use LiDAR or depth cameras to perceive the environment, estimate terrain geometry and contact points, and compute feasible flipper angles to maintain stability during climbing. Most approaches simplify the problem by approximating the terrain and motion direction in two dimensions and synchronizing the left and right flippers to reduce the computational load.

Although efficient, this simplification reduces adaptability. Robots encounter difficulties when the terrain varies laterally, such as during diagonal approaches to obstacles or traversal of side slopes, and are limited in handling fully three-dimensional interactions, particularly avoiding side collisions during turning.

The objective of this study is to achieve independent left–right flipper control for crawler robots, enabling climbing and turning maneuvers in fully three-dimensional (3D) terrain. Unlike existing studies that rely on two-dimensional approximations and synchronized flipper control, the proposed method introduces a contact estimation strategy that explicitly accounts for flipper width, body orientation (roll and pitch), and yaw rotation. Furthermore, instead of relying on precise localization or pre-built maps, the method employs 2.5D elevation maps generated directly from raw point clouds, providing a flexible and generalizable control framework for irregular 3D terrain.

This study makes the following contributions to crawler robots.

- **Terrain-aware flipper control using only 2.5D maps:** Determines the reference body height and automatically switches between driving and climbing modes based on the terrain shape.
- **Collision avoidance and posture control based on flipper contact estimation:** Enables safe motion during both straight driving and turning by independently controlling all four flippers.

2. Related Work

2.1. Environmental Perception

Accurate perception of the surrounding terrain is essential for motion planning in mobile robots. Point clouds obtained from RGB-D cameras or LiDAR sensors are commonly used for this purpose. Direct use of raw point clouds can lead to unstable perception when data are sparse due to sensor blind spots or limited measurements. To achieve more stable posture control of crawler robots, Okada et al. proposed a method for estimating planar surfaces by fitting them to observed points [1]. Conventional approaches transform large point cloud data into simplified representations for effective management. Occupancy grid maps are commonly used for 2D environments, whereas OctoMap [2] provides a 3D voxel-based representation. GridMap [3] is widely employed for 2.5D terrain representation. Although the 2.5D GridMap cannot fully represent 3D obstacles, it enables motion planning strategies on uneven terrain based on the local gradient, supporting long-range navigation with integrated localization [4] and gait planning of legged robots [5]. In this study, we focused on the behavior of a crawler robot on various terrain shapes. Therefore, we employed a 2.5D GridMap without relying on localization, which allows flexible adaptation to irregular terrain while maintaining computational efficiency.

2.2. Contact Estimation

Stable locomotion of crawler robots requires accurate estimation of the flipper contact states and resulting body posture. Unlike legged robots with discrete foot contacts, crawler robots maintain a continuous contact area along the belts and flippers, and the contact locations vary. This characteristic makes it challenging to predict the body tilt based on local contacts alone. Oehler et al. [6] proposed a method that simultaneously estimates flipper contacts and full-body postures using kinematics and force–angle stability measurements. Their method provides accurate posture predictions but requires a complete prior terrain map and computes body posture after determining joint angles, which limits its real-time applicability. In contrast, Yuan et al. [7] first determined body posture and then computed joint angles, enabling real-time flipper control. However, this approach mainly addresses forward motion and does not fully account for turning maneuvers or lateral contact. In this study, we built upon the method of Yuan et al. to incorporate turning motions while estimating the contact states and body posture. This approach improves maneuverability on three-dimensional irregular terrain, although dynamically feasible postures are not always guaranteed. Thus, the proposed method balances computational efficiency, real-time control, and practical applicability.

2.3. Flipper Control

Flipper control directly affects both the climbing performance and postural stability of crawler robots. Existing methods have various limitations. Passive joint approaches achieve low computational costs but cannot adequately handle turning or constrained environments [8]. Posture control methods maintain a horizontal orientation but struggle with large obstacles or structural interference [9]. Planning approaches based on environmental perception are effective for step climbing, but they often rely on two-dimensional approximations and synchronization of either the left–right or front–rear flippers, which limits their adaptability to fully three-dimensional terrains [10], [11], [12], [13], [14], [15]. Geometric methods allow independent flipper control; however, challenges remain in maintaining stability during turning or complex terrain [1], [7]. Deep reinforcement learning provides autonomous adaptation; however, generalization to unknown environments and retraining costs are major concerns [16], [17]. In this study, we extended the geometric approach by determining flipper motion according to terrain features, thereby reducing the reliance on optimization-based computation and lowering the computational load. Furthermore, we introduce a contact estimation strategy that explicitly accounts for the flipper width and body orientation (roll, pitch, and yaw), enabling the integrated control of climbing and turning maneuvers on three-dimensional irregular terrain. Because the method is based on the geometric parameters of the robot, such as the link length, body width, and center of mass, it has the potential for generalization to robots of different sizes and structures, although only a single configuration is demonstrated here.

3. Problem Definition

This section describes the crawler robot model considered in this study and formulates the problem of flipper control during climbing and turning. The robot was equipped with four flipper joints and received input from point-cloud sensors and an onboard IMU. The objective is to determine the flipper joint angles that ensure stable locomotion while avoiding undesirable contact between the robot body and surrounding obstacles based on terrain information. The IMU, which was mounted on the robot body, provided estimates of the roll and pitch angles. A LiDAR sensor captures the point clouds of the surrounding environment. In this study, we assumed that sufficiently dense voxelized point clouds were available and that occlusions or missing regions caused by blind spots were not considered.

3.1. Target Robot Model

The target robot, illustrated in Figure 2, is a crawler-type platform with two pairs of flippers at the front and rear. The geometric parameters include body width w_b, body length l_b, flipper width w_f, flipper length l_f, and pulley radius r_p, Each flipper is identified as left-front (LF), left-back(LB), right-front (RF), and right-back (RB), is associated with a joint angle θ_f and a joint position $\boldsymbol{p}_f^O$, where $f \in \{LF, LB, RF, RB\}$.Accordingly, the joint angles of the four flippers are denoted as $\theta_{LF}, \theta_{LB}, \theta_{RF}, \theta_{RB}$. Three coordinate frames are defined as follows:

- **World coordinate frame** Σ_O: The global reference frame represents terrain and absolute position.
- **Robot coordinate frame** Σ_R: A frame fixed at the center of the robot, used as the reference for LiDAR point cloud and IMU posture measurements. The robot position is represented as $\boldsymbol{p}_R^O \in \mathbb{R}^3$, and orientation as $\boldsymbol{R}_R^O = R_z(\psi)R_y(\theta)R_x(\phi)$, where $\phi, \theta,$ and ψ denote the roll, pitch, and yaw angles, respectively.
- **Base coordinate frame** Σ_B: Defined at the robot position with an orientation aligned to gravity, serving as the reference for body posture control and flipper angle computation. Its position satisfies $\boldsymbol{p}_B^O = \boldsymbol{p}_R^O$, and its orientation is given by $\boldsymbol{R}_B^O = R_z(\psi)$.

3.2. Skelton Model

For the contact evaluation between the crawler robot and the terrain, we adopted the skeleton model proposed by Yuan et al. [7]. A conceptual illustration of the model is shown in Figure 3. In this framework, each link of the crawler robot is approximated by a straight-line segment, and the terrain is dilated by the pulley radius to efficiently perform collision detection. To enable a three-dimensional contact evaluation, this study employed a 2.5D map representation of the surrounding environment.

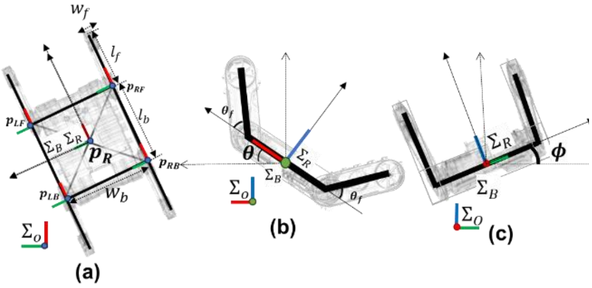

Figure 2 Coordinate system of the crawler robot: (a) top view, (b) side view from the left, and (c) front view.

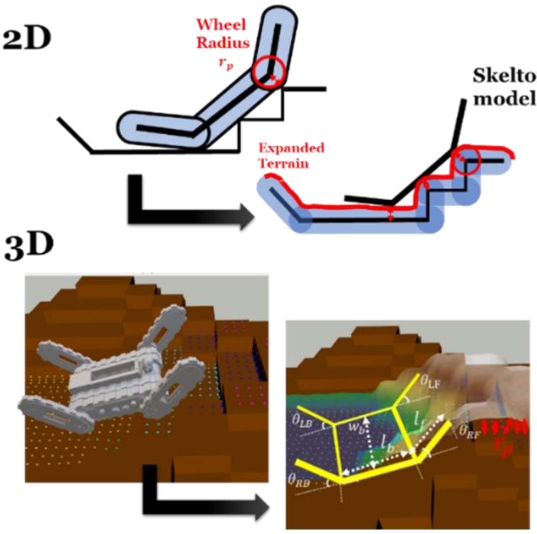

Figure 3 Skelton model for crawler type robot.

3.3. Target Terrain

The target scenarios in this study included (a) approaching a step obstacle from an oblique angle, (b) traversing a slope such as stairs. These conditions are difficult to address using conventional two-dimensional terrain approximations or synchronized left–right flipper control. Instead, independent flipper actuation and three-dimensional contact estimation are required for this purpose. Terrains with multiple height values for the same coordinates (e.g., tunnels or multilayered structures) were excluded from consideration.

4. Controller

This section describes the proposed flipper control method and the processing flow of the system. Based on the robot model and coordinate frames defined in Section 3, the method determines both the body pose and flipper joint angles according to the commanded velocity inputs.

4.1. System Overview

A overview of the proposed control system is shown in Figure 4, which consists of four main modules: (1) Terrain processing, which filters 3D LiDAR data and generates 2.5D elevation maps for each modules (2) Mode Selector, which determines the mode of body controller (3) Body Pose Control, which computes the desired body orientation based on the filtered map to maintain stability and enable terrain traversal; and (4) Flipper Control, which determines the target flipper angles to maintain ground contact. The system computes the body pose and flipper joint angles of the crawler robot according to velocity commands provided either by a human operator or a path-following algorithm. To satisfy the commanded velocity, the system first determines the desired body pose and then computes the corresponding flipper angles. Point cloud data acquired from the LiDAR sensor and orientation measurements obtained from the IMU are used to predict terrain contact and enable appropriate climbing maneuvers.

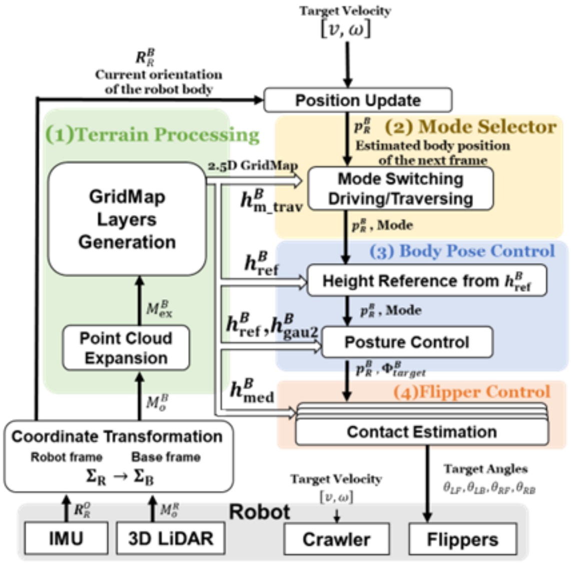

Figure 4 Overview of the proposed flipper control system. The system consists of four main modules: (1) Terrain processing, (2) Mode Selector, (3) Body Pose Control, and (4) Flipper Control.

4.2. *Terrain Processing*

4.2.1. *Point Cloud Processing*

The raw point cloud $M_o^R = \{m_i^R\}_{i=1}^n$ obtained in the robot coordinate system Σ_R is first transformed into the base coordinate system Σ_B. From this transformed cloud, only the points within a radius $d_{\max}$ on the XY-plane are retained, and voxelization is applied to homogenize the density, yielding the environment point cloud M^B. To incorporate the physical dimensions of the robot, each point in M^B is dilated to reflect the wheel radius r_p and flipper width w_f. The expanded point set M^B_{ex} is defined as:

$$M^B_{\text{ex}} = \bigcup_{i=1}^{n\prime} \left\{ m_i^B + \begin{bmatrix} r_p cos\theta \\ h \\ r_p sin\theta \end{bmatrix} \middle| \theta \in [0, 2\pi], \quad h \in \left[-\frac{w_f}{2}, \quad \frac{w_f}{2}\right] \right\}, \tag{1}$$

where r_p is the wheel radius and w_f is the flipper width. This formulation expands each point into a vertical cylindrical region, effectively modeling the swept volumes of the wheels and flippers. Such dilation enables subsequent collision detection using the skeleton model to consider the actual geometry of the robot, thereby preventing unexpected contacts. From the expanded point cloud M^B_{ex}, a 2.5D elevation map h^B_{elev} is generated with lengths l_x and l_y and a resolution of α along the Z-axis. This elevation map is then used for collision evaluation with the skeleton model.

4.2.2. *Reference Layer Generation*

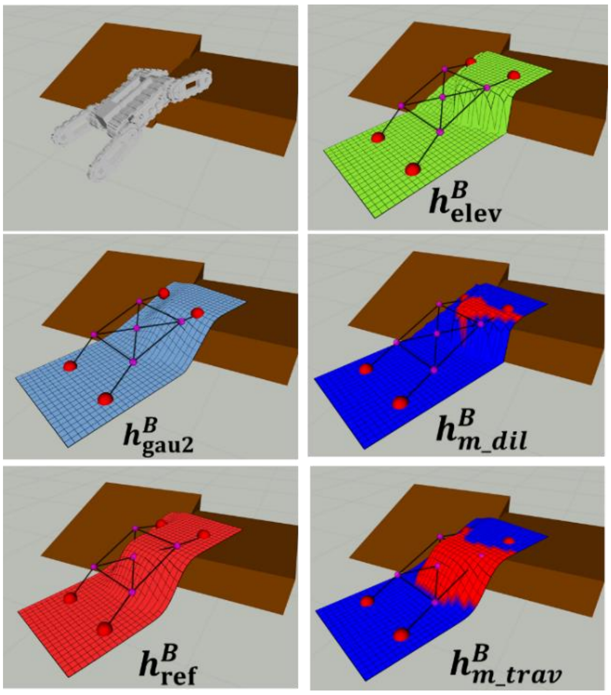

Figure 5 GridMap layers: h^B_{elev} (terrain), h^B_{gau2} (filtered terrain), $h^B_{m\_dil}$ (edges detection, blue=0, red=1), h^B_{ref} (dilated and smoothed around step), and $h^B_{\text{m\_trav}}$ (areas exceeding a height threshold above ground, blue=0, red=1).

From the elevation map h^B_{elev}, multiple reference layers are constructed as shown in Figure 5. First, a median filter is applied to obtain h^B_{med}. Then, Gaussian filters (Appendix 1) are applied to generate h^B_{gau1} and h^B_{gau2}. Let the kernel parameter for h^B_{gau2} be k_2; then the kernel for h^B_{gau1} is $2k_2 + 1$, making h^B_{gau1} smoother than h^B_{gau2}.

$$h^B_{\text{gau1}} = \text{Gauss}_{2k_2+1,\sigma_1}(h^B_{\text{elev}}), \tag{2}$$

$$h^B_{\text{gau2}} = \text{Gauss}_{k_2,\sigma_2}(h^B_{\text{elev}}). \tag{3}$$

Here, σ₁ and σ₂ represent the standard deviations of the Gaussian filters, which determine the level of smoothing applied to the elevation map. A larger σ result in stronger smoothing, as each cell value is influenced by a wider neighborhood. Two Gaussian filters were employed to distinguish whether a height discontinuity corresponds to a single step or a continuous sequence of steps, such as stairs. In the Gaussian filtering process, the kernel size defines the spatial extent of the terrain considered for smoothing, while σ controls the degree to which neighboring elevations influence the filtered value. These parameters depend on the desired sensitivity to terrain variation and the resolution of the grid map, and they were determined empirically through preliminary trials. The difference layer is defined as $h^B_{\mathrm{d\_gau}} = h^B_{\mathrm{gau2}} - h^B_{\mathrm{gau1}}$. A binary mask layer is generated according to a threshold $d_{\mathrm{th\_ref}}$:

$$h^B_{\mathrm{m\_dil}} = \begin{cases} 1, & h^B_{\mathrm{d\_gau}} > d_{\mathrm{th\_ref}} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Maximum filter (Appendix 2) is applied to the masked map, resulting in

$$\tilde{h}^B = \mathrm{Dilate}_{\mathrm{k_{dx}},\mathrm{k_{dy}}}\left(h^B_{\mathrm{m\_dil}}\ h^B_{\mathrm{med}}\right), \tag{5}$$

where $\mathrm{k_{dx}}, \mathrm{k_{dy}}$ are dilation kernels, computed from the kinematic parameters:

$$\mathrm{k_{dx}} = \frac{\left(\frac{l_b}{2} + l_f\right)}{\alpha}, \qquad \mathrm{k_{dy}} = \frac{\left(\frac{w_b}{2} + w_f\right)}{\alpha}. \tag{6}$$

The dilated elevation layer is given by

$$h^B_{\mathrm{dil}} = \max\left(\tilde{h}^B, h^B_{\mathrm{med}}\right). \tag{7}$$

Finally, a Gaussian filter is applied to obtain the reference layer:

$$h^B_{\mathrm{ref}} = \mathrm{Gauss}_{k_{\mathrm{refx}},k_{\mathrm{refy}},\sigma_{\mathrm{ref}}}\left(h^B_{\mathrm{dil}}\right). \tag{8}$$

To switch between driving and climbing strategies, the difference

$$h^B_{\mathrm{d\_ref}} = \left|h^B_{\mathrm{ref}} - h^B_{\mathrm{med}}\right| \tag{9}$$

is used to construct a traversal mask:

$$h^B_{\mathrm{m\_trav}} = \begin{cases} 1, & h^B_{\mathrm{d\_ref}} > d_{\mathrm{th\_trav}} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

4.3. *Body Pose Control*

4.3.1. *Reference Point Computation*

The commanded velocity input in the robot coordinate frame is defined as: $\boldsymbol{V}^R = [\boldsymbol{v}^R\ \boldsymbol{\omega}_R]^\top$ $\boldsymbol{v}^R = [v_x\ 0\ 0]^\top$, $\boldsymbol{\omega}^R = [0\ 0\ \omega_z]^\top$. The predicted position of the robot center after Δt seconds is:

$$\Delta\boldsymbol{p}^R_R = \frac{v^R_x}{\omega^R_z}\begin{bmatrix} \sin(\omega_z \Delta t) \\ 1 - \cos(\omega_z \Delta t) \\ 0 \end{bmatrix}. \tag{11}$$

Transforming this displacement into the base frame using the current orientation $R^B_R = R_y(\theta_0)\, R_x(\phi_0)$, we obtain

$$\boldsymbol{p}^B_R(\Delta t) = p^B_R(0) + R^B_R \Delta\boldsymbol{p}^B_R. \tag{12}$$

Additional reference points for front, back, left, and right edges of the body are defined as:

$$\begin{aligned} \boldsymbol{p}^B_{\mathrm{front}} &= \boldsymbol{p}^B_R + \left[\frac{l_b}{2}\ 0\ 0\right]^\top, \\ \boldsymbol{p}^B_{\mathrm{back}} &= \boldsymbol{p}^B_R + \left[-\frac{l_b}{2}\ 0\ 0\right]^\top, \\ \boldsymbol{p}^B_{\mathrm{left}} &= \boldsymbol{p}^B_R + \left[0\ \frac{w_b}{2}\ 0\right]^\top, \\ \boldsymbol{p}^B_{\mathrm{right}} &= \boldsymbol{p}^B_R + \left[0\ -\frac{w_b}{2}\ 0\right]^\top. \end{aligned} \tag{13}$$

The elevation at these reference points is obtained from the terrain layers h^B_{ref}, giving the predicted robot center after Δt:

$$\boldsymbol{p}^B_{\mathrm{R}}(\Delta t) = \begin{bmatrix} x^B_R(\Delta t) \\ y^B_R(\Delta t) \\ h^B_{\mathrm{ref}}\left(x^B_R(\Delta t), y^B_R(\Delta t)\right) \end{bmatrix} \tag{14}$$

4.3.2. *Target Pose Computation*

The target body orientation is determined by switching between **Traversing Mode** and **Driving Mode** according to the traversal mask:

$$\mathbf{Mode} = \begin{cases} \text{Traversing,} & h^B_{\mathrm{m_{trav}}}\left(x^B_R(\Delta t), y^B_R(\Delta t)\right) = 1 \\ \text{Driving,} & h^B_{\mathrm{m_{trav}}}\left(x^B_R(\Delta t), y^B_R(\Delta t)\right) = 0 \end{cases} \tag{15}$$

4.3.2.1. *Traversing Mode*

The pitch angle is computed from the elevation difference between front and back points:

$$\theta' = \begin{cases} -\tan^{-1}\left(\dfrac{h^B_{\mathrm{ref}}\left(x^B_{\mathrm{front}}, y^B_{\mathrm{front}}\right)}{\frac{l_b}{2}}\right), & h^B_{\mathrm{ref}}(x^B_{\mathrm{front}}, y^B_{\mathrm{front}}) \geq h^B_{\mathrm{ref}}(x^B_{\mathrm{back}}, y^B_{\mathrm{back}}) \\ \tan^{-1}\left(\dfrac{h^B_{\mathrm{ref}}(x^B_{\mathrm{back}}, y^B_{\mathrm{back}})}{\frac{l_b}{2}}\right), & \text{otherwise.} \end{cases} \tag{16}$$

and roll angle is computed $\phi' = 0.5 \times \phi$. To avoid excessive pitch angle increase during climbing, an additional offset is introduced to raise the body center height:

$$z_R^B(\Delta t) += \frac{k_{z\theta} l_b}{2}|\sin\theta'| + \frac{k_{z\phi} w_b}{2}|\sin\phi'|. \tag{17}$$

Here, $z_R^B(\Delta t)$ is the target body height, l_b is the body length, $k_{z\theta}$, $k_{z\phi}$ is a correction coefficient for θ', ϕ', respectively.

4.3.2.2. Driving Mode:

The gradient at the reference layer yields θ_{ref} and ϕ_{ref}. To prevent abrupt changes, clearance-based estimates are introduced:

$$\theta_{\text{clearance}} = -\tan^{-1}\frac{h^B_{\text{gau2}}(x^B_{\text{front}}, y^B_{\text{front}}) - h^B_{\text{gau2}}(x^B_{\text{back}}, y^B_{\text{back}})}{l_b} \tag{18}$$

$$\phi_{\text{clearance}} = \tan^{-1}\frac{h^B_{\text{gau2}}(x^B_{\text{left}}, y^B_{\text{left}}) - h^B_{\text{gau2}}(x^B_{\text{right}}, y^B_{\text{right}})}{w_b} \tag{19}$$

$$\theta' = \begin{cases} \theta_{\text{clearance}}, & |\theta_{\text{clearance}}| < |\theta_{\text{ref}}| \\ \theta_{\text{ref}}, & \text{otherwise}, \end{cases} \tag{20}$$

$$\phi' = \begin{cases} \phi_{\text{clearance}}, & |\phi_{\text{clearance}}| < |\phi_{\text{ref}}| \\ \phi_{\text{ref}}, & \text{otherwise}. \end{cases} \tag{21}$$

The final target angles are smoothed by weighted averaging:

$$\theta_{\text{target}} = \alpha_\theta \theta' + (1 - \alpha_\theta)\theta_{\text{prev}}, \tag{22}$$

$$\phi_{\text{target}} = \alpha_\phi \phi' + (1 - \alpha_\phi)\phi_{\text{prev}}, \tag{23}$$

where $\alpha_\theta, \alpha_\phi \in [0,1]$ are damping coefficients, and θ_{prev} and ϕ_{prev} are target value of previous time step.

4.4. *Autonomous Flipper Control Method*

Based on the 2.5D elevation map h^B_{med} and the estimated body pose in the base frame, target flipper angles are determined. The objective is to satisfy the target body pose while ensuring stable contact with the terrain and preventing unexpected collisions. Algorithm 1 outlines the computation procedure. For each flipper, the joint position is calculated, and the contact angle is estimated by searching for the first intersection of the flipper link (length l_f) with the terrain. The flipper joint positions in the robot frame are determined as:

$$p_f^R = \begin{bmatrix} (-1)^{a_f}\frac{l_b}{2} \\ (-1)^{b_f}\frac{w_b}{2} \\ 0 \end{bmatrix}, \qquad f \in \{LF, LB, RF, RB\}, \tag{24}$$

where $(a_f, b_f) \in \{(0,0), (1,0), (0,1), (1,1)\}$ correspond to LF, LB, RF, and RB, respectively. For contact evaluation, a search point on the flipper link is defined as:

$$s_f^B = p_f^B + \begin{bmatrix} \sigma_f l' cos\theta_f \\ 0 \\ l' sin\theta_f \end{bmatrix}, \qquad 0 \le l' \le l_f, \tag{25}$$

where σ_f denotes the sign distinguishing the front (+) and rear (-) flippers.

Algorithm 1 Flipper Ground Contact Angle and Length Optimization (with σ_f)

```
Require: Reference pose $T_{R\to B}$, 2.5D terrain map $h^B_{\text{med}}$, Flipper parameters $\{l_f, p_f^B, \text{is\_front}\}$, Wheel radius $r_p$
Ensure: Optimized flipper angles $\theta_f$ and contact lengths $l'_f$
for each flipper $f \in \{LF, RF, LB, RB\}$ do
    Initialize maximum contact angle: $\theta_{\max} = -70 - \text{pitch}$
    Initial angle: $\theta_{\text{initial}} = \min(70 - \text{pitch}, 70)$
    Set $l'_f = l_f$, contact = false
    for $l' = l_f$ down to $r_p$ (step $-0.02$) do
        for $\theta_f = \theta_{\text{initial}}$ down to $\theta_{\max}$ (step 5° in rad) do
            Compute flipper tip (exploration) point in base frame:
                $s_f^B = p_f^B + \begin{bmatrix} \sigma_f l' \cos\theta_f \\ 0 \\ l' \sin\theta_f \end{bmatrix}$
            Transform tip to base coordinates: $P_B = T_{R\to B} \cdot s_f^B$
            if $P_B.z \le h^B_{\text{med}}(P_B.x, P_B.y)$ then
                Update $\theta_{\max} = \max(\theta_{\max}, \theta_f)$
                Update $l'_f = l'$, contact = true
            end if
        end for
    end for
    Set optimized flipper angle and length: $\theta_f = \theta_{\max}$, $l'_f = l'_f$
end for
```

5. Experiments and Results

5.1. *Method*

Experiments were conducted in the Gazebo simulation environment. The controller was implemented in C++ on ROS Noetic. Each joint operated within a range of –70° to +70° under 100 Hz position control, regulated by an internal PD controller. The belt geometry was simulated using the plugin developed by Okada et al. [18]. All simulation, control, and data recording processes were performed on a single desktop PC. The system operated at 20 Hz for GridMap generation, 20 Hz for flipper angle computation, and 100 Hz for joint control. The simulator ran in real time at 100 Hz, while robot pose, joint states, and mode states were recorded at 20 Hz. The parameters used in the experiment are listed in Table 1. We conducted two experiments, **Step and stair climbing tasks** and **Oblique step traversal tasks,** as shown in Figure 6.

Table 1. Parameters used in the experiments

	Parameters	
Kinematics	$l_b = 0.45, w_b = 0.45,\ l_f = 0.29,$ $w_f = 9.1\ \ r_p = 0.082$	
GridMap	Common	$l_x = 0.8, l_y = 0.4,$ $\alpha = 0.04$
	Gauss	$k_2 = 7, \sigma_1 = \sigma_2 = 5.0$
	Mask	$d_{\mathrm{th\_ref}} =\ 0.03$
	Dilation	$k_{\mathrm{dx}} = 13, k_{\mathrm{dy}} = 9$
	Reference	$k_{\mathrm{refx}} = 25,$ $k_{\mathrm{refy}} = 25, \sigma_{\mathrm{ref}} = 5.0$
	Traverse	$d_{\mathrm{th\_trav}}\ = 0.05$
Controller	$k_{z\theta} = 0.25,\ k_{z\phi} = 1.0,\ \Delta t = 0.05,$ $\alpha_\theta = 0.1, \alpha_\phi = 0.1$	

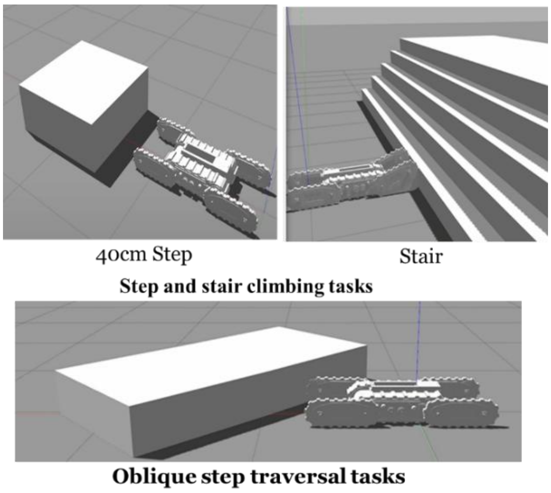

Figure 6 Gazebo Simulation Environment

5.1.1. *Step and stair climbing tasks*

These experiments aimed to evaluate the robot's capability to transition between traverse and drive modes during straight-ahead climbing. When negotiating discrete steps, the rear flippers were employed to lift the body and mitigate excessive pitch angles. In contrast, during stair climbing with continuous elevation changes, the robot was required to conform its posture to the terrain profile. The target linear velocity was fixed at 0.2 m/s. Step heights were set to 20 cm, 30 cm, and 40 cm. The stair configuration consisted of steps with a uniform riser height of 20 cm and an overall inclination angle of 45°.

5.1.2. *Oblique step traversal tasks*

These experiments assessed the robot's mobility when approaching steps obliquely, thereby verifying its adaptability in three-dimensional unstructured environments. In this scenario, roll-angle correction using the left and right flippers was critical. The robot advanced toward steps of 30 cm in height from a 45° approach angle at a constant velocity of 0.2 m/s.

5.2. *Results*

Figure 7, Figure 8, Figure 9, and Figure 10 show the crawler robot's motion, the z-coordinate in the world frame centered on the robot in simulation, and the pitch angle in the robot frame during the step and stair climbing tasks. Figure 11 additionally presents the joint angle trajectories of the four flippers during the oblique step traversal tasks. The colors in the graphs indicate the driving mode and traversing mode.

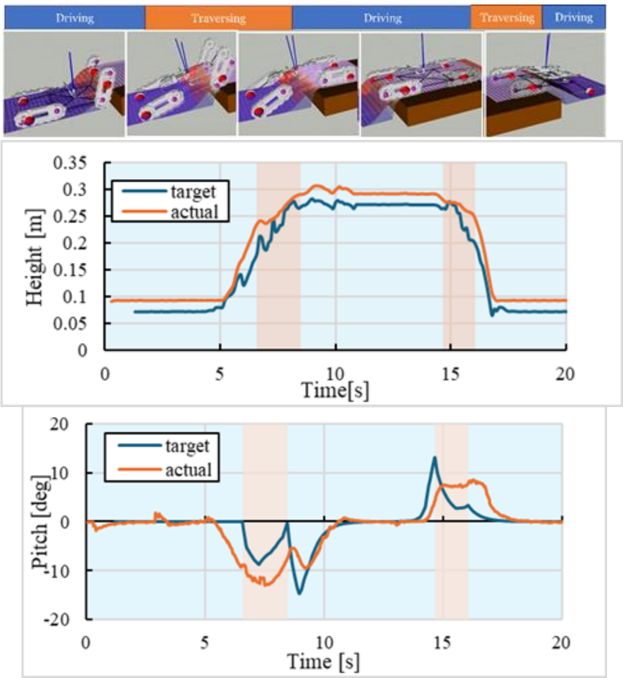

Figure 7 Robot motion over a 20 cm step, including time-series data of the z-coordinate and pitch angle.

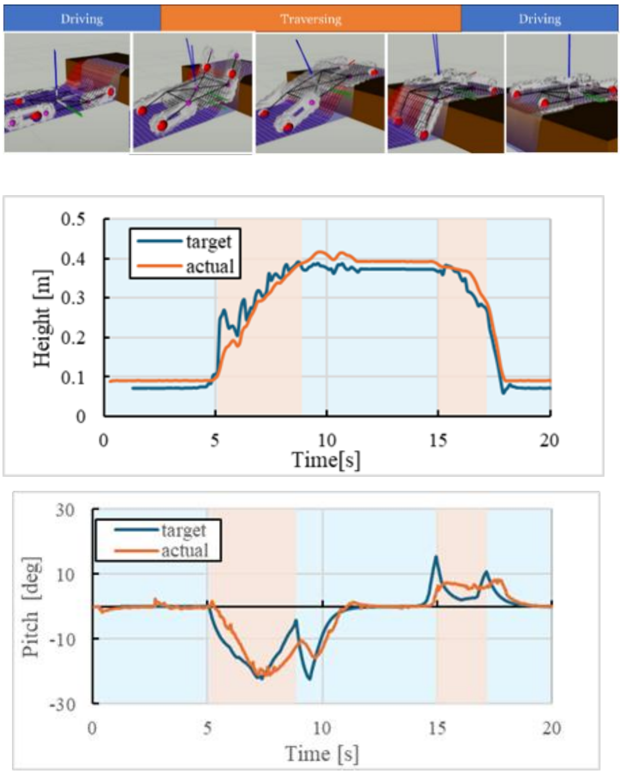

Figure 8 Robot motion over a 30 cm step, including time-series data of the z-coordinate and pitch angle

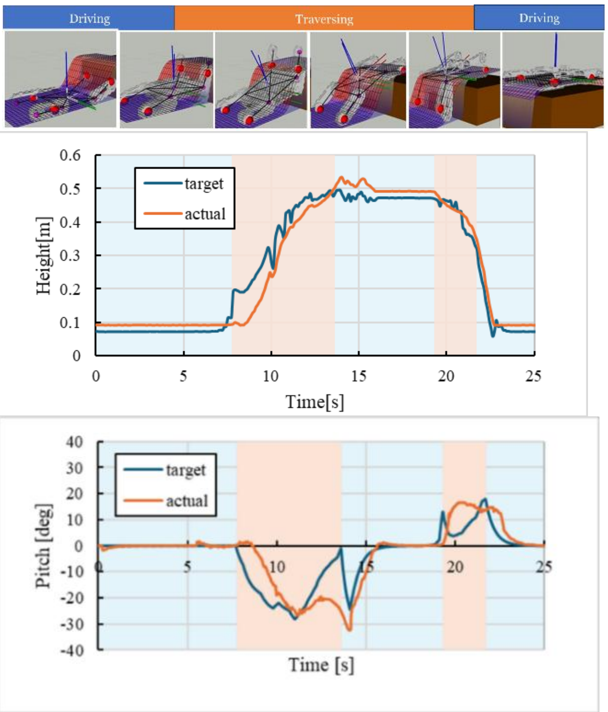

Figure 9 Robot motion over a 40 cm step, including time-series data of the z-coordinate and pitch angle.

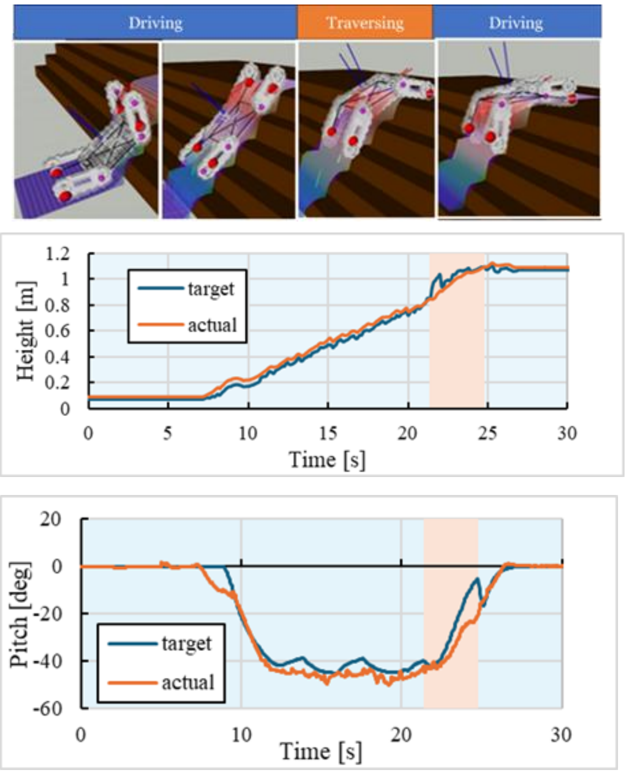

Figure 10 Robot motion over stairs. Each step is 20 cm high with an incline angle of 45°, showing time-series data of the z-coordinate and pitch angle.

6. Discussion

6.1. Mode Switching

For step heights of 20 cm, 30 cm, and 40 cm, the robot consistently switched to traversing mode during both ascent and descent. These results indicate that the traversing-mode controller effectively suppressed variations in body height and pitch angle during climbing and descending. Comparison of stair climbing with the 20 cm step traversal (Figure 7), revealed that mode switching depends not only on step height but also on the terrain ahead of the robot. As shown in Figure 10, the robot initially engaged in climbing, switched to driving mode at the midsection, and finally returned to traversing mode near the upper steps to complete the ascent. During stair traversal, the crawler robot aligned its body orientation with the slope to secure sufficient traction. The body pitch angle in driving mode approximately matched the stair inclination of 45°, indicating successful alignment. Traversing mode further adjusted the posture to reduce the pitch angle toward 0°. These results demonstrate that the proposed controller effectively manages both step climbing and slope traversal.

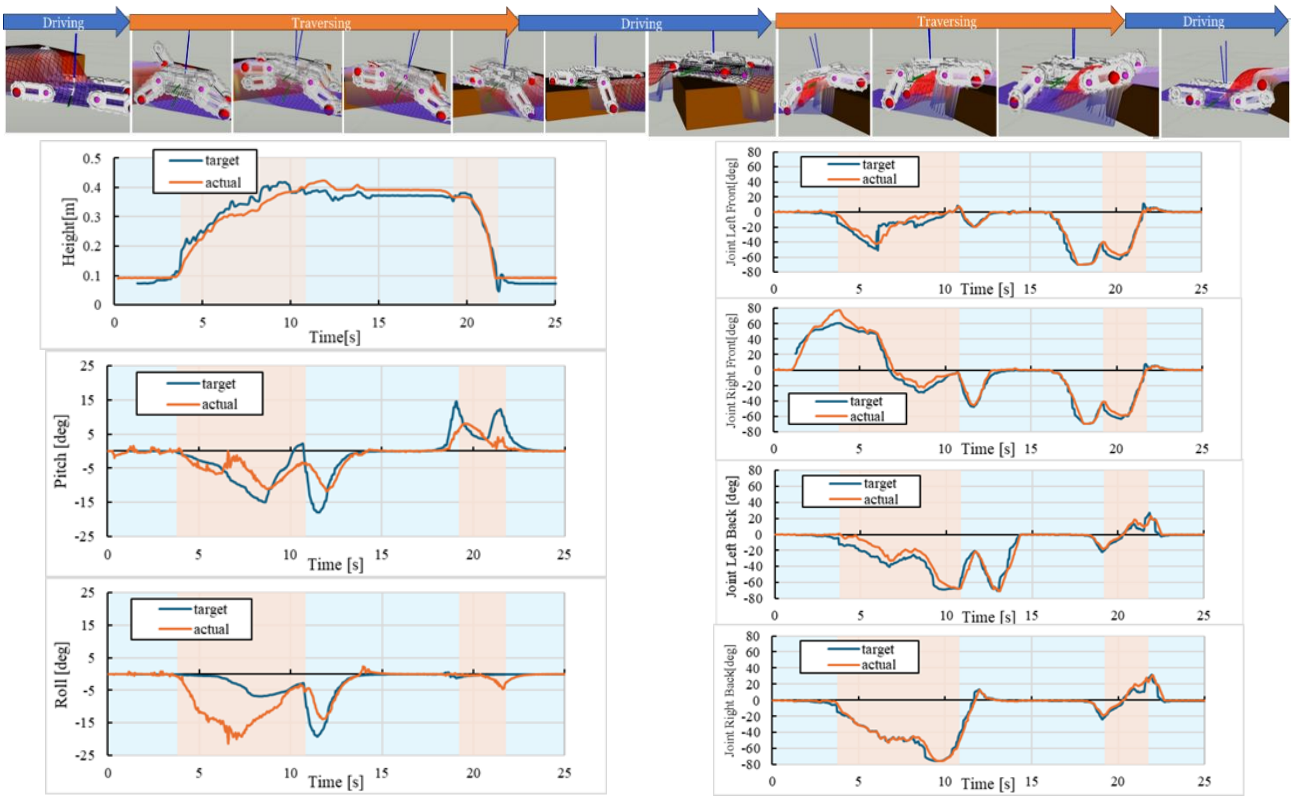

Figure 11 Robot motion during oblique step traversal. The step is 30 cm high with an approach angle of 45°, showing time-series data of the z-coordinate, pitch angle, roll angle, and joint angles.

6.2. Flipper Control

The proposed flipper control, based on ground contact prediction, enabled independent control of all four flippers. During step climbing, raising the flippers prior to contact with the step successfully prevented the robot from becoming stuck. In oblique step climbing (Figure 11), the right-front flipper was raised first to avoid collision while simultaneously contributing to posture regulation in terms of body height, roll angle, and pitch angle. As the robot advanced, the left-front flipper was also raised to prevent interference with the step. This behavior demonstrates that consideration of the flipper width effectively prevented entrapment. During descent, lowering the front flippers beneath the step suppressed fluctuations in the body angle. These results indicate that the proposed method ensures both collision avoidance and postural stability.

7. Conclusion

This study proposes a flipper control method that does not require sequential optimization and instead relies on terrain geometry. By filtering a 2.5D grid map of the terrain, the method extends the conventional step-climbing sequence of crawler robots to three-dimensional unstructured environments, enabling the generation of target body postures. Flipper angle computation is based on contact prediction with the environment, accounting not only for frontal interactions but also for contacts during turning maneuvers and lateral approaches. This allows the robot to maintain stable body postures while adapting to irregular terrain. As a result, the proposed method enables autonomous flipper control without manual intervention and contributes to expanding the operational range of crawler robots in climbing tasks.

Future work will focus on addressing sensing challenges to construct accurate 2.5D maps of the environment surrounding the robot.

Appendix

1. Gaussian Filter

$$\mathrm{Gauss}_{k_x,k_y,\sigma}\big(h(i,j)\big) = \sum_{m=-\frac{k_x-1}{2}}^{\frac{k_x-1}{2}} \sum_{n=-\frac{k_y-1}{2}}^{\frac{k_y-1}{2}} \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{m^2+n^2}{2\sigma^2}} h(i+m,j+n) \qquad (26)$$

2. Maximum Filter

$$\mathrm{Dilate}_{k_x,k_y}\big(h(i,j)\big) = \max_{m=-\frac{k_x-1}{2}}^{\frac{k_x-1}{2}} \max_{n=-\frac{k_y-1}{2}}^{\frac{k_y-1}{2}} h(i+m,j+n) \qquad (27)$$

Acknowledgements

This work was supported by JST SPRING, Grant Number JPMJSP2112.

References

1. Y. Okada, K. Nagatani, K. Yoshida, S. Tadokoro, T. Yoshida, and E. Koyanagi, Shared Autonomy System for Tracked Vehicles on Rough Terrain Based on Continuous Three-Dimensional Terrain Scanning," Journal of Field Robotics, vol. 28, no. 6, 2011, pp. 875–893.
2. A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, Autonomous Robots, vol. 34, no. 3, 2013, pp. 189–206.
3. P. Fankhauser and M. Hutter, A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation, in Robot Operating System (ROS)-The Complete Reference (Vol. 1), A. Koubaa (ed.), Springer, Cham, 2016.
4. T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, Elevation Mapping for Locomotion and Navigation Using GPU, in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2022, pp. 2273–2280.
5. F. Jenelten, R. Grandia, F. Farshidian, and M. Hutter, Tamols: Terrain-Aware Motion Optimization for Legged Systems, IEEE Transactions on Robotics, vol. 38, no. 6, 2022, pp. 3395–3413.
6. M. Oehler and O. von Stryk, Accurate Pose Prediction on Signed Distance Fields for Mobile Ground Robots in Rough Terrain, in Proc. of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), IEEE, 2023, pp. 47–52.
7. Y. Yuan, Q. Xu, and S. Schwertfeger, Configuration-Space Flipper Planning on 3D Terrain, in Proc. Of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2020, pp. 318–325.
8. A. Watanabe, T. Mitsuhashi, M. Okugawa, K. Ogane, T. Kimura, T. Kinugasa, and Y. Ohtsubo, Ground Adaptability of Crawler Mobile Robots with Sub-Crawler Rotary Joint Compliance, Journal of Robotics and Mechatronics, vol. 36, no. 3, 2024, pp. 732-745.
9. F. Rocha, A. Cid, M. Delunardo, R. P. Junior, N. S. Thiago, L. Barros, J. D. Domingues, G. Pessin, G. Freitas, and R. Costa, Body Posture Controller for Actively Articulated Tracked Vehicles Moving over Rough and Unknown Terrains, in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2023, pp. 2330–2337.
10. M. Gianni, F. Ferri, M. Menna, and F. Pirri, Adaptive Robust Three-Dimensional Trajectory Tracking for Actively Articulated Tracked Vehicles, Journal of Field Robotics, vol. 33, no. 7, 2016, pp. 901– 930.
11. M. Norouzi, J. V. Miro, and G. Dissanayake, Probabilistic Stable Motion Planning with Stability Uncertainty for Articulated Vehicles on Challenging Terrains, Autonomous Robots, vol. 40, no. 2, 2016, pp. 361–381.
12. M. Norouzi, J. V. Miro, and G. Dissanayake, Planning Stable and Efficient Paths for Reconfigurable Robots on Uneven Terrain," Journal of Intelligent & Robotic Systems, vol. 87, 2017, pp. 291–312.
13. M. Oehler, S. Kohlbrecher, and O. von Stryk, Optimization-Based Planning for Autonomous Traversal of Obstacles with Mobile Ground Robots," International Journal of Mechanics and Control, vol. 1, 2020, pp. 33–40.
14. B. Chen, K. Huang, H. Pan, H. Ren, X. Chen, J. Xiao, W. Wu, and H. Lu, Geometry-Based Flipper Motion Planning for Articulated Tracked Robots Traversing Rough Terrain in Real Time, Journal of Field Robotics, vol. 40, no. 8, 2023, pp. 2010–2029.
15. Z. Xu, Y. Chen, Z. Jian, J. Tan, X. Wang, and B. L. Liang, Hybrid Trajectory Optimization for Autonomous Terrain Traversal of Articulated Tracked Robots, IEEE Robotics and Automation Letters, vol. 9, no. 1, 2023, pp. 755–762.
16. H. Pan, X. Chen, J. Ren, B. Chen, K. Huang, H. Zhang, and H. Lu, Deep Reinforcement Learning for Flipper Control of Tracked Robots in Urban Rescuing Environments, Remote Sensing, vol. 15, no. 18, 2023, p. 4616.
17. Z. Gao, F. Shen, and J. Zhao, Control of Double Swing Arm Tracked Robot Based on Deep Reinforcement Learning in Various Uneven Terrains, Neural Processing Letters, vol. 57, no. 3, 2025, p. 42.
18. Y. Okada, S. Kojima, K. Ohno, and S. Tadokoro, Real-Time Simulation of Non-Deformable Continuous Tracks with Explicit Consideration of Friction and Grouser Geometry, in Proc. of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Paris, France, 2020, pp. 948-954.

Authors Introduction

Mr. Kotaro Kanazawa

He received his Bachelor's and Master's degree in Engineering,2022 and 2024, respectively, from the Department of Engineering, Nagoya Institute of technology in Japan. He is currently a Ph.D. student in Nagoya Institute of Technology, Japan.

Dr. Noritaka Sato

He is an Associate Professor in the Electrical and Mechanical Engineering Program, Department of Engineering, at Nagoya Institute of Technology. He received his PhD in Engineering from the University of Electro-Communications in 2009. His current research interests include rescue robots and human-robot interaction.

Dr. Yoshifumi Morita

He is a professor in the Electrical and Mechanical Engineering Program at the Department of Engineering, Nagoya Institute of Technology. He received his Ph.D. from the Nagoya Institute of Technology in 1998. His current research interests include robots and devices for rehabilitation assistance, and robot teaching.