

## Article

# A Deformable Configuration Planning Framework for a Parallel Wheel-Legged Robot Equipped with Lidar

Fei Guo <sup>1,2,3</sup>, Shoukun Wang <sup>1,2,3\*</sup>, Binkai Yue <sup>1,2,3</sup> and Junzheng Wang <sup>1,2,3</sup>

<sup>1</sup> School of Automation, Beijing Institute of Technology, No. 5 South Zhongguancun Street, Haidian District, Beijing 100081, China; guofei@bit.edu.cn (F.G.); binkay@126.com (B.Y.); wangjz@bit.edu.cn (J.W.)

<sup>2</sup> State Key Laboratory of Intelligent Control and Decision of Complex System at School of Automation, Beijing Institute of Technology, Beijing 100081, China

<sup>3</sup> Key Laboratory Ministry of Industry and Information Technology, Beijing Institute of Technology, No. 5 South Zhongguancun Street, Haidian District, Beijing 100081, China

\* Correspondence: bitwsk@bit.edu.cn

Received: 31 August 2020; Accepted: 27 September 2020; Published: 1 October 2020



**Abstract:** The wheel-legged hybrid robot (WLHR) is capable of adapting height and wheelbase configuration to traverse obstacles or rolling in confined space. Compared with legged and wheeled machines, it can be applied for more challenging mobile robotic exercises using the enhanced environment adapting performance. To make full use of the deformability and traversability of WLHR with parallel Stewart mechanism, this paper presents an optimization-driven planning framework for WLHR with parallel Stewart mechanism by abstracting the robot as a deformable bounding box. It will improve the obstacle negotiation ability of the high degree-of-freedom robot, resulting in a shorter path through adjusting wheelbase of support polygon or trunk height instead of using a fixed configuration for wheeled robots. In the planning framework, we firstly proposed a pre-calculated signed distance field (SDF) mapping method based on point cloud data collected from a lidar sensor and a KD-tree-based point cloud fusion approach. Then, a covariant gradient optimization method is presented, which generates smooth, deformable-configuration, as well as collision-free trajectories in confined narrow spaces. Finally, with the user-defined driving velocity and position as motion inputs, obstacle-avoidance actions including expanding or shrinking foothold polygon and lifting trunk were effectively testified in realistic conditions, demonstrating the practicability of our methodology. We analyzed the success rate of proposed framework in four different terrain scenarios through deforming configuration rather than bypassing obstacles.

**Keywords:** wheel-legged hybrid robot; trajectory optimization; motion planning; obstacle negotiation; parallel mechanism

## 1. Introduction

The legged machine shows its excellent adaptability of negotiating the convex obstacles. However, these actions necessitate substantial planning calculation [1], complicated model to sustain walking stability while swinging foot [2,3], and obstacle with a flat surface and relatively low height to maintain foothold safety [4]. As legged robots have been developing to become popular, there is growing interest in incorporating benefits from the wheeled and legged mechanism to realize hybrid locomotion, intensifying the intention of boosting mobility and efficiency. Several wheel-legged hybrid robots (WLHRs) have shown their ability of adapting to convex environment. The robot CENTAURO [5] which are equipped with steerable wheels on legs' ending allows for executing steps, omnidirectional steering, and dominating a large variety of mobile manipulation missions, demonstrating its various moving modes in different types of terrains.

Within the correlative application territory, obstacle negotiation has extensively involved several aspects among mobile robots, under the priority optimizing the robot stability and trajectory smoothness. With sustaining walking stability and kinematic feasibility, Yue [6] adopted the concept of artificial potential field on a hexapod robot to avoid obstacles. A novel obstacle avoidance method [7], Follow the Gap Method, eliminated local minimum problem in path planning on an autonomous ground vehicle with Ackermann steering geometry. The collision-free paths [8] were planned using probabilistic roadmaps targeting on a given location. A laser-based people tracking component estimated the motions of humans. Guaranteeing smoothness and collision-free, a reciprocal orientation algorithm [9] planned trajectories for multi-robot without direct communication with other robots. With the assumption of flat terrain, an optimal global path was planned [10]. Connecting states through local trajectory generator is set up as a graph-search problem, which is solved by best-first search. Furthermore, a precise tracking controller is also important for underactuated systems especially with unknown parameters and disturbances. Adaptive neural networks [11] approximated unknown dynamics and updated parameters to decrease tracking error.

To autonomously drive in different environment, specific terrain description can effectively enhance robot shifting availability. The hierarchical data structure [12] identified and stored obstacle cells as non-uniform  $2^m$  trees, which is crucial before planning. The surface normals [13] on robot-centric elevation map are employed to select feasible footholds, organizing collision-free trajectories for swing feet. The planar laser scan, stereo vision, and preoperative sensing mounted on BigDog robot transformed data collection into a 2D cost representation. BigDog [14] equipped with a combination of planar laser scans, stereo vision, and preoperative sensing to perceive obstacles and placed their data collection into a 2D cost representation. Then it planned paths through a variation classic A\* search and steered four legs to follow them. The Messor II hexapod [15] employed RGB-D data to combine OctoMap and elevation grid into a 3-D semantic labeling of natural environments, conducting its motion planning. Establishing a bridge between discrete and continuous planning primitively [16] leveraged each individual configuration of ground contact points (footholds) in legged locomotion which are explored to forge specified motion sequences to avoid collision without compromising COM agility. However, none of these approaches concentrated on obstacle avoidance for robot body and reconfigured relationship between limbs, which is independent in narrow space.

As for the planning algorithm for configurable robots or executing obstacle negotiation through changing arrangement between different robot components, have been implemented in several kinds of robots. Combining with point clouds segmentation and traversability analysis, an autonomous 3D path planning [17] executed rolling motion using a tracked reconfigurable vehicles on stairway and slope. Maximising the height of sensor payload, stability [18] and moving capability [19] originated from A\* search to enhance on uneven terrain. Anh [20] advanced an efficient coverage path planning in a novel self-reconfigurable cleaning robot. A modified A-Star zigzag global planner can generate waypoints for the purpose of maximizing robot's coverage area. A reconfigurable snake-like robot [21] integrated path and motion planning approach to move in challenging environments populated with obstacles like stairs. The terrain information used for motion planning is presented by a multi-resolution map. Then an extended RRT\* is utilized to form a multi-layered planning model, generating a multi-stepped planning process. In Reference [22], a contact dynamic roadmaps is constructed, which generates discretized motions in each leg's workspaces offline. Then a mapping method is designed to obtain motions in its configuration space. In the online phase, the planner can adapt to environment rapidly and generate collision-free foothold positions. In Reference [23], to acquire multifaceted navigation skill such as body slimming or lifting for obstacle avoidance, an end-to-end planning method based on deep reinforcement learning is presented, which generates motor commands directly from the height map. Furthermore, a multi-modal PRM is proposed in Reference [24] for planning problems with finite number of intersecting manifolds. It applied to determine support polygons and configurations of ATHLETE and HRP2 robot walking on flat, stairs, and undulating terrains. The convergence of an incremental variant of PRM was also proved. A wheel-legged hybrid locomotion was also

executed on the ATHLETE robot simulation, but the calculation quantity of mentioned planner is considerably abundant when obstacle avoidance did not involve. The pattern of wheel-legged hybrid locomotion involves not only the driving-stepping but also the foothold polygon adaptive locomotion. MOMARO robot parameterized the stepping and driving locomotion into a single ARA\* planner [25] and combined three-level abstract traversing manoeuvres [26] on stairs. Furthermore, the Value Iteration Networks (VINs) [27] planned omnidirectional driving for support polygon obstacle avoidance in cluttered terrain that combined multiple levels of abstraction. The computational cost of these planner are expensive or highly depended on number of robot joints or offline training.

Intuitively, the obstacle negotiation issue for robot in confined space can be simplified as a deformable bounding box abstraction of robot model, which is inspired from reconfigurable and soft robots. The Randomized Possibility Graphs [28] combined the bounding boxes attached to biped robot to rapidly explore the possible actions in a variety of semi-unstructured environments. The nested robot [29] was utilized to gradually induce number of abstract models, which is called the Quotient-space roadMap Planner to search action graph until a valid path had been found. A hexapod robot Weaver [30] has the ability to adapt walking posture to tunnel environment, in which the scheduled method smoothed trajectories by Covariant Hamiltonian Optimization for Motion Planning (CHOMP) [31] for Efficient Motion Planning. The CENTAURO robot [32] transformed the Octomap into 2D occupation map to determine robot polygon shrinking and expansion to go over obstacles through traditional A\* search program.

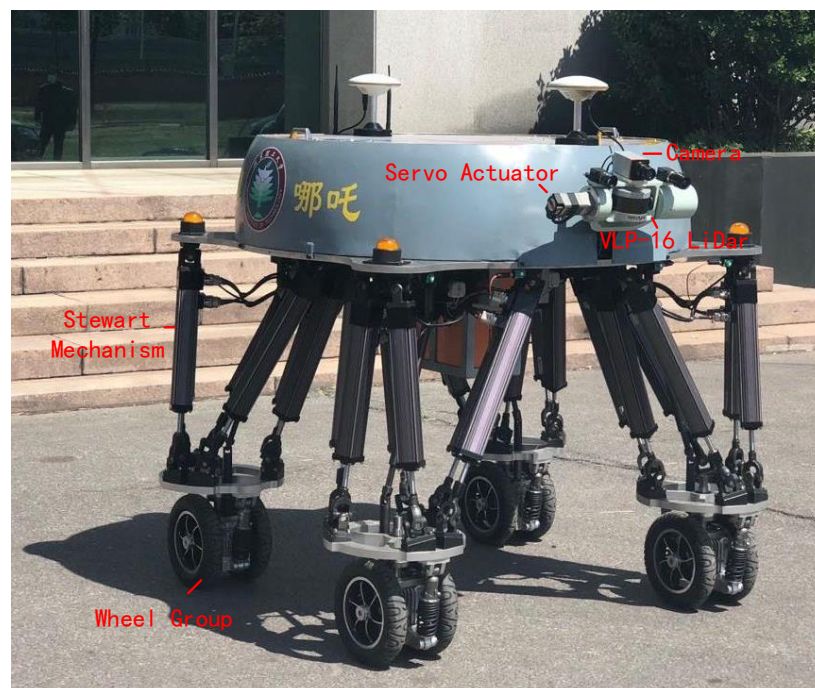
WLHR inherits the compounded superiority, including flexible obstacle-crossing function, efficiency, and agility from legged and wheeled systems. Compared to common serial leg mechanism, Stewart parallel counterpart holds higher load capacity and permits six-dimensional movability in the narrower zone, which generally creates omnidirectional movement and adjustable volume deformation on a rigid and mix-and-match manner of wheels and feet robot. There have been some researches concerning grid-searched algorithm for stepping-driving hybrid locomotion [25,26] and changeable foothold polygon [32] on two-dimensional planning to traverse obstacles, but no optimization-based composition for WLHR in three-dimensional obstacle avoidance. The BIT-NAZA robot is able to roll towards arbitrary orientation without rotating body configuration as well as adaptable trunk height and foothold polygon, which is immensely suitable constituting a rigid deformable robot. The covariant gradient optimization devotes to identify a collision-free trajectory, even when the initial trajectory contained many collisions. Actually, the proposed strategy initiates BIT-NAZA WLHR to change the wheel-legged configuration, bringing different kinds of foothold polygon into existence while in a rolling motion. To our best knowledge, this is the first time that incorporates bounding box abstraction and covariant gradient optimization algorithm to design a numerical solver on a rigid object which is also a WLHR regarded as a deformable robot. In this case, the robot is capable of independently navigating in confined spaces via morphing configuration, negotiating certain obstacles. We suppress crash between each element on the morphing geometry and obstacles in confined space since whether the crash can happen relates to the signed distance field (SDF) whose array is calculated from a robot-centric obstacle information. Succinctly, our main contribution can be enumerated as follows:

- Present and fabricate a novel WLHR prototype with Stewart parallel mechanism. The parallel structure enable robot to afford more than 300 kg burden. The full software and hardware layouts are exhibited from an individual electric motor controller to perception sensor mapper, contributing to expose how the intricate system work from low to high level.
- Calculate the signed distance field from robot-centric obstacle dimension box, which is collect from lidar after filter the ground point cloud. The incremental linear fitting method and KD tree clustering approach were adopted to describe width and height of obstacle box.
- Introduce a deformable bounding box abstraction for WLHR model, which is incorporated with SDF representing obstacles to check collision.

- Propose a covariant gradients-based trajectory optimization formulation that is applied to generate smooth and collision-free motions for a rigid deformable WLHR through changing trunk height and wheelbase. Integrate the planning framework into morphing locomotion in confined space, satisfying the kinematic reachability, smoothness, and obstacle avoidance constraints.
- Validate feasibility of our methodology on a simulated model and practical WLHR prototype in four scenes. A suite of intensive analysis and comparison for algorithm computation and robot deformable properties are displayed through line chart and histogram.

## 2. A Wheel-Legged Robot (WLHR) Description and Framework Overview

The BIT-NAZA robot showed in Figure 1 is designed as a quadrupedal machine equipped with four active wheel groups at four pelmas respectively. The mechatronic structure of each leg is comprised of six stretchable linkages with electrical actuators in parallel. The parallel construction enhances the body's payload capability and possesses six freedom of degrees, rendering the feasibility of horizontally rotating the wheel orientation. Furthermore, there are a linear displacement encoder and a force sensor internally-installed in each linkage scaling the displacement and force along stretching orientation. An inertial measurement unit (IMU) is mounted on robot torso to supply angle, angular velocity, and angular acceleration data along three axes. The robot stands approximately 1.4 m high when every linkage reaches its neutral position, as well as can maximally afford 350 kg with a total mass of about 300 kg (encompassing overall hardware, batteries and sensors).



**Figure 1.** The physical prototype of BIT-NAZA robot.

The BIT-NAZA was developed as a research platform for unmanned autonomous moving missions in challenging environments. Accounting for running on rugged terrains, the wheel-legged robot is capable of switching different types of locomotion among regular quadruped stepping, full-wheeled driving, and stepping-rolling hybrid locomotion, in which the Stewart mechanism urges wheels and robot base to change relative positions constituting a deformable robot independently. The proposed system architecture contains multiple hardware and software components embodied in Figure 2. In addition to low-level actuator and middle-level gait control module operating in DOS



and VxWorks separately, the mapping data disposing and planning module leverages the program environment of ROS in Linux where the scheduled framework engaged in.

The block drawing scratched in Figure 3 depicts the exploited planning architecture with the input of point cloud data (PCD) collected from lidar or other depth perceptive sensors and user commands including reference velocity and requested target waypoint. The ultimate task of *Planner* is to plan online a pertinent sequence of trajectories for collision checking points distributed on bounding box, which allows the robot to traverse convex obstacles or pass confined space pursuing assigned waypoints. The LiDar equipped on robot accumulates point cloud data, which is transformed into width, height, and length dimension of obstacle in *Mapper* module. A deformable planning approach exploits covariant gradient method to prevent robot from obstacles through signed distance field, forming a smooth trajectory for bounding box abstraction. The optimal trajectories produced from *planner* are transformed into subsequent motion states for body and wheels in terms of current robot configuration through robot kinematic definition in *Motion Generator*. Additionally, wheeled odometry that accumulates the total turning angles of wheels in charge of the current location of robot and collects the trunk posture information from body-mounted IMU . The modular layer where *Motion Follower* formulates reactive maneuver helps us to seamlessly connect the *Motion Generator* and *Joint Controller*. The corresponding commands for different motion units are transferred and converted into joint tracking controller to follow the planned Stewart-based kinematic solution. The final output from our planning framework is actuator input containing required joint displacement, velocity and acceleration for 24 electrical cylinders and wheel motors.

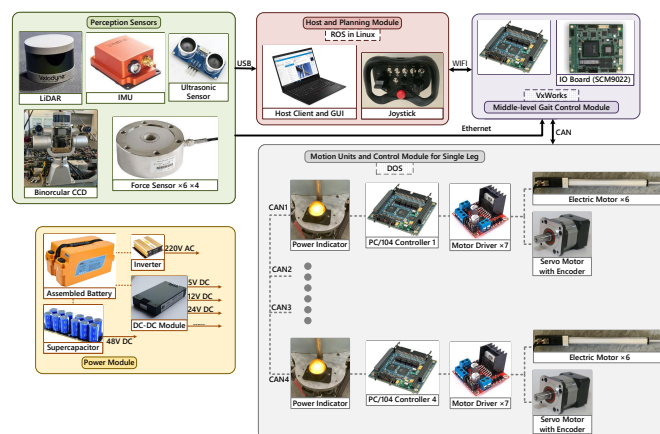
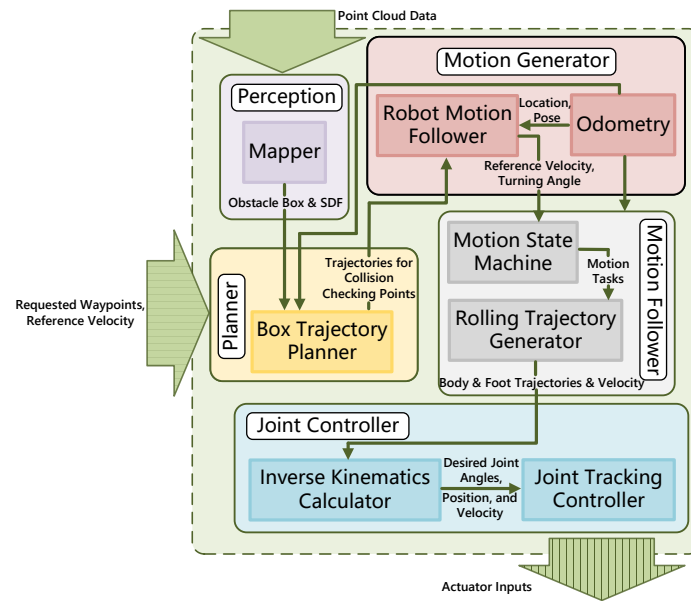


Figure 2. The hardware and software components of the WLHR system.



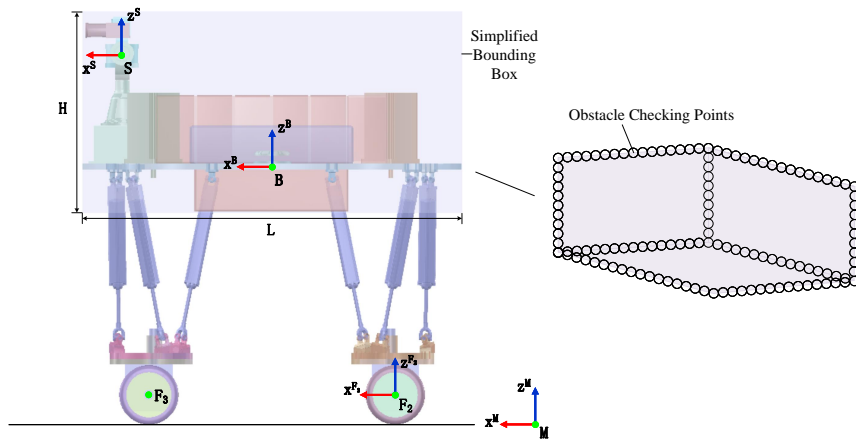
**Figure 3.** The schematic diagram of planning framework working on BIT-NAZA robot.

### 3. Problem Formulation

Different from standard quadruped locomotion demonstrated in Reference [33], this work only concentrated on full-wheeled driving and legged-rolling hybrid locomotion to follow rough terrain variation. The whole body motion trajectories in three-dimension are executed on five action units including four wheels and body base when legs adjust space configuration between these different units with simultaneous wheeled velocity. This activity not only leads to absolute motion for robot entirety but also activates the relative motions for robot components between each other. To adapt robot to several types of terrains and avoid the high-dimensional calculation restriction, we simplify this problem through outlining a deformable bounding box which can cover the entire robot body. The box fixed on robot base would not stretch downwards to accommodate wheel-legs but can widen or narrow according to united volume of the left and right wheel-legs. The obstacle avoidance in three-dimensional confined space for a WLHR with such large quantities of actuating joints, we simplified the robot model into an bounding box in order to find smooth trajectories for different joints that satisfied several constraints. The abstract box model is convenient for finding a collision-free trajectory and in charge of width and height of whole robot, playing an important role in the framework with our trajectory definition.

#### 3.1. Deformable Bounding Box Abstraction

For the purpose of avoiding obstacles, we allocate several collision checking points distributed on a bounding box  $\mathcal{O} \in \mathbb{R}^3$  which outlines the robot base with constant height and length but changeable width. The abstracted geometry is attached to the body coordinate frame  $\mathcal{B}$  as displayed in Figure 4. The simplification tremendously accelerates collision checking exploration. Entire kinds of motions are encapsulated, where the robot is able to perform while rolling in any height, direction, and width. Despite the bounding box does not encase legs, its width extending to reach the outermost points quantifies how width robot broadens. It means width of bounding box broaden or shrink its width over varying wheelbase, accounting for the lateral component of leg collision checking. If nothing in explored space collides with this geometry, the robot is ensured to be prevented from collisions.



**Figure 4.** The robot model and obstacle checking points on abstract outline, which are regarded as a part of planner input.

The lateral expansion of robot  $e$  is prescribed as half width of the bounding box so that the box expression is  $(L, 2 \cdot e, H)$  in three-dimensional space. A dense sampling distributes 10 cm interval between collision checking points on edges of the bounding box. The number of points relies on the volume and sampling density, resulting in 168 points in our case. Then we need to adopt motion sequence generator to find solutions for optimized trajectories satisfying these sufficient conditions.

### 3.2. Trajectory Definition

Once the simplified geometry is defined, the remaining theme which is whole body trajectories planning is guided in our routine. Iterating the route time and time again acquires the optimal solution, which is entailed as a series of configurations  $\zeta$  at time  $t$  from a start configuration  $\zeta(0)$  to an appointed configuration  $\zeta(1)$  in the three-dimensional description.

$$\zeta(t) = [x(t), y(t), z(t), \phi(t), e(t)] \quad (1)$$

in which  $x$ ,  $y$ , and  $z$  compose the position expression and  $\phi$  the yaw of the robot base  $B$  in map coordinate frame  $\mathcal{M}$ , then the planner does not involve pitch and roll in this work. The bounding geometry also deforms in accordance with time  $t$  through the extendable width as a function of time. In practice, the trajectories for collision checking points on edges of the bounding box are separately discretized via sampling interval. Furthermore, the  $i_{th}$  collision checking point can be calculated, which is relevant to the bounding box dimension definition.

$$\mathbf{c}_i^B = [c_x \cdot L, c_y \cdot e(z), c_z \cdot H]^T \quad (2)$$

where coefficient  $c_{x,y,z} \in [-1, 1]$  uniquely represents the value how coordinate of  $i_{th}$  collision checking point scales the coordinate of its corresponding vertex in body frame  $\mathcal{B}$ , so that  $\mathbf{c}^B$  is also a vector expression in frame  $\mathcal{B}$ . For instance, a collision checking point is situated at hind, left, and top vertex of bounding box, whose coefficients concerns  $(c_x, c_y, c_z) = (-1, 1, -1)$ . Besides, the coordinate term  $\mathbf{c}_i^B$  in body frame  $\mathcal{B}$  can be transformed into map frame  $\mathcal{M}$  as

$$\mathbf{c}_i^M(\zeta) = \mathbf{p}_B^M + {}^M\mathbf{R}_B \cdot \mathbf{c}_i^B \quad (3)$$

where  $\mathbf{p}_B^M = (x, y, z(e), \phi, e)$  indicates the known corresponding coordinates of collision checking point in map frame  $\mathcal{M}$ , rotation matrix  ${}^M\mathbf{R}_B$  signifies the known transformation matrix from body frame  $\mathcal{B}$  to map frame  $\mathcal{M}$ . Formally,  $e(z)$  in Equation (2) views  $e$  as a function of  $z$ , since it is anticipated

that the width of bounding box is able to deform as a consequence that robot lowers or lifts its trunk. Another matching expression in  $\mathbf{p}_B^M$  is written as inverse formulation.

Progressively, the respective kinematic Jacobian matrix of the  $i_{th}$  collision checking point  $\mathbf{J}_{c_i} = \frac{\partial}{\partial \xi} \mathbf{c}_i^M(\xi) \in \mathbb{R}^{3 \times 5}$  is inferred as

$$\begin{aligned} \mathbf{J}_{c_i} &= \left[ \mathbf{I}_{3 \times 2}, \mathbf{R} \cdot \frac{\partial}{\partial z} \mathbf{c}_i^B, \frac{\partial}{\partial \phi} \mathbf{R} \cdot \mathbf{c}_i^B, \mathbf{R} \cdot \frac{\partial}{\partial e} \mathbf{c}_i^B \right] \\ &= \begin{bmatrix} 1 & 0 & c_y \cdot \sin \phi \cdot \frac{\partial e(z)}{\partial z} & -c_x \cdot L \cdot \sin \phi + c_y \cdot e(z) \cdot \cos \phi & 0 \\ 0 & 1 & \cos \phi \cdot c_y \cdot \frac{\partial e(z)}{\partial z} & -\sin \phi \cdot c_y \cdot e(z) - c_x \cdot L \cdot \cos \phi & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial z(e)}{\partial e} \end{bmatrix} \end{aligned} \quad (4)$$

with differential notations  $\frac{\partial}{\partial z} \mathbf{c}_j^B$  and  $\frac{\partial}{\partial e} \mathbf{c}_j^B$  revealing how the lateral expansion changes with  $z$  as well as the converse fashion. This work regards the robot as a deformable configuration among the robot body and four wheels, whose constraint depends on its body reachability [34] resulting from Stewart inverse kinematic solutions. With the consideration of this formulation, we can establish the deformable relationship between  $e$  and  $z$  as linear, scaling between the restrictive maximum and minimum limits of the robot's workspace. Nevertheless, it could employ a more complicated mathematical description. For the purpose of guaranteeing optimizing efficiency, the work focused on a simple model which might potentially fail in a few very narrow spaces.

#### 4. Obstacles Representation Collected from VLP-16 Lidar

As for the procedure of acquisition and evaluation of terrain information in this work, an on-board terrain data server enforces a Velodyne VLP-16 LiDAR sensor to collect PCD, quantitating terrain topology that is adjacent to robot. The field of view (FOV) of VLP-16 ranges 360 degrees horizontally and  $\pm 15$  degrees vertically, whose available distance extends to 100 m. Additionally, the servo actuator mounted on turntable is responsible for sensor orientation adjustment to perceive farther region environment data.

After filter the ground point through multi-frame point cloud fusion and incremental linear fitting approach, the obstacle PCD is directly transformed into obstacle box information including corresponding dimension and distance from robot in map frame [35]. The topology relationship between discrete PCD is constructed by KD-Tree method. The Pairwise Linkage manages the obstacle cluster. Set a global distance  $d_c$  to distinguish neighborhood set and other PCD sets. For each point  $p_i$ , record corresponding nearest distance in sequence  $D_c$ . Calculate cut-off distance  $d_c$  with customized weight *scale* and mid-value in  $D_c$ :

$$d_c = scale \cdot median(D_c) \quad (5)$$

The point density is calculated through recent point number and Gaussian kernel function as follows:

$$\rho_i = \sum_{j \in [1, N], j \neq i} \exp \left( - \left( \frac{d_{ij}}{d_c} \right)^2 \right) \quad (6)$$

in which  $d_{ij}$  represents distance between point  $p_i$  and  $p_j$  inside epsilon neighborhood  $L_i$ .  $p_j$  is the nearest point whose corresponding density of is larger than that of  $p_i$  and mark that they are in a same cluster. If the density of  $p_i$  is a local maximum, we consider  $p_i$  as center of cluster. Project all of obstacle points on X-O-Y ground surface and X-O-Z surface model, we can compute precise width and height dimension of obstacle.

The metric which evaluates the distance from collision checking element  $\mathbf{c}_i(x, y, z) \in \mathbb{R}^3$  to the boundary of nearest obstacle enables the function  $\mathcal{D}(\mathbf{c}) : \mathbb{R}^3 \rightarrow \mathbb{R}$  to be pre-calculated and stored that employs Euclidean Transform Norm. A straightforward strategy is implemented to estimate obstacle potential in static exploring space. The SDF calculation is constructed as a pre-processing procedure before optimization to realize collision checking.



A workspace cost function  $u : \mathbb{R}^3 \rightarrow \mathbb{R}$  is to penalize the robot configuration  $\mathbf{q}$  in the workspace for serving collision checking elements  $\mathbf{c}_i \in \mathcal{O}$  to approach obstacles.

$$u(\mathbf{c}) = \begin{cases} \mathcal{D}(\mathbf{c}) + \frac{1}{2}\epsilon, & \text{if } \mathcal{D}(\mathbf{c}) < 0 \\ \frac{1}{2\epsilon}(\mathcal{D}(\mathbf{c}) - \epsilon)^2, & \text{if } 0 < \mathcal{D}(\mathbf{c}) \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Furthermore, we adopt the finite differencing to approximate the obstacle distance gradient  $\nabla u = (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z})$ , boosting the collision checking speed utilizing the primitives on geometric abstraction.

Subsequently, the SDF computation is updated with resolution in our case of 5 cm per pixel through Equation (7), whose disposing course is displayed in Figure 5. The top figures visualized PCD collection from Velodyne VLP-16, obstacle box visualization and SDF representation that describe the environment information. A robot-centric 4 m  $\times$  3 m  $\times$  0.6 m SDF layouts cell by cell with different color following the obstacle cost function  $u(\mathbf{c})$  variation.

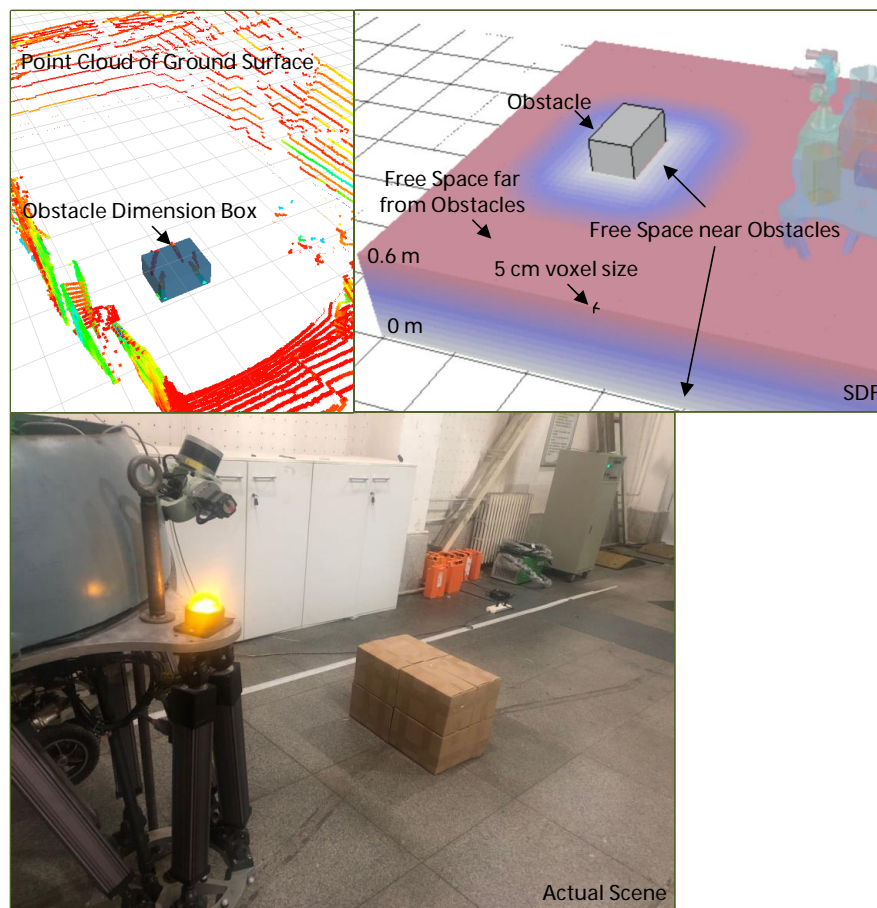


Figure 5. Obstacle description visualization in different transformation.

## 5. Planning and Motion Execution in Confined Space

Here we analyze how the mapping and planning module formalized in Figure 3, which operates deformable trajectory optimizing for robot obstacle negotiation in confined space. With the update rule of gradient optimization, the proposed framework produces a high-quality trajectory with three degrees of freedom which are both smooth removing unfeasible motions and collision-free. Originally, the simplified robot abstraction and SDF deriving from obstacle dimension are discretized

as geometric depiction. The planner exposition conceptualizes covariant gradient descent to minimize an objective over trajectory set through workspace and joint limit constrained by robot kinematics, along with a potential which measures the amount of element configuration for trajectory containing velocity and acceleration.

### 5.1. Deformation Planning

Formally, deformable abstraction is continuous and well transformed through the Jacobian Matrix. Our numerical optimal solver is pretty appropriate for this deformable planning issue. It discretizes trajectory into a sequence of  $n$  waypoints  $[\mathbf{q}_1^T, \dots, \mathbf{q}_n^T]^T \in \mathbb{R}^{n \times 5}$  over equal time duration  $\Delta t$  for robot base  $\mathcal{B}$ . The dynamical quantities is also evaluated utilizing finite differential for corresponding collision checking elements on the bounding box. It is ascertained that the initial configuration  $\xi(0) = \mathbf{q}_0$  and requested waypoint  $\xi(1) = \mathbf{q}_{n+1}$  are constant once they are assigned.

$$\xi_{k+1} = \xi_k - \frac{1}{\eta} A^{-1} \bar{\nabla} \mathcal{U}[\xi_k] \quad (8)$$

The iterating criterion for deformable optimization integrates upgrading rule that is functional covariant gradient descent with a Hessian matrix  $A$  behaving as a smoothing operator and expanding gradient across trajectories. The learning rate  $\eta$  determines how fast or slow we will move towards the optimal solution over each iteration  $k$ . For what the objective functional  $\bar{\nabla} \mathcal{U}[\xi_k]$  measures, two complementary aspects involve encouraging trajectory smoothness and refraining trajectories close to obstacles through the prescribed amount. Two gradient terms  $\bar{\nabla} \mathcal{F}_{obs}[\xi_k]$  and  $\bar{\nabla} \mathcal{F}_{smooth}[\xi_k]$  acting on trajectory  $\xi_k$  can be incorporated as a weighted sum.

$$\bar{\nabla} \mathcal{U}[\xi_k] = \bar{\nabla} \mathcal{F}_{obs}[\xi_k] + \lambda \bar{\nabla} \mathcal{F}_{smooth}[\xi_k] \quad (9)$$

where the prior term  $\bar{\nabla} \mathcal{F}_{smooth}[\xi]$  minimizes dynamical amount and is assumed as independent with environment from iteration to iteration.

$$\bar{\nabla} \mathcal{F}_{smooth}[\xi] = \frac{1}{2(n+1)} \sum_{t=1}^{n+1} \left\| \frac{\mathbf{q}_{t+1} - \mathbf{q}_t}{\Delta t} \right\|^2$$

The formulation is stated as finite differencing sense and can be rewrite as

$$\bar{\nabla} \mathcal{F}_{smooth}[\xi] = \frac{1}{2} \|K\xi + e\|^2 = \frac{1}{2} \xi^T A \xi + \xi^T b + c$$

with a pertinent finite differencing matrix  $K =$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \\ 0 & 0 & 0 & \dots & 0 & -1 \end{bmatrix} \otimes I_{(n+2) \times (n+2)},$$

a constant vector  $e = [-q_0^T, 0, \dots, 0, q_{n+1}^T]^T$  governed by boundary conditions  $q_0$  and  $q_{n+1}$ , as well as a constant  $c = \frac{e^T e}{2}$ . Accordingly, the acceleration and velocity amount are quantified via Hessian as mentioned above  $A = K^T K$  and gradient  $b = K a^T e$ .

Alternatively, the obstacle avoidance gradient  $\bar{\nabla} \mathcal{F}_{obs}[\xi]$  prevents per collision checking element from nearing obstacles, or already in a collision.

$$\bar{\nabla} \mathcal{F}_{obs}[\xi] = \sum_{i=1}^c J_{c_i}^T \|\mathbf{X}'\| [(\mathbf{I} - \hat{\mathbf{X}}' \hat{\mathbf{X}}'^T) \nabla u - u \mathcal{K}] \quad (10)$$

in which  $\mathbf{J}_{c_i}$  indicates the corresponding position Jacobian matrix mapping the robot configuration  $\mathbf{q} \in \xi$  to the position of collision checking element  $\mathbf{j}_i$ , which is computed in Equation (4) and finally sums their total quantity  $\mathcal{C}$ , as well as  $\mathcal{K} = \|\mathbf{X}'\|^{-2} (\mathbf{I} - \hat{\mathbf{X}}' \hat{\mathbf{X}}'^T) \mathbf{X}''$ , denotes curvature vector along the workspace trajectory tracked by a collision checking element.  $\mathbf{X}'$  and  $\mathbf{X}''$  signify the velocity and acceleration of collision checking element,  $\hat{\mathbf{X}}'$  expresses the normalized velocity vector. To maintain the workspace gradient as an update direction, the matrix  $\mathbf{I} - \hat{\mathbf{X}}' \hat{\mathbf{X}}'^T$  projects workspace gradient orthogonally to the motion orientation, averting direct influence on the speed profile of trajectory.

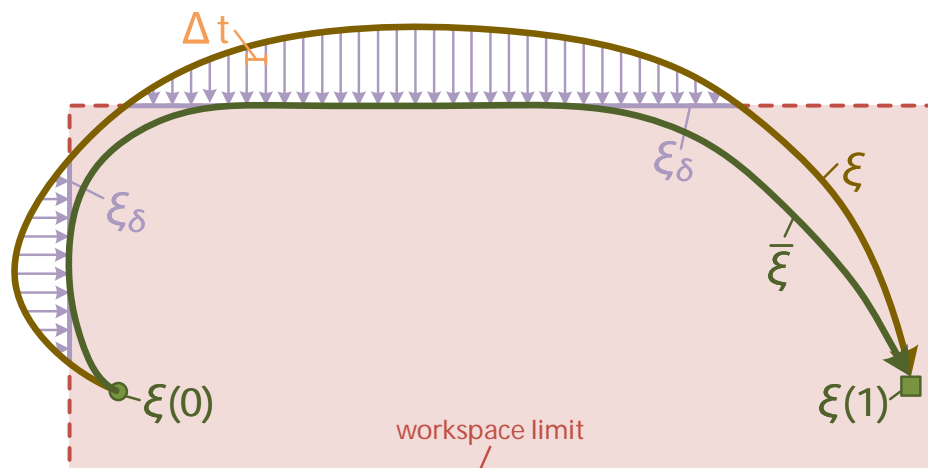
## 5.2. Robot Workspace Constraints

To make the optimization query manageable, we need to develop a profitable constraint about robot model. In practice, let the kinematic measurement  $\mathbf{f}(\mathbf{q}_j, \mathbf{c}) : \mathbb{R}^3 \in \mathcal{O}$  quantity the collision checking elements  $\mathbf{c}$  on bounding box  $\mathcal{O}$ , given robot configuration  $\mathbf{q}_j$ . The body and leg workspace account for the robot deformable criterion limits and are determined by inverse kinematics of Stewart structure. The specific definition and description has been illustrated in Reference [33]. However, a trajectory  $\xi$  is collision-free if the distance from any collision checking point  $\mathbf{c}_i$  to any obstacle is longer than default threshold  $\epsilon > 0$  and satisfies the case where each robot configuration belongs to the trajectory ( $\mathbf{q}_j \in \xi$ ).

A practical circumstance is a fact that after the updating course illustrated in Section 5.1, much or less part of trajectory configuration might exceed the robot workspace limit even though conducted planning happens in confined space. When all of configurations are optimized for body workspace, the trajectory generation method maintains footholds contacting with ground that is indispensable. With this prerequisite condition, reachable zone of Stewart wheel-leg determines body workspace limit. For the initial purpose of sustaining the trajectory smoothness, one ideal scheme is to create a projection  $\xi_\delta$  onto the collection of reachable workspace once a workspace violation is recognized. We enforce the approximated projection technique applying the Riemannian metric  $\tilde{\xi}_\delta = A^{-1} \xi_\delta$ , where  $\xi_\delta$  is equipped with the most significant feasible boundary of workspace shown in Figure 6. Simultaneously, eliminate workspace violation through measuring value  $\alpha$  such that

$$\bar{\xi} = \xi + \alpha \tilde{\xi}_\delta$$

attains the future optimized trajectory  $\bar{\xi}$ , until it is entirely inside the workspace limit or iterated to a maximum number times.



**Figure 6.** The smooth projection technology for workspace limit demonstrated in two-dimensional plane.

Furthermore, it is necessary for robot to achieve a smooth transition between two sequential body configuration within one time segment, avoiding acceleration generation. We specify a maximum step interval constraint  $s_{max}$  which is described through euclidean distance to ensure the transition.

### 5.3. Inverse Kinematic Controller

The wheel-quadruped robot can be regarded as a floating-base system, in which the state variables  $\mathbf{x} = [\mathbf{p}_B^T \ \boldsymbol{\theta}_B^T]^T \in \mathbb{R}^{6 \times 1}$  denote the position and Euler angle attitude of robot body with respect to body frame  $\mathcal{B}$ . The velocity vector of  $i_{th}$  wheel-leg end-effector in map frame  $\mathcal{M}$  can be written as follows.

$${}^{\mathcal{M}}\dot{\mathbf{F}}_i = {}^{\mathcal{M}}\dot{\mathbf{p}}_B + \boldsymbol{\omega} \times {}^{\mathcal{M}}\mathbf{R}_B({}^B\dot{\mathbf{F}}_i + {}^B\mathbf{J}_i\dot{\mathbf{l}}_i), \quad (11)$$

where  $\boldsymbol{\omega}$  denotes angular speed of robot body in map frame  $\mathcal{M}$ ,  ${}^B\mathbf{J}_i$  represents Jacobian mapping from  $i_{th}$  foot point to corresponding joint variables with respect to body frame  $\mathcal{B}$ .  $\mathbf{l}_i \in \mathbb{R}^{6 \times 1}$  is the actuator position inputs for  $i_{th}$  Stewart mechanism wheel-leg. According to theorem of cross product  $\boldsymbol{\omega} \times {}^{\mathcal{M}}\dot{\mathbf{F}}_i = -{}^{\mathcal{M}}\dot{\mathbf{F}}_i \times \boldsymbol{\omega}$  and  ${}^{\mathcal{M}}\mathbf{J}_i = {}^{\mathcal{M}}\mathbf{R}_B{}^B\mathbf{J}_i$ , we have

$$\dot{\mathbf{F}}_i = \dot{\mathbf{p}}_B - \mathbf{S}({}^{\mathcal{M}}\dot{\mathbf{F}}_i)\boldsymbol{\omega} + {}^{\mathcal{M}}\mathbf{J}_i\dot{\mathbf{l}}_i, \quad (12)$$

in which  ${}^{\mathcal{M}}\dot{\mathbf{F}}_i \times = \mathbf{S}({}^{\mathcal{M}}\dot{\mathbf{F}}_i)$  is skew-symmetric matrix. Hereafter, with describing compound motion of robot body and actuators, Equation (12) can be rewritten as matrix formulation that regards  $\mathbf{I}$  as a unit matrix.

$$\dot{\mathbf{F}}_i = \begin{bmatrix} \mathbf{I} & -\mathbf{S}({}^{\mathcal{M}}\dot{\mathbf{F}}_i) & {}^{\mathcal{M}}\mathbf{J}_i \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{l}}_i \end{bmatrix} \quad (13)$$

Let  $\mathbf{J}_i = [\mathbf{I} \quad -\mathbf{S}({}^{\mathcal{M}}\dot{\mathbf{F}}_i) \quad {}^{\mathcal{M}}\mathbf{J}_i]$ , this equation becomes

$$\dot{\mathbf{F}}_i = \mathbf{J}_i \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{l}}_i \end{bmatrix} \quad (14)$$

Calculate the corresponding motion of robot in map coordinate frame  $\mathcal{M}$  through full-rank pseudo-inverse of  $\mathbf{J}_i$  that is  $\mathbf{J}_i^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$ .

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{l}}_i \end{bmatrix} = \mathbf{J}_i^+ \dot{\mathbf{F}}_i. \quad (15)$$

Due to contacting with ground of wheel-leg while robot is rolling through confined space, we can compute actuator trajectories of each electrical cylinder via robot body's motion.

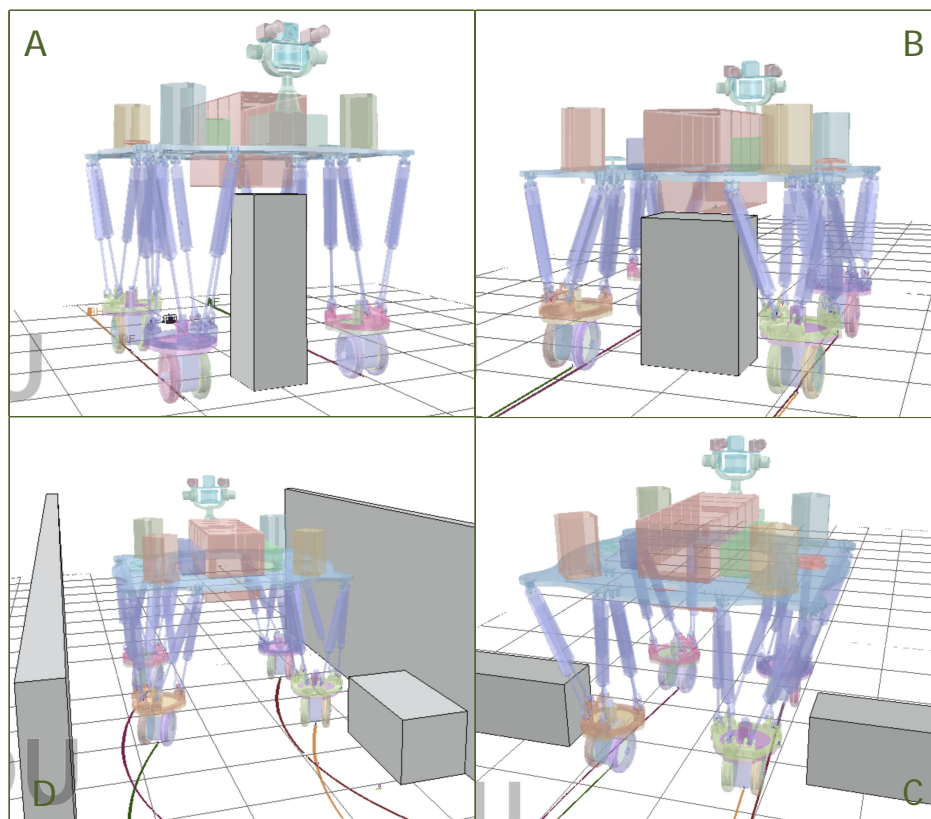
$$\dot{\mathbf{l}}_i = {}^{\mathcal{M}}\mathbf{J}_i^{-1}(\dot{\mathbf{p}}_B - \mathbf{S}({}^{\mathcal{M}}\dot{\mathbf{F}}_i)\boldsymbol{\omega}). \quad (16)$$

## 6. Results and Discussions

This section represents and discusses the simulation and experiment results based on the entire planning framework shown in Figure 3. Simulations are applied on a simulated BIT-NAZA robot model in Virtual Robot Experimentation Platform (V-REP) furnishing with VORTEX dynamic engine. The BIT-NAZA robot described in Section 2 is used for the experiment verification. Our ultimate prospect is to plan a set of trajectories for COM and deformable footholds polygon from the current configuration to a requested waypoint and traverse the obstacles in various challenging exercises. Four sorts of essential implements endeavor to examine whether the motion generated from the proposed framework is feasible, containing broaden accommodation, lift overhang, rotating clearance, slim gap assignments. All of the planning trajectories were created on the basis of SDF representation of obstacle produced by PCD and obstacle box dimension. We underline that each motion appeared deformably

credits to assigned waypoints, user-input velocity, and situational requirement. It also showcased that further guidance in the form of how our method can solve analogical issues with remarkably better terrain-adaptive performance and computational efficiency assessment.

Progressively, Figure 7 depicts the produced motion that deformably performs the negotiating capability in four diverse assignments. Each task starts from the neutral configuration when all of the electronic cylinders extend to half of their maximum length as a preparation to willingly shrink and expand. For lift overhang, the robot has to lift its trunk, which results in a corresponding increment in width. We achieved this performance through coupling iterating update Equation (8) with bounding box abstraction through Jacobian  $\frac{\partial}{\partial z} \mathbf{c}_j^B$ , which was notably mapped to the changes over Z-position  $z$  that grows. Briefly, the selecting treatment of initial trajectory  $\xi_0$  was stochastic and even already collides with some obstacles.  $\nabla \mathcal{F}_{obs}[\xi_i]$  and stored SDF prevent bounding box from obstacles and then proceed with numerical optimization. Broaden accommodation event enables the robot to broaden its span and increase the width of bounding box, as evidenced by the linear implementation of  $\frac{\partial}{\partial e} \mathbf{c}_i^B$ . Slim gap function allows the robot to shrink the width of the bounding box around the body, which is equivalent to that of foothold polygon and passes over the narrow lane, also motivating the differential application of  $\frac{\partial}{\partial e} \mathbf{c}_i^B$ . Rotate clearance behavior is an extraordinarily non-deformable action that requires the robot to orient its wheels and detour obstacle in a narrow passage. The configurational parameter edits, the steering angle of wheels  $\theta(t)$ , are regulated by synergistic trajectory command in gradient descent update. We note that to obtain these motions, it is necessary to define an input moving velocity 0.1 m/s for active wheels from user manual law. To diminish the frictional error from contacting wheels with the ground and possible slippage error from high dynamic wheeled turnings, mastering adequately low wheeled speed features of great concern.



**Figure 7.** Four assignments to default destination in V-REP simulation: (A) Lift overhang routine (B) Broaden accommodation routine (C) Rotate clearance routine (D) Slim gap routine.



### 6.1. Simulations

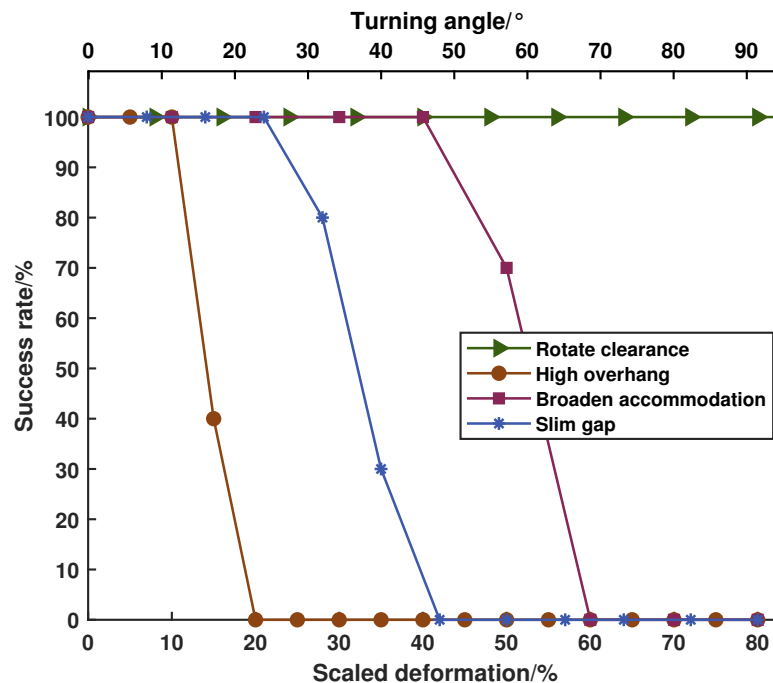
The simulation experiments can help us to analyze and compare the result of the proposed algorithm and robot configuration variation, which is evaluated in 10 trials for each circumstance and obstacle event. In the first *lift overhang* case (Figure 7A), we progressively change the height of obstacle with increments of 5 cm additional passable height while rolling with neutral configuration. Each time it enables robot to lift accommodate space to pass through obstacle. We found robot could pass over 90 cm obstacle with 100% success rate and the supreme 105 cm with 40%.

During the *broaden accommodation* event (Figure 7B) in simulation, broadening wheelbase to accommodate wider obstacle rather than bypassing is a better approach to enhance motion efficiency. We examined this case on obstacles with different width and found that robot can pass a 62 cm width box with 100% success and 70 cm with 70% success separately in 10 trials.

Subsequently in the third simulation (Figure 7C), we gradually shrink the gap breadth with reduction of 10 cm to test robot. It is demonstrated that robot could get through a 1.2 m gap with 100% success, 1.1 m with 80%, and 1 m with 30%.

The *rotate clearance* in (Figure 7D) steers robot to change wheel orientation without altering configuration, avoiding obstacle. How much the the maximum turn angle is relies on distance between obstacle and robot. We fixed obstacle on a spot 1 m far from robot and changed different goal to evaluate the turning angle that is  $90^\circ$  with 100% success.

Although the success rate shown in Figure 8 described the performance of our planning approach, the results may change when we select different parameters. There are many user-defined parameters in the optimization process, such as the learning rate  $\nabla$  and the default obstacle threshold  $\epsilon$ . Larger learning rate can obtain larger trajectory update velocity, boosting sampling interval for next trajectory. But it could miss the local minima point, resulting in a smaller objective function point. Enlarging default obstacle threshold can increase obstacle avoidance ability that prevents trajectory farther from obstacle, but exploring an optimal trajectory in extremely confined space becomes difficult. In our method, the user-defined parameters tuning method generally is based on our experiences or prior knowledge. For example, when optimization fails, we will try to reduce the learning rate. In our future work, we will attempt to apply learning-based methods to select and optimize those parameters intelligently.



**Figure 8.** Success rates of increasingly conditional restraints over 10 trials, which are verified in simulation for four specific events.

## 6.2. Performance Evaluation

For the purpose of graphically visualizing the comparison for deformable adaptation percentage among four mentioned assignments, and we analyze it in simulated cases for security. Furthermore, we examined the success rate of the proposed planner and recognized that it is precisely the absence of diverse environment restriction influences the success rate. 10 simulated examinations were governed for each specific event and condition, whose results are depicted in Figure 8. Only the examining condition for *rotate clearance* case is rotating angle, while the ones for the other three are scaled deformation percentage defined in Section 6.1, which equals to the maximum deformable measurement as a percentage of original volume. Obviously, the maximum steerable angle  $\pm 90^\circ$  evaluated by Stewart kinematics, the success rate for all of rotatable angle in testing *rotate clearance* task reached 100%. Commonly, the success rate of deformable planning algorithm reaches 100% when conditional constraints of target terrain templates satisfied our model abstraction limits, and its variation tendency matches and are smaller than the deformable scale we calculated through kinematic analysis in accordance with Reference [33]. For instance in *lift overhang*, the robot body can be lifted maximum 15 cm that equals to 15% of robot's neutral height, but the success rate of corresponding environment for robot to attain this configuration is only 40%.

## 6.3. Experiments

In addition to experiments, we implemented the entire pipeline from Figure 3 on the BIT-NAZA WLHR such that they demonstrate the deformable capability. Likewise, all of the obstacle-negotiation tasks in a confined space were also performed in realistic situations, as disclosed in Figure 9. The robot detected obstacle locations  ${}^S\mathbf{P}_i$  in sensor frame and transformed it together with robot configuration on uniform map frame  $\mathbf{M}$ , which is defined though the projection of initial COM position and foot contacting surface. The robot posture can be measured by IMU and height can be computed through basic mechanism size and Stewart forward kinematics through extending amount of electrical cylinders. In each experiment, we generated a SDF description around robot with  $6\text{ m} \times 4\text{ m} \times 3\text{ m}$  dimension scanned from Velodyne-16 LIDAR, which has been elaborated in Section 4. To record the final

trajectories generated with four wheels and COG of the robot, we take advantage of displacement sensors embedded in each electrical cylinder. They allow Newton-Raphson iteration of forward kinematics to calculate lateral, altitude trajectories and rotation angles of four Stewart mechanism. The wheeled odometry accumulates and averages the traveled distance of whole robot, that is COM. Apart from the environment perception sensor that VLP-16 LIDAR, we employed an additional camera jointly installed on the controllable holder, identifying the white color line on the ground. Especially in *rotate clearance* assignment, stamping white lines adhered to the ground as 1.6 m-high walls which generates a cluster of corresponding SDF, as a result of avoiding superfluous scene construction. Due to the field size constraint, we only can test each task for one way planning and all of goals are about 3 m far away from COM.

Note that we utilized the RGB color convention to draw diverse action units and their relationship with obstacles in Figure 10, demonstrating the executive smooth trajectories. The thick brown and green line fractions represent front and hind wheeled trajectories separately, whose corresponding thin lines and enclosed shadow area indicate their dimensional boundary of wheels. The thick strawberry line describes the COM trajectory, how the truck border moves were also portrayed in the same color. Besides, the gray shadow boxes present obstacles in various sizes and scenes. It is sufficiently apparent that robot could negotiate different obstacles and reach the user-specified target waypoints, while deformable iterations optimized the entire motion generation in four experimental cases.

While conducting the first case (Figure 9a), robot encountered an obstacle which had increments of 9 cm additional passable height while rolling with neutral configuration, that was a totally 1 m high box. The goal of the *lift overhang* task is to enable the robot to explore the clearing space upwards, finally recovering to its initial arrangement.

Over the *broaden accommodation* task (Figure 9b), the robot spanned about 7 cm and 12 cm towards left and right side independently for the obstacle 2.4 m far away. The left and right wheels turn towards outside to enlarge the support polygon which assists robot with the deformable task of accommodating relatively wider obstacle beneath its trunk.

Whereas (Figure 9c) constructed by two obstacles with close distance, the execution of *slim gap* routine demonstrated the adaptive capability to a narrow passing space which possessed the width of 1.3 m. The foothold support polygon shrunk about 12 cm and 9 cm towards left and right respectively. As the width of the bounding box determines the lateral shrinkage, the lateral displacements for front and hind wheels which are on the same side are the same.

The final case (Figure 9d) performed the *rotate clearance* conduction without a morphological configuration of wheelbase. As expected, the proposed planner instructs robot to rotate and get round the right box, steaming into free space on the left side. The left and right walls are too high to cross, so it is impotent to conduct *lift overhang* behavior and reasonably mastery the robot in a confined space. The dimension of the obstacle are length of 0.8 m and width of 0.43 m.

Besides, we examine and compare the numerical traits of path length and planning time for four diverse simulated trials, whose results are illustrated in Figure 11. In general, we choose a goal which is about 3 m far from current robot configuration and explore the target trajectory. Although the path lengths for four trials were almost the same, the planning time displayed irregularly in which the one for *slim gap* event was the supreme example. The quantity and intensity of obstacles, as well as unoccupied space in narrow environment directly influence the collision checking efficiency and planning time.

Moreover, it can be carefully scrutinized that after traversed obstacle, the planner had been continuing to gradually lift the robot to extend the trajectory, which picked *lift overhang* behavior as an instance. The appearance of this phenomenon denotes that covariant gradients optimization undertakes searching trajectory with the lowest objective functional metric  $\bar{\nabla}\mathcal{U}$  where obstacles end up inside current examined coverage between collision checking points. This developed into a trade-off among weighted parts  $\bar{\nabla}\mathcal{F}_{obs}$  and  $\bar{\nabla}\mathcal{F}_{smooth}$ , since simply adjusting  $\bar{\nabla}\mathcal{F}_{obs}$  higher than  $\bar{\nabla}\mathcal{F}_{smooth}$  could cause a more significant possibility of collisions.



(a) lift overhang task



(b) Broaden accommodation task

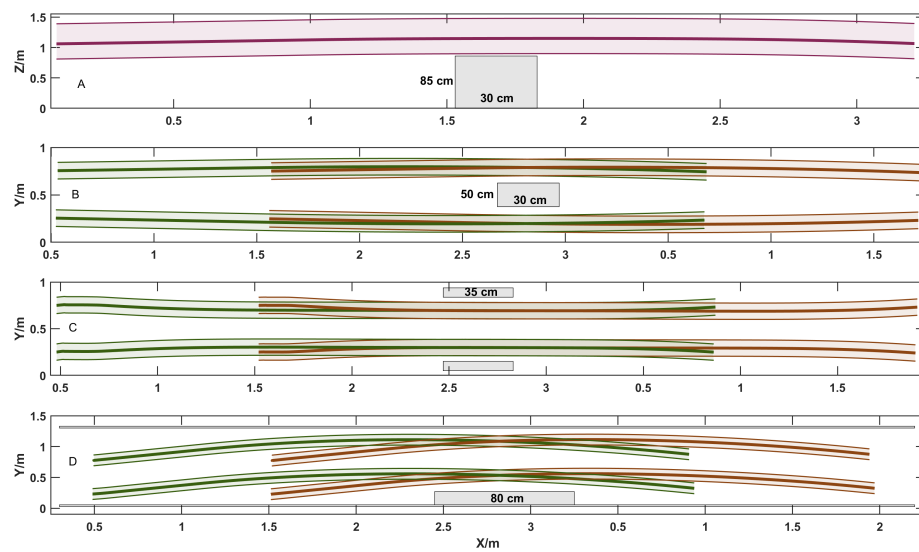


(c) Slim gap task

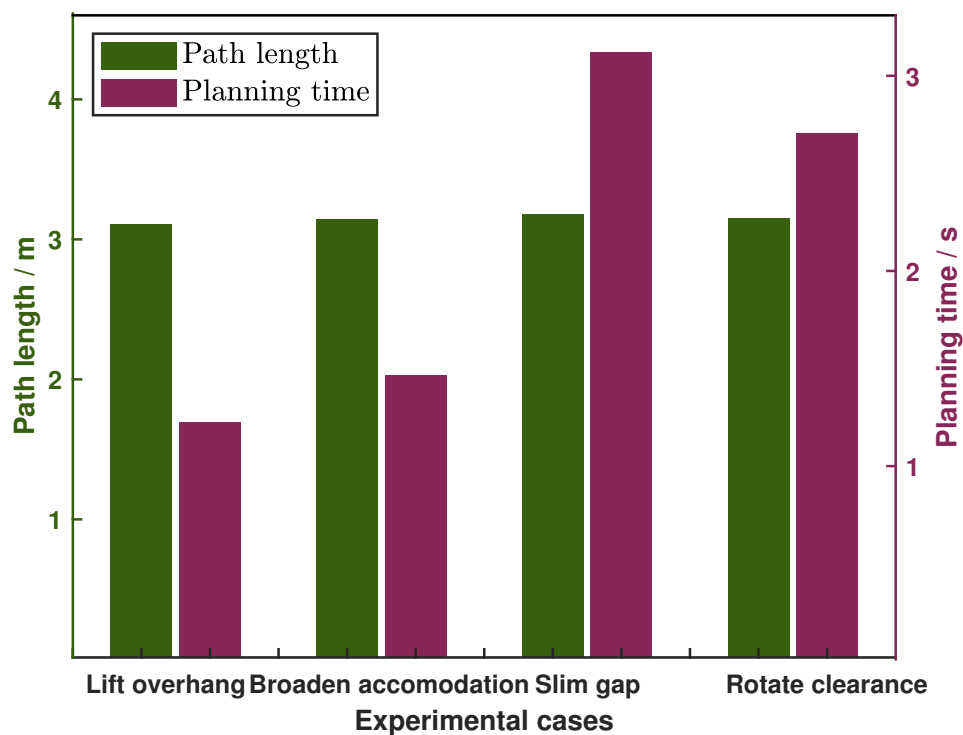


(d) Rotate clearance task

**Figure 9.** Realistic experiments of four tasks conducting on BIT-NAZA robot.



**Figure 10.** Experiment results from four deformable planning and collision avoidance tasks in confined space.



**Figure 11.** Planning profile plots for four experimental assignments.

## 7. Conclusions

To obtain a versatile trajectory optimization formulation for WLHR, we presented a novel pipeline system based on point cloud data to negotiate obstacle in confined space. With the omnidirectional rolling ability, the wheel-legged hybrid locomotion further exhibited how to leverage a deformable bounding box to abstract the robot model and significantly simplified calculation complexity while merging into the planner. Given PCD around the robot, we developed a suite of obstacle-representing



approach that pre-create SDF to store before optimization. This allows our planner to interactively produce deformable trajectories when robot desires to reach requested waypoints. Four assignments that contain *lift overhang*, *broaden accommodation*, *slip gap*, and *rotate clearance* are consequently conducted as compelling examinations for the proposed framework. The simulation analyzed its success rate and limitation for different circumstance. In conclusion, experiments demonstrated the feasibility of our deformable adaptation technique.

The proposed approach manages deformable motion planning and intensifies moving dexterity of wheel-legged robot. Compared to common wheeled robot with fixed configuration, our framework enables robot to go through narrow spaces and roll along a shorter path when confronts with an obstacle.

**Author Contributions:** F.G.: Methodology, Software, Writing; S.W.: Formal Analysis, Investigation, Resource; B.Y.: Experiment, Data Curation; Validation; J.W.: Supervision; Mechanism Project Administration. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by Natural Science Foundation of China: 61773060.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

COM    Center of Mass

## References

1. Fahmi, S.; Mastalli, C.; Focchi, M.; Semini, C. Passive Whole-body Control for Quadruped Robots: Experimental Validation over Challenging Terrain. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2553–2560. [\[CrossRef\]](#)
2. Hua, D.; Guiyang, X.; Guoliang, Z.; Michael, M. Gait and trajectory rolling planning and control of hexapod robots for disaster rescue applications. *Robot. Auton. Syst.* **2017**, *95*, 13–24.
3. Wensing, P.M.; Wang, A.; Seok, S.; Otten, D.; Lang, J.; Kim, S. Proprioceptive actuator design in the MIT Cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Trans. Robot.* **2017**, *3*, 509–522. [\[CrossRef\]](#)
4. Peng, X.B.; Berseth, G.; van de Panne, M. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Trans. Graph.* **2016**, *35*, 81. [\[CrossRef\]](#)
5. Klamt, T.; Schwarz, M.; Lenz, C.; Baccelliere, L.; Buongiorno, D.; Cichon, T.; Di Guardo, A.; Droeschel, D.; Gabardi, M.; Kamedula, M.; et al. Tsagarakis and Sven Behnke. Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance. *J. Field Robot.* **2020**, *37*, 889–919. [\[CrossRef\]](#)
6. Zhao, Y.; Chai, X.; Gao, F.; Qi, C. Obstacle avoidance and motion planning scheme for a hexapod robot Octopus-III. *Robot. Auton. Syst.* **2018**, *103*, 199–212. [\[CrossRef\]](#)
7. Sezer, V.; Gokasan, M. A novel obstacle avoidance algorithm: “Follow the Gap Method”. *Robot. Auton. Syst.* **2012**, *60*, 1123–1134. [\[CrossRef\]](#)
8. Hoeller, F.; Schulz, D.; Moors, M.; Schneider, F.E. Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps. In Proceedings of the International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 1260–1265.
9. Rashid, A.T.; Ali, A.A.; Frasca, M.; Fortuna, L. Multi-robot collision-free navigation based on reciprocal orientation. *Robot. Auton. Syst.* **2012**, *60*, 1221–1230. [\[CrossRef\]](#)
10. Howard, T.M.; Kelly, A. Optimal rough terrain trajectory generation for wheeled mobile robots. *J. Field Robot.* **2007**, *26*, 141–166. [\[CrossRef\]](#)
11. Liu, P.; Yu, H.; Cang, S. Adaptive neural network tracking control for underactuated systems with matched and mismatched disturbances. *Nonlinear Dyn.* **2019**, *98*, 1447–1464. [\[CrossRef\]](#)
12. Wu, X.J.; Tang, J.; Li, Q.; Henga, K.H. Development of a configuration space motion planner for robot in dynamic environment. *Robot. Comput. Integr. Manuf.* **2009**, *25*, 13–31. [\[CrossRef\]](#)

13. Fankhauser, P.; Bjelonic, M.; Bellicoso, C.D.; Miki, T.; Hutter, M. Robust Rough-Terrain Locomotion with a Quadrupedal Robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018.
14. Wooden, D.; Malchano, M.; Blankespoor, K.; Howard, A.; Rizzi, A.A.; Raibert, M.; Fortuna, L. Autonomous Navigation for BigDog. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2012; pp. 4736–4741.
15. Belter, D.; Wietrzykowski, J.; Skrzypczynski, P. Employing Natural Terrain Semantics in Motion Planning for a Multi-Legged Robot. *J. Intell. Robot. Syst.* **2019**, *93*, 723–743. [[CrossRef](#)]
16. Perrin, N.; Ott, C.; Engelsberger, J.; Stasse, O.; Lamiroux, F.; Caldwell, D.G. Continuous legged locomotion planning. *IEEE Trans. Robot.* **2016**, *33*, 234–239. [[CrossRef](#)]
17. Menna, M.; Gianni, M.; Ferri, F.; Pirri, F. Real-time Autonomous 3D Navigation for Tracked Vehicles in Rescue Environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 696–702.
18. Norouzi, M.; Miro, J.V.; Dissanayake, G. Planning High-Visibility Stable Paths for Reconfigurable Robots On Uneven Terrain. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 2844–2849.
19. Brunner, M.; Bruggemann, B.; Schulz, D. Motion Planning for Actively Reconfigurable Mobile Robots in Search and Rescue Scenarios. In Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, College Station, TX, USA, 5–8 November 2012; pp. 1–6.
20. Le, A.V.; Prabakaran, V.; Sivanantham, V.; Mohan, E. Modified AStar Algorithm for Efficient Coverage Path Planning in Tetris Inspired Self-Reconfigurable Robot with Integrated Laser Sensor. *Sensors* **2012**, *8*, 2585.
21. Le, A.V.; Prabakaran, V.; Sivanantham, V.; Mohan, E. Autonomous Navigation for Reconfigurable Snake-like Robots in Challenging, Unknown Environments. *Robot. Auton. Syst.* **2017**, *89*, 123–135.
22. Short, A.; Bandyopadhyay, T. Legged motion planning in complex three-dimensional environments. *IEEE Robot. Autom. Lett.* **2018**, *3*, 29–36. [[CrossRef](#)]
23. Chen, X.; Ghadirzadeh, A.; Folkesson, J.; Jensfelt, P. Deep reinforcement learning to acquire navigation skills for wheel-legged robots in complex environments. *Int. Conf. Intell. Robot. Syst.* **2018**, *4*, 3110–3116.
24. Hauser, K.; Latombe, J.C. Multi-modal motion planning in non-expansive spaces. *Int. J. Robot. Res.* **2010**, *29*, 897–915. [[CrossRef](#)]
25. Klamt, T.; Behnke, S. Anytime hybrid driving-stepping locomotion planning. *Int. Conf. Intell. Robot. Syst.* **2017**, *4*, 4444–4451.
26. Klamt, T.; Behnke, S. Planning hybrid driving-stepping locomotion on multiple levels of abstraction. In Proceedings of the International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018; pp. 1695–1702.
27. Daniel, S.; Klamt, T.; Behnke, S. Value Iteration Networks on Multiple Levels of Abstraction. In Proceedings of the Robotics: Science and Systems, Brisbane, Australia, 21–25 May 2006; pp. 1905–11068.
28. Grey, M.X.; Ames, A.D.; Liu, C.K. Footstep and motion planning in semi-unstructured environments using randomized possibility graphs. *Int. Conf. Robot. Autom.* **2017**, *29*, 4747–4753.
29. Orthey, A.; Escand, A.; Yoshida, E. Quotient-space motion planning. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018.
30. Buchanan, R.; Bandyopadhyay, T.; Bjelonic, M.; Wellhausen, L.; Hutter, M.; Kottege, N. Walking Posture Adaptation for Legged Robot Navigation in Confined Spaces. *IEEE Robot. Autom. Lett.* **2018**, *4*, 3110–3116. [[CrossRef](#)]
31. Zucker, M.; Ratliff, N.; Dragan, A.D.; Pivtoraiko, M.; Klingensmith, M.; Dellin, C.M.; Bagnell, J.A.; Srinivasa, S.S. CHOMP: Covariant hamiltonian optimization for motion planning. *Int. J. Robot. Res.* **2018**, *32*, 1164–1193. [[CrossRef](#)]
32. Raghavan, V.S.; Kanoulas, D.; Laurenzi, A.; Caldwell, D.G.; Tsagarakis, N.G. Variable Configuration Planner for Legged-Rolling Obstacle Negotiation Locomotion Application on the CENTAURO Robot. *Int. Conf. Intell. Robot. Syst.* **2018**, *4*, 3110–3116.
33. Guo, F.; Wang, S.; Wang, J.; Yu, H. Kinematics-searched framework for quadruped traversal in a parallel robot. *Ind. Robot. Int. J. Robot. Res. Appl.* **2019**, *47*, 267–279. [[CrossRef](#)]

34. Gia, L.V.; Koo, I.M.; Tran, D.T.; Park, S.; Moon, H.; Choi, H.R. Improving traversability of quadruped walking robots using body movement in 3D rough terrains. *Robot. Auton. Syst.* **2011**, *59*, 1036–1048.
35. Gia, L.V.; Koo, I.M.; Tran, D.T.; Park, S.; Moon, H.; Choi, H.R. Obstacle information detection method based on multiframe three-dimensional lidar point cloud fusion. *Opt. Eng.* **2019**, *58*, 116102.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).