Caterpillar Heuristic for Gait-Free Planning with Multi-legged Robot

David Valouch and Jan Faigl

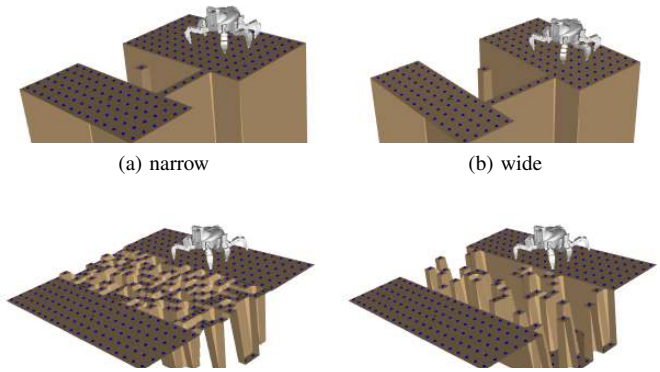
***Abstract*—In this paper, we address path planning for the quasi-static locomotion of a multi-legged walking robot on terrains with limited available footholds, such as passing a water stream over rocks. The task is to find a feasible sequence of steps to navigate the robot in environments where precise foot placement and order of the leg movements are necessary for successful traversal. A finite set of the considered footholds forms a state-space search domain, where states are defined by pairing the robot legs with footholds. The actions represent the connectivity of submanifolds of the robot configuration space approximating the robot's kinematic constraints indicating possible steps in a given stance. We propose a novel heuristic that significantly reduces the number of expanded states in the A* planner by avoiding local minima exhibited by commonly used heuristics. The computational requirements are nearly an order of magnitude lower than for the existing contact-driven solutions reported in the literature for similarly formulated planning problems. The viability of the proposed approach is further supported by an experimental deployment.**

***Index Terms*—Legged Robots; Motion and Path Planning; Multi-Contact Whole-Body Motion Planning and Control**

I. Introduction

MULTI-LEGGED walking robots can be considered more versatile platforms than wheeled or tracked robots as they can deliberately choose their interactions with the terrain using their legs. The enhanced locomotion capabilities come at the cost of increased computational requirements of the control and planning [1], [2]. A possible solution is to employ a regular motion gait pattern such as tripod gait that effectively transforms the multi-legged robot into a platform controllable by a velocity vector [3]. However, the robot would lose its advantage to place its legs individually on sparse footholds [4].

The presented research focuses on scenarios where simplifications of repetitive motion patterns are not applicable, but we still need a solution suitable for online deployment without off-line pre-computations [5]. Since the six legs of hexapod walking robots, compared to bipeds or quadrupeds [6], enable the robot to move in a *quasi-statically* stable manner while maintaining redundant contact with the environment, the robot can move in a way that cannot be described by a regular gait pattern. We call such a motion *gait-free* and the addressed problem *gait-free planning*. The problem is to traverse terrains with only limited footholds, where it is necessary to adjust the center of gravity to reach the next footholds, as in the example depicted in Fig. 1.



(a) narrow (b) wide

(c) $p_{\text{sparse}} = 0.2$ (d) $p_{\text{sparse}} = 0.8$

Fig. 1. Studied "gap-crossing" where gait-free planning is necessary to traverse the terrain with only limited footholds. Limited foothold scenarios generated for dense and spare sparsity level p_{sparse} are in the bottom row.

The proposed approach follows [7], [8] with partial mapping of the robot effectors to a set of footholds called *stance*. Each stance entails non-linear constraints defining a feasible subset of the configuration space. Finding a sequence of stances with appropriate footholds and guaranteed motion feasibility between stances is very demanding [9]. Nevertheless, finding the whole plan is necessary for limited available footholds when short-horizon planning would not yield a feasible solution. Selecting footholds after planning the motion of the body, such as [10], [11], might lead to situations where no suitable footholds are available. Therefore, finding a plausible sequence with guaranteed motion feasibility has been studied [6], [8], [12]. However, the existing methods are computationally demanding, with the reported times in tens of minutes, which is not usable for online deployment.

We identified that the computational bottleneck is caused by the heuristic in the state-space search finding the plausible sequences that guide the search into a local minimum, leading to excessive state expansions. The proposed quick-to-compute heuristic is motivated by caterpillar motion, where, in forward motion, the body center can move backward locally. Thus, the heuristic eliminates some local minima by suggesting expanding states, yielding plausible paths. The evaluation results support the online deployment of the developed precise

Manuscript received: February, 23, 2023; Revised May, 22, 2023; Accepted June, 20, 2023.

This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported by the Czech Science Foundation (GAČR) under research project No. 21-33041J. The access to the computational infrastructure of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" is also gratefully acknowledged.

The authors are with the Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic. {valoudav, faiglj}@fel.cvut.cz

Digital Object Identifier (DOI): see top of this page.

motion planning.

The paper is organized as follows. Existing approaches to precise motion and gait-free planning for multi-legged walking robots are overviewed in the following section. The problem is formally introduced in Section III and the proposed approach in Section IV. Results from the empirical evaluation and experimental deployment are presented in Section V. Concluding remarks are in Section VI.

II. Related Work

Planning for multi-legged robots can be roughly categorized into *contact-driven* and *body-driven*. In contact-driven planning, the robot motion is secondary to and driven by the sequence of contacts. So, a sequence of plausible steps is planned first, and the final motion is planned second, verifying the sequence. In contrast, body-driven planning determines the robot's body path first. Then, the contacts (footholds) are selected to facilitate the desired body motion. The body-first approaches aim to simplify the problem using established planning approaches as reported in [4], [10], [13], [14]. However, for limited footholds, a body-driven approach might fail in finding suitable footholds, given the planned body motion. Therefore, contact-driven methods are more suitable for our scenarios with limited footholds.

We can find many approaches using "terrain aware" legged locomotion in the literature, such as [1], [15]–[18]. The techniques include foothold evaluation and pose optimization to achieve robust locomotion in challenging terrain. However, they can be considered short-horizon planning methods because they plan only a few steps ahead and use regular gait patterns. Although robust motion over rough terrain can be achieved, they are not suitable for limited foothold scenarios, where they will likely fail.

The focus of the presented work is on the contact-driven approaches that operate with discrete stances of the robot. They are called *gait-free* methods because they are not constrained by a specific gait pattern. Instead, a graph (state-space) representation models feasible steps from one stance to another. Then, existing state-space search techniques can find a suitable sequence of steps.

A graph search with vertices representing support polygons (stances) is presented in [6], where the authors exploit a fixed gait. A fine-tuned cost function penalizes edges that will likely lead to a plan failure in the verification phase. Despite no formal guarantee of the plan's feasibility, the reported planning times are over a minute for moderately rough terrains. However, the approach would fail in scenarios requiring robot body adjustment and irregular alternation of the individual leg movements of a hexapod walking robot.

The most extensive work (to the best of the authors' knowledge) on the contact-driven planning for legged robots is covered by [2], [7], [12] following on [19], where the legged locomotion is studied in a broad context of multi-modal planning for systems whose properties change discretely. Multi-modal planning includes picking up and placing objects with a manipulator and walking robots lifting and placing their legs. Sampling the submanifolds and their intersection is equivalent to solving sets of non-linear equations. In particular, the authors employ a combination of randomized sampling and projection using the Newton-Raphson method. Therein proposed general approach is probabilistically complete, and the reported planning times are in the order of minutes to tens of minutes in challenging rough terrain scenarios, which is considered promising but still relatively too demanding for online deployments.

Planning a sequence of contact points without a predefined set of footholds is presented in [8]. The method uses a rough path of the robot's body to construct a potential field heuristic to guide the search and avoid getting stuck in local minima caused by obstacles. Footholds are generated dynamically with the reported planning times in tens of minutes to hours.

A heuristic based on discrete leads for planning in a space composed of mode families representing modes parametrized by a continuous parameter is proposed in [20]. The reported planning times for a simplified model of a two-limbed robot in 2D are in the order of minutes to tens of minutes. Thus, similarly to [8], the reported times are considered too high for our motivational online deployment.

A step planning framework focused on the implementation modularity has been recently presented in [21]. It provides efficient usage of multi-threading and adaptive constraint evaluation. It can accommodate gait-free planning but needs to improve the informativeness of state-space search.

Several existing motion planning solvers address the studied gait-free contact-driven planning; however, the computational times reported so far are in minutes or tens of minutes, which is considered unsuitable for online deployment. The most promising methods are based on multi-modal planning [2], [8], [20], [21] that still needs to address high computational requirements. Two possible sources for improvements can be identified. The first is to support the state-space search with a suitable heuristic to find plausible sequences further validated by feasible motion planning efficiently. The second is a fast constraints validation, which requires solving complex non-linear problems as seen in [2], [11], [19], [22]. In the present work, we focus on finding suitable heuristic while exploiting existing formulation for constraints evaluation.

III. Problem Statement

The walking robot movements can be described as repetitive acquiring and releasing contact of the legs with the environment at the footholds. Let us denote a set of the robot's effectors (feet) $\mathcal{E}$ and a set of footholds $\mathcal{H} \subset \mathbb{R}^3$. A partial assignment of the robot's feets $\mathcal{E}$ to footholds $\mathcal{H}$ is called a *stance* [12] that is further denoted $\sigma : \mathcal{E} \to \mathcal{H}$. Notice that some feet might be free in a stance. For a formal problem definition, let us denote Σ as the set of all possible assignments (stances), although it might be practical to enumerate only some of them.

In addition to assigning feet to footholds, each stance entails a set of constraints defining a submanifold $\mathcal{F}_\sigma \subset \mathcal{C}$ of the robot configuration space $\mathcal{C}$. For the studied motion planning, we consider the following constraints.

- *Kinematic reachability* constraint requests that the feet of the robot must be at the assigned footholds.

- *Stability* constraint is to compensate gravity through forces acting at the footholds.
- *Self-collision* constraint ensures the robot's parts do not mutually intersect.
- *Terrain-collision* constraint ensures the robot parts do not intersect with the environment.

The constraints can be expressed for the robot configuration $q \in \mathcal{C}$ using multivariate functions $f_\sigma(q)$ and $g_\sigma(q)$:

$$\mathcal{F}_\sigma = \{q \,:\, f_\sigma(q) = \mathbf{0} \;\wedge\; g_\sigma(q) \geq \mathbf{0} \text{ for } q \in \mathcal{C}\}\,. \tag{1}$$

Then, we can express the feasible subset $\mathcal{C}_{\text{feasible}}$ of $\mathcal{C}$ as

$$\mathcal{C}_{\text{feasible}} = \bigcup_{\sigma \in \Sigma} \mathcal{F}_\sigma\,. \tag{2}$$

The addressed multi-legged robot planning task is to find a path from an initial robot configuration $q_0 \in \mathcal{C}_{\text{feasible}}$ to a goal configuration defined as a region $\mathcal{G} \subset \mathcal{C}_{\text{feasible}}$ using assignments of feet to footholds Σ. Thus, the task can be defined as a tuple $\mathcal{P} = (q_0, \mathcal{G}, \Sigma)$, where each stance $\sigma \in \Sigma$ entails the constraints imposed as of (1).

A solution to $\mathcal{P}$ is a feasible path in $\mathcal{C}_{\text{feasible}}$ supported by a sequence of stances $\mathcal{S}_{\text{contact}}$ corresponding to the assigned footholds to the feet that support the robot by their contact with the footholds. We can request a path with the minimum number of stances k because of the limited foothold availability. Therefore, a solution to $\mathcal{P}$ can be expressed as a sequence of stances $\mathcal{S}_{\text{contact}} = (\sigma_1, \ldots, \sigma_k)$, and subpaths $\Pi = (\pi_1, \ldots, \pi_k)$. The final path can then be formed from the concatenation of the subpaths defining the robot's motion from a stance σ_i to the following stance σ_{i+1}. Finding a solution to $\mathcal{P}$ can be thus addressed as the optimization problem to find a feasible path with the minimal number of stances, formally expressed as Problem 3.1.

Problem 3.1 (Gait-Free Planning $\mathcal{P} = (q_0, \mathcal{G}, \Sigma)$):

$$\mathcal{S}^*_{\text{contact}}, \Pi^* = \operatorname{argmin}_{\mathcal{S}_{\text{contact}}, \Pi} k \tag{3a}$$

s.t.

$$\mathcal{S}_{\text{contact}} = (\sigma_1, \ldots, \sigma_k)\,, \tag{3b}$$
$$\Pi = (\pi_1, \ldots, \pi_k)\,, \tag{3c}$$
$$q_o \in \mathcal{F}_{\sigma_1}\,, \quad \mathcal{F}_{\sigma_k} \cap \mathcal{G} \neq \emptyset\,, \tag{3d}$$
$$\mathcal{F}_{\sigma_i} \cap \mathcal{F}_{\sigma_{i-1}} \neq \emptyset \text{ for } 2 \leq i \leq k\,, \tag{3e}$$
$$\pi_i : [0,1] \to \mathcal{F}_{\sigma_i} \text{ for } 1 \leq i \leq k\,, \tag{3f}$$
$$\pi_i(0) = \pi_{i-1}(1) \text{ for } 2 \leq i \leq k\,, \tag{3g}$$
$$\pi_1(0) = q_0 \text{ and } \pi_k(1) \in \mathcal{G}\,. \tag{3h}$$

The solution feasibility is ensured by (3d) and connectivity of subpaths by (3e–3g). The solution's initial and final configurations are defined by (3h). The following assumptions with the associated justifications are made to address Problem 3.1.

- *Quasi-static stability* is a sensible restriction for scenarios with deliberate motion planning and required safety.
- *Finite set of footholds* with relatively a few suitable footholds arising from motivational deployment scenarios. However, we can further assume that even on rough terrains, there are locally optimal safe footholds such as small concavities or centers of flat patches [13].
- *Rigid environment* is considered for the sake of simplicity. In real-world scenarios, avoiding compliant parts of the terrain can be desirable. Therefore, footholds on the compliant terrain can be excluded from the set of footholds. Moreover, additional constraints can be added into g_σ to limit forces acting on compliant footholds preventing deformations.
- *Continuous submanifold* $\mathcal{F}_\sigma$ assumption allows searching a single connected set of feasible configurations within the stance. With that assumption, we can exploit an intermediate configuration in the intersection of two consecutive stances $\mathcal{F}_{\sigma_i} \cap \mathcal{F}_{\sigma_{i+1}}$. There exists a feasible path from any configuration of $\mathcal{F}_{\sigma_i}$ to any configuration of $\mathcal{F}_{\sigma_{i+1}}$ via the intermediate configuration $q^{\sigma_i}_{\sigma_{i-1}} = \pi_{i-1}(1) = \pi_i(0)$ for $2 \leq i \leq k$.
  We can further assume that the body workspace of the multi-legged robot is convex [23]. A path within a single feasible connected set mostly moves the body, lifting and placing the legs. Thus, if the robot's workspace is not significantly interfering with obstacles, $\mathcal{F}_{\sigma_i}$ is likely convex or close to convex, which can help search for the optimal subpaths within the stances.

IV. Proposed Gait-Free Planning Method

The proposed gait-free planning follows the contact-driven approach [7] with two steps: finding a sequence $\mathcal{S}_{\text{contact}}$ and sequence validating, which is the determination of the paths connecting stances in the sequence satisfying the constraints (3d–3h) with feasibility constraints (1). Since we assume continuous $\mathcal{F}_\sigma$ (Section III), finding a valid contact sequence guarantees the existence of the paths. Even for the cases where the assumption does not hold, the robot motion is the most constrained at the *intermediate configurations* laying in the intersections of $\mathcal{F}_{\sigma_i} \cap \mathcal{F}_{\sigma_{i+1}}$ and a feasible contact sequence is likely to be validated in practice [12]. Therefore, we are focused on finding $\mathcal{S}_{\text{contact}}$ using the A* search algorithm, and we propose a new search heuristic improving A* performance that is detailed in Section IV-A. Its usage in the search algorithm is summarized in Section IV-B, and our solution to constraints satisfaction is presented in Section IV-C.

A. Proposed "Caterpillar" Search Heuristic

The proposed search heuristic guides A* toward promising stances. It is a function $h(\sigma, \mathcal{G})$ of the expected cost from the stance σ to the goal region $\mathcal{G}$ given as a desired (x, y) position of the robot's base link. The heuristic h is designed to avoid local minima that can be observed for existing heuristics. It reflects that the robot body moves backward when the front legs need to be lifted for the next step. In contrast, such steps are always penalized by the *Support polygon* heuristic [6], [12] computed as the weighted Euclidean distance of the centroid of the support polygon to the goal region. It is because the centroid of the support polygon always moves backward when lifting the front legs. Thus, the search would avoid expansion of the states with front legs lift and prefer branches of the search tree that start with rear leg lift even for the states the front leg lift is desirable. Therefore, we take inspiration from

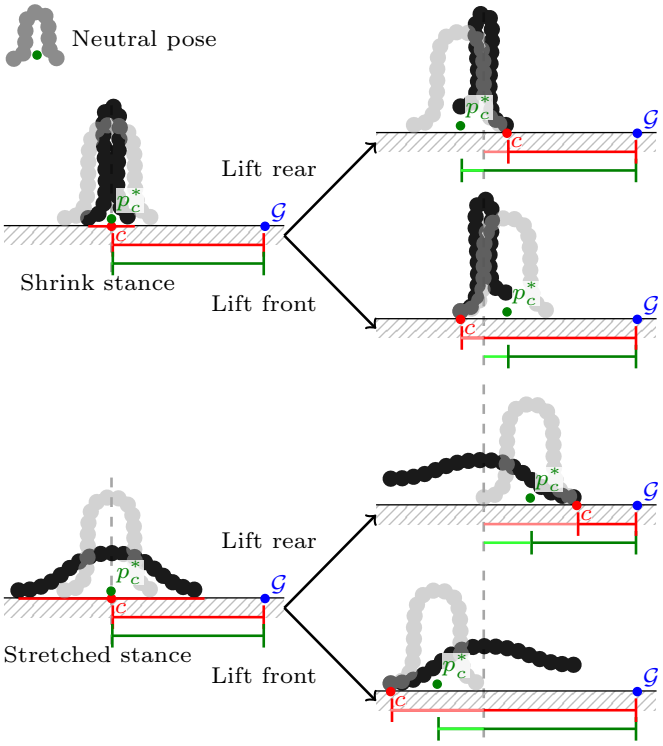


Fig. 2. Schematic principle of the proposed Caterpillar heuristic. The neutral pose is to minimize the distances to footholds in the least-squares sense, and it is depicted in gray for each scenario with its position $\mathbf{p}_\sigma^*$ marked in green. Notice that the centroid c of the support polygon (red) always moves front when lifting a rear leg and backward when lifting the front leg. For a superposition of the neutral pose, we can further observe $\mathbf{p}_\sigma^*$ moves backward when lifting the rear leg in the shrink stance while moving forward for the front leg lift. Similarly, for the stretched stance, $\mathbf{p}_\sigma^*$ moves forward when lifting the rear leg and backward for the front leg lift.

the caterpillar gait and propose *Caterpillar* heuristic to address the drawback.

The Caterpillar heuristic design originates from observing a caterpillar walking while shrinking and stretching its body as schematically visualized in one dimension in Fig. 2. It is computed using a reference point $\mathbf{p}_\sigma^*$ instead of the support polygon center. A configuration $(\mathbf{p}_\sigma^*, \boldsymbol{\theta}_0)$ is considered in which the *Controllable Degrees of Freedom* (CDoFs) are set to the "*neutral pose*" denoted $\boldsymbol{\theta}_0$ corresponding to the case the robot is standing still on flat terrain. The body position $\mathbf{p}_\sigma^*$ minimizes the distance of the feet to footholds mapped by σ in the least-squares sense to minimize the contact constraint

$$\mathbf{p}_\sigma^* = \underset{\mathbf{p}\in\mathrm{SE}(3)}{\operatorname{argmin}} \left\| f_{\mathrm{contact}_\sigma}((\mathbf{p}, \boldsymbol{\theta}_0)) \right\|_2 , \tag{4}$$

which is a *least-squares rigid motion* problem efficiently solved using singular value decomposition [24]. The heuristic value is the distance between $\mathbf{p}_\sigma^*$ and the goal region $\mathcal{G}$ that is scaled by α_h

$$h(\sigma, \mathcal{G}) = \alpha_h \left\| \mathbf{p}_\sigma^* - \mathcal{G} \right\|_2 . \tag{5}$$

The caterpillar takes steps with its rear/front legs in the stretched/squished stance, respectively. Observe the change in the position of the centroid c of the support polygon (reduced to a line segment for the model in Fig. 2) and the position of the *neutral pose* $\mathbf{p}_\sigma^*$. The centroid c moves toward the goal

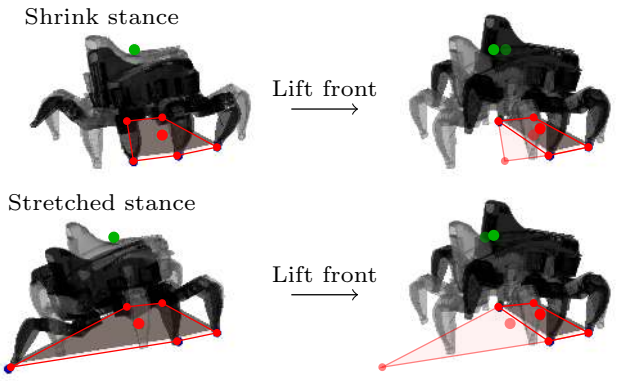


Fig. 3. The proposed Caterpillar heuristic is based on the reference point $\mathbf{p}_\sigma^*$ (a small green disk) that moves forward for a step with the front left lift contrary to the center of support polygon (a small red disk) of [12].

when the rear leg is lifted regardless of the current stance. The neutral pose $\mathbf{p}_\sigma^*$ moves such that it rewards the expected next step of the caterpillar gait. Hence, the heuristic is "aware" of the motion and possible further steps that can be reached by the lifted leg.

The principle of the *caterpillar* heuristic on the simplified one-dimensional model can be generalized to legged locomotion. The heuristic effect on the hexapod walking robot is illustrated in Fig. 3, showing the benefit of using $\mathbf{p}_\sigma^*$ compared to the center of the support polygon. Although the heuristic is relatively straightforward and easy to implement, its practical impact in finding a sequence $\mathcal{S}_{\text{contact}}$ is significant, as supported by the results reported in Section V.

B. Finding Contact Sequence

A candidate sequence is found using the A* state-space search algorithm, where states are stances $\sigma \in \Sigma$. Actions at each state represent a possible transition from one stance σ to another stance σ'. The action is validated by the existence of the shared configuration in the feasible sets of the stances $\mathcal{F}_\sigma \cap \mathcal{F}_{\sigma'} \neq \emptyset$. The goal region corresponds to the set of all stances σ that intersects with the goal set $\mathcal{G}$, $\mathcal{F}_\sigma \cap \mathcal{G} \neq \emptyset$. A solution to the planning problem is $\mathcal{S}_{\text{contact}}$, a contact sequence (3b) of Problem 3.1. The search procedure is summarized in Algorithm 1 that constructs the search tree Σ', from which the final sequence is determined by a graph search and backtracking from the goal.

A possible action at a state σ is adding or removing a single foothold. For each effector e fixed by σ, we consider a stance σ' that removes the mapping of a foothold to e. On the contrary, for each effector $\hat{e}$ not fixed in σ, the considered stance σ' is the one that adds mapping of a foothold to $\hat{e}$.

The feasibility of the action is verified by finding an intermediate configuration within the intersection $\mathcal{F}_\sigma \cap \mathcal{F}_{\sigma'}$. It is performed by Algorithm 2, which attempts to find a configuration satisfying the constraints of both sets. Here, we propose to use Levenberg-Marquardt (LM) method to solve the non-linear constraints. Compared to Newton's method utilized in [12], [25], we found out the LM algorithm converges even from relatively poor starting configurations and needs only a

Algorithm 1: A^* Search with Caterpillar Heuristic

FindCandidateSequence $(\sigma_{\text{start}}, q_{\text{start}}, \mathcal{H}, \mathcal{G})$:
$\Sigma' \leftarrow \{\sigma_{\text{start}}\}$ // Init. the search tree.
$c(\sigma_{\text{start}}) \leftarrow 0$ // Init. the cost c of the root vertex.
$Q \leftarrow \{\sigma_{\text{start}}\}$ // Init. the open list (priority queue).
while $Q \neq \emptyset$ **do**
  $\sigma \leftarrow \min_{\sigma \in Q} c(\sigma) + h(\sigma, \mathcal{G})$; // Using (5).
  $Q \leftarrow Q / \{\sigma\}$
  if $\mathcal{F}_\sigma \cap \mathcal{G} \neq \emptyset$ **return** END SUCCESS
  for $\sigma' \in \text{Successors}(\sigma, \mathcal{H})$ **do**
    if $\sigma' \notin \Sigma'$ **then** $c(\sigma') \leftarrow \infty$
    if $c(\sigma') \geq c(\sigma) + 1$ **and** $\text{Intersect}(\sigma, \sigma')$
      $c(\sigma') \leftarrow c(\sigma) + 1$
      $Q \leftarrow Q \cup \{\sigma'\}$; $\Sigma' \leftarrow \Sigma' \cup \{\sigma'\}$
return END FAILURE

Successors $(\sigma, \mathcal{H})$:
$S \leftarrow \emptyset$
for $e \in \{\exists \sigma(e) ; e \in \mathcal{E}\}$ **do** $S \leftarrow S \cup \sigma|_{\mathcal{E}/e}$
for $e \in \{\nexists \sigma(e) ; e \in \mathcal{E}\}$ **do**
  $\mathbf{e}_{\text{FK}} \leftarrow$ position of e in $(\mathbf{p}^*_\sigma, \boldsymbol{\theta}_0)$ // Using (4).
  for $h \in \{\|h - \mathbf{e}_{FK}\|_2 \leq R_{search} ; h \in \mathcal{H}\}$ **do**
    $S \leftarrow S \cup \{\sigma \cup (e, h)\}$
return S

Algorithm 2: Non-empty Intersection

Intersect (σ_1, σ_2):
for $i \in \{1 \dots MaxSamples\}$ **do**
  $q \leftarrow \text{SampleNeighbourhood}(\sigma_1, \sigma_2)$
  $q \leftarrow \text{SolveLM}(q, f_{\mathcal{F}_{\sigma_1} \cap \mathcal{F}_{\sigma_2}})$ // Find q^* s.t. $f(q^*) = 0$ using LM with q as initial guess.
  if $f_{\mathcal{F}_{\sigma_1} \cap \mathcal{F}_{\sigma_2}}(q) \approx 0$ **return** END SUCCESS
return END FAILURE

SampleNeighbourhood (σ_1, σ_2):
$\sigma_{\text{contact}} \leftarrow \text{argmax}_{\sigma \in \{\sigma_1, \sigma_2\}} |\sigma|$
return $(\mathbf{p}^*_{\sigma_{\text{contact}}}, \boldsymbol{\theta}_0) + \mathcal{N}(\mathbf{0}, \mathbf{D})$

few samples to obtain a valid configuration. The initial guess for the LM algorithm is a perturbation of $(\mathbf{p}^*_{\sigma_{\text{contact}}}, \boldsymbol{\theta}_0)$ with a normally distributed noise, where σ_{contact} is the stance with the higher number of footholds. Additive damping strategy [26] is employed with $\lambda_{\text{boost}} = 1.5$ and $\lambda_{\text{drop}} = 0.1$. Algorithm 2 is also used to test whether the goal set is reached.

C. Constraints

An important part of finding a contact sequence is determining that it satisfies the constraints for the feasibility sets $\mathcal{F}_\sigma$ defined by (1) and reformulated to

$$\mathcal{F}_\sigma = \{q \ : \ f_{\mathcal{F}_\sigma}(q) = \mathbf{0}\}, \tag{6}$$

where

$$f_{\mathcal{F}_\sigma}(q) = \begin{bmatrix} f_\sigma(q) \\ \min(g_\sigma(q), \mathbf{0}) \end{bmatrix}, \tag{7a}$$

$$f_\sigma(q) = \left[f_{\text{contact}_\sigma}(q) \right], \tag{7b}$$

$$g_\sigma(q) = \begin{bmatrix} g_{\text{support}_\sigma}(q) \\ g_{\text{SDF}_\sigma}(q) \end{bmatrix}. \tag{7c}$$

The configuration space $\mathcal{C}$ of the walking robot is formed by the position and orientation of the robot and all joint values of the robot's CDoFs $\boldsymbol{\theta}$ that can be expressed as

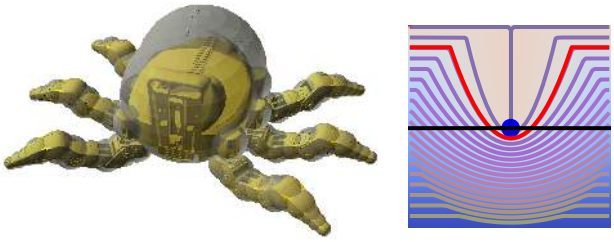
$$\mathcal{C} = \{q \in (SE(3) \times \boldsymbol{\theta})\}, \tag{8}$$

where $\boldsymbol{\theta}$ represents CDoFs and $SE(3)$ is a special Euclidean group $(x, y, z, \alpha, \beta, \gamma)$ representing the position of the robot base $x, y, z \in \mathbb{R}$, and three Tait-Bryan angles for the robot body orientation; the (x-y-z) extrinsic rotation order is used. The range of the joint angles $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{nCDoF})$ is given by the robot's construction as the minimal $\underline{\theta_i}$ and maximal $\overline{\theta_i}$ values, $\theta_i \in \left[\underline{\theta_i}, \overline{\theta_i}\right]$. Besides, the ranges of the angles are reduced to prevent self-collisions.

The utilized constraints' expression allows gradient-based solvers to satisfy the constraints. All constraints $f_{\text{contact}_\sigma}$, $g_{\text{support}_\sigma}$, and g_{SDF_σ} are expressed as continuous, differentiable functions allowing configuration projection onto the constraint manifold using numeric methods [22]. The constraints are managed as follows.

Contact constraint $f_{\text{contact}_\sigma}$ expresses the position of foot tips relative to the assigned footholds. However, the contact constraint is equality, and it is infeasible to use the standard sampling approach to obtain a configuration satisfying the constraint [22]. Therefore, the projection approach is used to obtain a configuration in the submanifold (Algorithm 2). We employ forward kinematics and Jacobian [27].

Stability constraint is reduced to requiring the projection of the robot's center of mass to the horizontal plane to be inside the convex hull of the projections of the footholds fixed by a given stance σ. The function $g_{\text{support}_\sigma}$ expresses the distances of the projection of the robot's center of mass from the sides of the support polygon.



(a) Sphere collision model (b) Collision relaxation

Fig. 4. Utilize collision model: (left) Approximation of robot's shape using spheres; (right) Relaxation of the collision field around the foothold.

Terrain collision constraint $g_{\text{SDF}_\sigma}(q)$ assessment is based on the robot's shape approximated with a set of spheres as depicted in Fig. 4a. The distance of the spheres' centers to the terrain is found using a precomputed *Signed Distance Fields* (SDF). The distance has to be longer than the corresponding sphere's radius to satisfy the constraint. The SDF provides a direction to the closest point on the terrain; therefore, it is differentiable. Both the value and gradient are retrieved in a constant time from the lookup table. The collision constraint function g_{SDF_σ} can then be expressed as

$$\tilde{g}_{\text{SDF}_\sigma}(q) = \left[M_{\text{col}} + r_i - \text{SDF}(c_i(q)) \right], \tag{9}$$

where $M_{\text{col}} \geq 0$ is the *collision margin*, r_i stands for the radius of the i-th sphere centered at $c_i(q)$ for the configuration q.

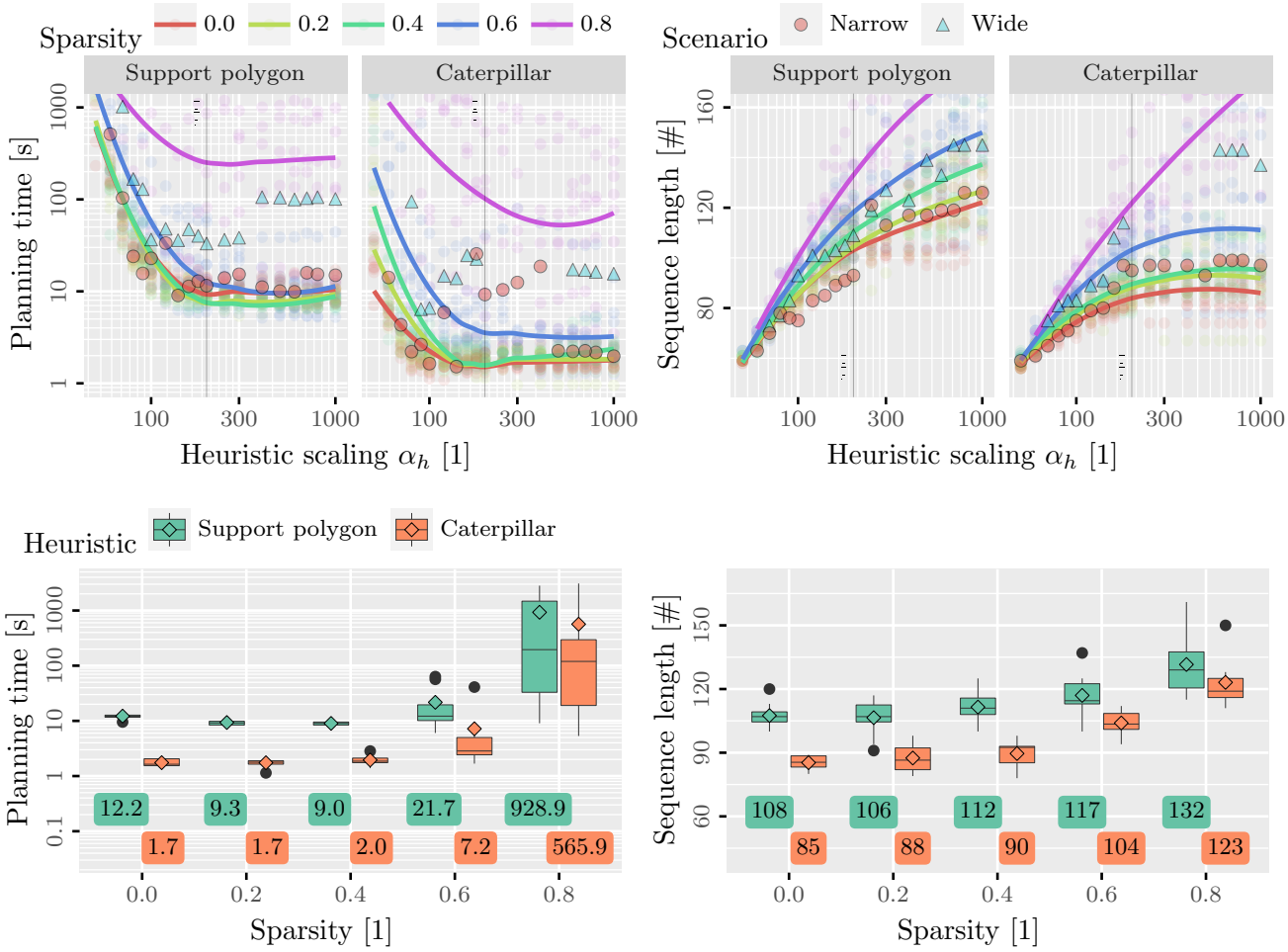


Fig. 5. Effect of Caterpillar heuristic and heuristic scaling on the planning performance. The relationship between the heuristic scaling α_h and planning time/sequence length is depicted in the top left/right plot. Results for the random scenarios with the particular sparsity level are depicted by the colored curves fitted by local polynomial regression to show the overall trend. Individual results are shown as transparent disks with the corresponding color to indicate a relatively high spread of the results caused by the random removal of the available footholds. Results for the handcrafted scenarios *narrow* and *wide* scenarios are shown as red disks and blue triangles, respectively. A detailed plot for heuristic scaling of $\alpha_h = 200$ is shown in the bottom plots. The box plot is used to hint at the distribution of the test scenarios, and an average of the results are denoted using a diamond marker and numeric label under the plot. The reported planning times correspond to the identical computational environment.

However, we need to address the contact of the feet with the terrain, which requires a distance below the collision margin. We also request to allow some penetration through the terrain to account for the footholds' compliance. Hence, the collision constraint is relaxed around the footholds of the stance by the relaxation term ρ_σ as

$$g_{\text{SDF}_\sigma}(q) = \left[M_{\text{col}} + r_i - \text{SDF}(c_i(q)) + \rho_\sigma(c_i(q))\right], \qquad (10)$$

where

$$\rho_\sigma(x) = \sum_{h \in \sigma(\mathcal{E}); \|x-h\| \leq R_{\text{relax}}} R_{\text{relax}} - \|x - h\|, \qquad (11)$$

with the radius R_{relax} in which the collision function is affected around a foothold h for the stance σ. A visualization of the deformed distance field around a foothold is shown in Fig. 4b. Outside the R_{relax}-neighborhood of the footholds, (10) is equivalent to (9).

V. Empirical Evaluation Results

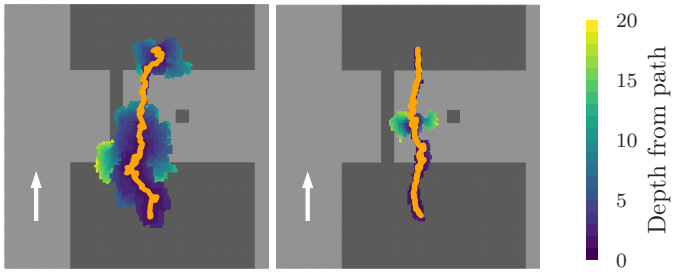
The proposed gait-free planning has been empirically evaluated with a hexapod walking robot in testing scenarios where not all legs can find a foothold in the gait-defined stance and support phases. We consider *stepping stone* with only a single foothold, see Fig. 1, which requires precise motion planning and a suitable sequence of stance and swing phases. The former *narrow* scenario is further modified to a more challenging *wide* scenario with a wider gap. Besides, we generate random scenarios with sparse footholds with $8\,\text{cm} \times 8\,\text{cm}$ large tiles, each with a single foothold in its center. The tiles are of random height uniformly distributed between $-h$ and h; however, the tiles are invalidated with the probability p_{sparse} by lowering the height to the value not reachable for the robot. Fifty scenarios were generated; ten for each of the sparsity levels $p_{\text{sparse}} \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$, see examples in Figs. 1c and 1d.

The performance of the proposed planner is studied us-

ing a nominal configuration without random perturbation in SampleNeighbourhood of Algorithm 2 to support the repeatability of the results. The identified most crucial parameter to planner performance is the heuristic scaling α_h. Therefore, we performed 1900 trials in the random step field scenarios for $50 \leq \alpha_\mathrm{h} \leq 1000$ for both the narrow and wide gap between the platforms. In addition to the proposed *Caterpillar* heuristic, we examined the *Support polygon* heuristic [12]. For each trial, the maximal computational time has been set to 2 h, which distorts the results. However, the effect of missing data is present only for low heuristic scaling and highly sparse terrains.

The results are summarized in Fig. 5. A significant spread in the results can be seen, specifically for the scenarios with high sparsity. It is caused by a random removal of the available footholds. A high sparsity scenario can be similarly difficult to low sparsity scenarios if the removed footholds are those that the planner would not use anyway. Hence, in such a case, limited available footholds have a small impact on the planner's performance.

The presented data support that the proposed Caterpillar heuristic improves performance significantly. The required planning time is sensitive to heuristic scaling up to about $\alpha_\mathrm{h} = 200$, and up to $\alpha_\mathrm{h} = 500$ for the most difficult scenario with $p_\mathrm{sparse} = 0.8$. The length of the found sequence increases logarithmically for the scenarios with $p_\mathrm{sparse} = 0.8$ and sub logarithmically for denser scenarios. The effect of scaling on sequence length diminishes for $\alpha_\mathrm{h} > 300$. Further, it is apparent from the plots that our Caterpillar heuristic significantly improves the planner's performance, leading to roughly 2–7 times faster search. It also reduces the negative effect of heuristic scaling on the sequence length.



(a) Support polygon (b) Caterpillar

Fig. 6. Visualization of the search tree with the Support polygon (left) and proposed Caterpillar (right) heuristics for the *wide* scenario (depicted in Fig. 1b) with $\alpha_\mathrm{h} = 1000$. The orange curve represents the center of the robot's body for the final found path. The vertices of the search tree are visualized as pixels overlaying the map at a distance from the final path, roughly corresponding to the depth of the vertex from the parent vertex of the final path, which is further highlighted by the vertex color. The displayed "blobs" surface corresponds to the number of expanded vertices.

The main source of the Caterpillar heuristic's improvements is demonstrated in Fig. 6, where a search tree is visualized for the Support polygon and Caterpillar heuristics. The number of expanded vertices for the Support polygon heuristic is 2862, significantly higher than only 712 expansions for the proposed Caterpillar heuristic. Moreover, the search with the Support polygon/Caterpillar heuristic yields the length of the final sequence 112 and 88, respectively. The effect of the proposed heuristic is further demonstrated in Fig. 7, showing the heuristic distance to the goal estimate along the found path. The estimate based on the centroid of the support polygon increases every couple of steps, which results in unnecessary branching of the search tree that is visualized in Fig. 6a. On the other hand, Caterpillar heuristic decreases almost monotonically.

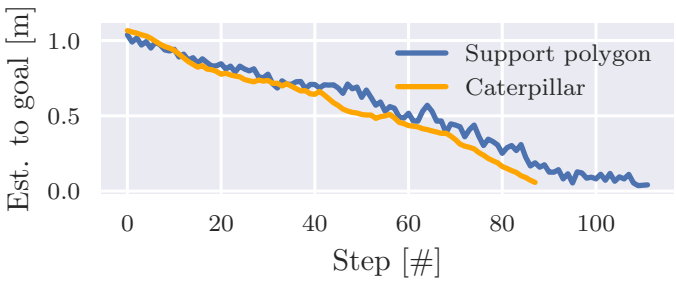


Fig. 7. Heuristic distance-to-goal estimate for the wide gap scenario with the search tree visualization in Fig. 6.

A. Real-world Deployment

The path planned by the proposed planner has been deployed on a real robot with a gap-crossing scenario depicted in Fig. 8a. The platform used is the SCARAB II [28] powered through an attached cable to reduce its weight. The robot's footprint is about 25 cm × 40 cm. It is driven by 18 ROBOTIS DYNAMIXEL XM430-W350 actuators and the Intel NUC i7. The Intel RealSense D435 and Intel RealSense T265 cameras are used for exteroception – mapping and localization, respectively. The Robot Centric Elevation Mapping [29], [30] library by ANYbotics was used for the mapping task with the Grid Map package [31] providing the used SDF-based collision checking.

(a) Planning scenario set up (b) Grid map visualization

Fig. 8. A deployment scenario, where gait-free planning is necessary to traverse the terrain with only limited footholds (left) and visualization of the internal grid map representation with generated footholds represented by green disks (right).

In the deployment, the robot is manually lifted first, and the terrain is scanned with the onboard cameras.

[1] Then, the robot is placed at the start position, and the x-y goal location is selected. The footholds on the seen flat terrain with the same height as the goal are sampled, and the K-means algorithm is used to reduce their number to the desired density of approx. 1 foothold/100cm^2; similar to the performed computational scenarios. If unfavorable sets of footholds were generated, the sampling was repeated. The step sequence is planned using Algorithm 1 with $\alpha_\mathrm{h} = 200$ in tens

[1]The deployment process is documented in an accompanying video with visualizations of planned sequences – https://youtu.be/MXHWPHYrskQ.

of seconds. The visualization of the robot's localization and the generated map is depicted in Fig. 8b. The map is used to adjust the robot's initial position to compensate for imprecision in localization and mapping. The individual sub-paths connecting intermediate configurations are planned using a path optimization method with Bézier curve parametrization of the path in the joint space, as in [32]. An open-loop positional controller then executes the plan. Even with such a raw setup, the robot successfully traversed the testing course, demonstrating the planner's viability for further practical applications.

VI. Conclusion

A gait-free motion planning method for multi-legged robots is presented. The method does not rely on a specific locomotion gait, thus enabling traverse environments with only a few footholds. The planning is based on searching for candidate sequences of stances while considering all the constraints, including collisions with the terrain. Based on the examined effects of scaling the heuristic function, we established a trade-off between the computational requirements and the quality of the found solutions in scenarios with a limited number of available footholds. The proposed Caterpillar heuristic addresses the computational demand of search-based planning and significantly improves the planner performance compared to the heuristics used in the literature. Caterpillar heuristic reduces the computational burden and decreases penalty to solution length caused by the scaling of the heuristic. These improvements allow online deployment of the developed motion planner, supported by the reported results of a real-world experiment.

References

[1] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust rough-terrain locomotion with a quadrupedal robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5761–5768.

[2] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 897–915, 2010.

[3] J. Faigl and P. Čížek, "Adaptive locomotion control of hexapod walking robot for traversing rough terrains with position feedback only," *Robotics and Autonomous Systems*, vol. 116, pp. 136–147, 2019.

[4] P. Čížek, D. Masri, and J. Faigl, "Foothold placement planning with a hexapod crawling robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4096–4101.

[5] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.

[6] P. Vernaza, M. Likhachev, S. Bhattacharya, S. Chitta, A. Kushleyev, and D. D. Lee, "Search-based planning for a legged robot over rough terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 2380–2387.

[7] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1325–1349, 2008.

[8] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 428–442, 2013.

[9] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, "Optimization and learning for rough terrain legged locomotion," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 175–191, 2011.

[10] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An efficient acyclic contact planner for multiped robots," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018.

[11] D. Belter, "Efficient modeling and evaluation of constraints in path planning for multi-legged walking robots," *IEEE Access*, vol. 7, pp. 107 845–107 862, 2019.

[12] K. Hauser, "Motion planning for legged and humanoid robots," Ph.D. dissertation, University of Illinois, 2008.

[13] D. Belter, P. Labecki, and P. Skrzypczynski, "Adaptive motion planning for autonomous rough terrain traversal with a walking robot," *Journal of Field Robotics*, vol. 33, no. 3, pp. 337–370, 2016.

[14] N. Perrin, C. Ott, J. Englsberger, O. Stasse, F. Lamiraux, and D. G. Caldwell, "Continuous legged locomotion planning," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 234–239, 2016.

[15] O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini, "MPC-based controller with terrain insight for dynamic legged locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2436–2442.

[16] S. Fahmi, V. Barasuol, D. Esteban, O. Villarreal, and C. Semini, "VITAL: Vision-based terrain-aware locomotion for legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 885–904, 2023.

[17] C. Boussema, M. J. Powell, G. Bledt, A. J. Ijspeert, P. M. Wensing, and S. Kim, "Online gait transitions and disturbance recovery for legged robots via the feasible impulse set," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1611–1618, 2019.

[18] D. Belter, J. Bednarek, H.-C. Lin, G. Xin, and M. Mistry, "Single-shot foothold selection and constraint evaluation for quadruped locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7441–7447.

[19] T. Bretl, S. Lall, J.-C. Latombe, and S. Rock, "Multi-step motion planning for free-climbing robots," in *Algorithmic Foundations of Robotics VI*. Springer, 2004, pp. 59–74.

[20] Z. Kingston, A. M. Wells, M. Moll, and L. E. Kavraki, "Informing multi-modal planning with synergistic discrete leads," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3199–3205.

[21] A. Stumpf and O. von Stryk, "A universal footstep planning methodology for continuous walking in challenging terrain applicable to different types of legged robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 420–10 427.

[22] Z. Kingston, M. Moll, and L. E. Kavraki, "Exploring implicit spaces for constrained sampling-based planning," *The International Journal of Robotics Research*, vol. 38, no. 10–11, pp. 1151–1178, 2019.

[23] V.-G. Loc, I. M. Koo, D. T. Tran, S. Park, H. Moon, and H. R. Choi, "Body workspace of quadruped walking robot and its applicability in legged locomotion," *Journal of Intelligent & Robotic Systems*, vol. 67, no. 3-4, pp. 271–284, 2012.

[24] O. Sorkine-Hornung and M. Rabinovich, "Least-squares rigid motion using svd," *Computing*, vol. 1, no. 1, pp. 1–5, 2017.

[25] Z. Kingston, M. Moll, and L. E. Kavraki, "Sampling-based methods for motion planning with constraints," *Annual review of control, robotics, and autonomous systems*, vol. 1, pp. 159–185, 2018.

[26] M. Lampton, "Damping–undamping strategies for the levenberg–marquardt nonlinear least-squares method," *Computers in Physics*, vol. 11, no. 1, pp. 110–115, 1997.

[27] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

[28] M. Forouhar, P. Čížek, and J. Faigl, "SCARAB II: A small versatile six-legged walking robot," in *5th Full-Day Workshop on Legged Robots at IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1–2.

[29] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[30] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3019–3026, 2018.

[31] 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 (Volume 1)*, A. Koubaa, Ed., 2016, ch. 5.

[32] D. Valouch and J. Faigl, "Gait-free planning for hexapod walking robot," in *European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–8.