Gait-Free Planning for Hexapod Walking Robot

David Valouch Jan Faigl

***Abstract*— This paper presents a gait-free motion planning approach for quasi-static walking of hexapod walking robots on terrains with limited available footholds. The proposed approach avoids using a prescribed gait pattern allowing an arbitrary sequence of leg swings. Furthermore, it is allowed that some legs do not need to be placed on the terrain for an extended duration. The proposed method is based on a decomposition of the motion planning into: (i) finding a candidate sequence of stances and intermediate configurations representing plausible steps using a graph-search; and (ii) connecting the intermediate configurations by feasible paths satisfying the motion constraints of the walking robot. The individual one-step paths are determined using a Bézier curve-based parametrization that seems to be sufficient for the relatively simple paths of a single step, and the low-capacity parametrization yields natural-looking motion.**

I. Introduction

Hexapod walking robots provide increased stability over their "competitors" such as quadruped Spot [1] and ANYmal [2] or biped/humanoid robots such as Atlas [3]. For an exploration of highly structured, unsafe environments, such as collapsed buildings, where fast dynamic motions are undesirable, hexapod crawlers might be superior due to their inherent stability and redundancy. Hexapod walking robots can locomote using a fast, quasi-statically stable gait for which only two steps are required to complete one gait cycle, compared to four steps of a quadruped. Hexapod crawlers also do not need to use all legs for statically stable locomotion compared to quadrupeds; indeed, six is the least even number of legs that allow gait-free quasi-static movement.

In the paper, we present our results on gait-free planning that builds on the method of K. Hauser [4]. Further motivated by the relatively recently proposed robust rough-terrain locomotion [5], we propose to evaluate the terrain-collision constraint using a *Signed Distance Field* (SDF) based on Distance Transformation [6] instead of a mesh interference detection as K. Hauser. We also use an optimization-based approach motivated by [5], [7] for trajectory planning, where we proposed to employ a Bézier curve parametrization instead of computationally demanding sampling-based search for planning the single-step motions. Based on the achieved results, the proposed approach seems viable and promising to enable the gait-free motion of hexapod walking robots.

The remainder of the paper is as follows. An overview of related work is presented in the following section. The addressed problem is formulated in Section III together with the introduction of the used notation. The proposed method is presented in Section IV. A description of the empirical evaluation and report on the achieved results are presented in Section V. Finally, the concluding comments are in Section VI.

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

This work has been supported by the Czech Science Foundation (GAČR) under research project No. 21-33041J.

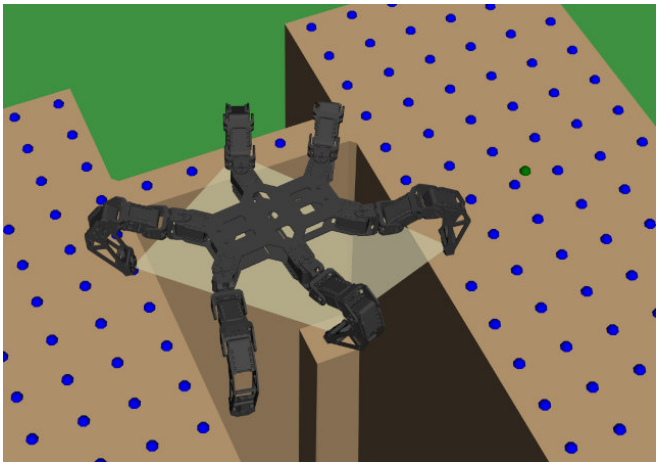


Fig. 1. Hexapod walking robot traversing an environment with missing footholds. The support polygon is highlighted.

II. Related Work

Existing approaches for motion planning with hexapod walking robots in rough terrains consider motion between configurations with six ground contacts using a "semi-regular gait" such as [8]–[10] to list a few here. It is because full-motion planning without the prescribed sequence of leg swing/stance phases is computationally very demanding; hence, the full-motion capabilities of hexapod walking robots in rough terrains are not fully exploited. In [8], RRT-based [11] motion planner is employed for two-dimensional body position and heading that are sampled, and valid leg footholds are determined and assigned for the individual stance configuration. Image processing techniques are employed in [9] to rate possible body positions in two dimensions for the fixed robot height and orientation. Then, a roadmap [12] is constructed for planning a robot path in real-time.

The authors of [10] define a concept of "weak collision freeness" to capture the notion that the workspace of the robot's legs must contain part of the terrain. The concept ensures the object (workspace) contains at least some "free space" (terrain). In [13], [14], a similarly relaxed approach is employed to ensure contact with the terrain. An "acyclic" contact planner is used in [14] for humanoid and quadruped robots, where a rough path of the body is planned first, and

the contacts are selected in the second stage. Less related to our quasi-static setting but notable approach is [13] with planning dynamic motions for a quadruped robot.

The listed methods share a unifying assumption that it is possible to plan only the motion of the robot's body. It is assumed that the ground is always reachable for all legs in some capacity. Besides, it is assumed that the footholds for the individual legs and the required leg-swing motions will be found *ad Hoc* in a way that all motion constraints would be satisfied. Even though such approaches exhibited a wide range of practical applications, there can still be challenging problems where it is necessary to precisely plan the motion of the robot legs in an arbitrary sequence of individual leg swings. Therefore, we are interested in gait-free planning to better exploit walking robots' motion capabilities, despite being computationally demanding because of the complexity of the precise motion planning in high-dimensional configuration space.

Regarding the quasi-static gait-free planning for hexapod robots, to the best of the authors' knowledge, the only work has been done by T. Bretl *et al.* [15] and K. Hauser *et al.* [16]. Besides, related research has been proposed by P. Vernaza *et al.* [17] on a quadruped robot, where, however, the planner is using a regular gait that can be "bypassed" by allowing a leg to step on a sample place, which exhibited that any (quasi-static) "gait-free" motion of a quadruped robot can be represented using a regular gait cycle.

All the approaches [15]–[17] use a graph to represent the discrete components (*modes*) of the configuration space of a walking robot caused by the contacts with the footholds on the terrain. A similar graph representation is successfully used with humanoid (biped) robots, e.g., reported in [18], albeit with a regular gait. In [19], *mode families* induced by contact surfaces representing flat surfaces of the environment are considered when planning a sequence of contacts for a humanoid robot in challenging environments. More recently, similar *mode families* are represented using a combination of discrete graph and real-valued parameters in [20], where continuous sets of "footholds" result from a pick-and-place domain that represents possible handrails for grasping.

In this paper, we follow the work of Bretl and Hauser on quasi-static motion and gait-free planning for hexapod walking robots. The proposed approach is based on the decomposition of the complex planning problem into determining candidate sequence of robot stances followed by planning feasible paths between the stances. The problem is more formally defines in the following section.

III. Problem Statement

In the studied gait-free planning for hexapod walking robot, we consider the three assumptions with the justifications as follows.

- **Quasi-static motion** – Superior stability is the main advantage of the hexapod platform. Therefore, we are targeting gait-free but quasi-static motion, and thus we assume the support polygon is formed by at least three footholds. Besides, the motion is assumed to be deliberate, such that any dynamic forces are negligible.
- **Rigid environment** – In scenarios such as exploring collapsed buildings, disturbing the environment is undesirable; therefore, any deformable terrain should be avoided. However, we consider a model of the non-rigid environment would be possible to include later. It is expected to restrict possible footholds further and put additional requirements on motion planning between the stances in examining the feasibility of the particular sequence of stances [21].
- **Finite set of footholds** – Locally optimal footholds such as small concavities might be desirable in a highly structured environment without smooth surfaces. In general, increasing the number of possible footholds would increase computational burden; however, due to the physical dimensions of the foot tips, the footholds cannot be selected completely arbitrarily. Besides, we believe footholds set might be locally optimized within the proposed planning framework, which is considered to be out of the scope of this paper and is considered for future work. Thus, foothold selection is out of the scope of this paper and the footholds on a grid are considered (see Fig. 1). An informed foothold selection is done similarly as in [22].

The motion planning is performed within the configuration space of the considered hexapod walking robot, defined as

$$\mathcal{C} = SE(3) \times SO(2)^{6N} \tag{1}$$

where N is the number of Degree of Freedom (DOF) per leg. In our practical deployments of the proposed method, we use $N = 3$.

Let a *stance* σ be defined as a partial mapping of the robots foot tips to footholds. Let the set of all stances be denoted Σ; since the set of footholds is finite, Σ is finite. Each stance σ defines a constraint function $f_\sigma : \mathcal{C} \to \mathbb{R}^n$ and a subset of the configuration space $\mathcal{F}_\sigma \subseteq \mathcal{C}$,

$$\mathcal{F}_\sigma = \{q \mid q \in \mathcal{C} \wedge f_\sigma(q) = \mathbf{0}\} \ . \tag{2}$$

In our case, the constraint function f_σ consists of the four sub-functions as

$$f_\sigma(q) = \begin{bmatrix} \mathbf{d}_\sigma(q) \\ \mathbf{p}_\sigma(q) \\ \mathbf{l}(q) \\ \mathbf{c}_\sigma(q) \end{bmatrix} .$$

The constraints considered are as follows.

- **Kinematic contact constraint** $\mathbf{d}_\sigma(q)$ describes the position of the robot's foot tips relative to the footholds assigned to them by σ.
- **Support polygon constraint** $\mathbf{p}_\sigma(q)$ defines that the center of mass has to be inside of the convex hull defined by the footholds of σ. The convex support polygon is defined by a set of affine inequalities $\tilde{\mathbf{p}}_\sigma(q) \leq \mathbf{0}$. We use $\tilde{\mathbf{p}}_\sigma$ to produce the equality constraint in our paradigm by setting

$$[\ \mathbf{p}_{\sigma,i}(q)\ ] = \left[\ \max\left(0, \tilde{\mathbf{p}}_{\sigma,i}(q)\right)\ \right] . \tag{3}$$

- **Joint limits** $\mathbf{l}(q)$ specify the joint angles must be within their respective ranges. The ranges are also set to prevent collisions between neighbouring legs of the robot. It is a trivial inequality constraint that is transformed to an equality constraint in the same way as in (3).
- **Collision constraint** $\mathbf{c}_\sigma(q)$ defines that no part of the robot may intersect the environment. This constraint is not as straightforward to define as the previous ones, and our approach is further discussed in Section IV-C.

The valid subset of the configuration space $\mathcal{C}_{\text{valid}} \subseteq \mathcal{C}$ is defined as

$$\mathcal{C}_{\text{valid}} = \bigcup_{\sigma \in \Sigma} \mathcal{F}_\sigma \,. \tag{4}$$

Let us have a planning task $\mathcal{P} = (\mathcal{C}_{\text{valid}}, q_{\text{start}}, \mathcal{G})$ with $\mathcal{G}$ being the set of goal configurations. A sequence

$$\begin{aligned} \mathcal{S} = \quad & q_0 \ \sigma_0 \ q_1 \ \sigma_1 \ q_2 \ \cdots \ q_n \ \sigma_n \ q_{n+1} \\ & q_{\text{start}} \in \mathcal{F}_{\sigma_0}, q_0 = q_{\text{start}}, \\ & q_i \in \mathcal{F}_{\sigma_i} \cap \mathcal{F}_{\sigma_{i+1}}, 1 \leq i \leq n \\ & q_{n+1} \in \mathcal{G}\,, q_{n+1} \in \mathcal{F}_{\sigma_n} \end{aligned} \tag{5}$$

is called a *candidate sequence* of $\mathcal{P}$. The structure of $\mathcal{C}_{\text{valid}}$ implies that each valid path $\pi : [0,1] \to \mathcal{C}_{\text{valid}}$ is associated with such a sequence of stances $(\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_n)$ corresponding to the sequence of sets $(\mathcal{F}_{\sigma 0}, \mathcal{F}_{\sigma 1}, \mathcal{F}_{\sigma 2}, \ldots, \mathcal{F}_{\sigma n})$, and configurations $(q_1, q_2, \ldots, q_n)$, where $q_i \in \mathcal{F}_{\sigma i-1} \cap \mathcal{F}_{\sigma i}$ are further called *intermediate configurations*. It follows that an existence of a candidate sequence of the planning task $\mathcal{P}$ is a necessary condition for the existence of π with $\pi(0) = q_0$ and $\pi(1) = q_{n+1}$. The full solution from the candidate sequence is a path π consisting of subpaths $\pi_0, \pi_1, \pi_2, \ldots \pi_n$ such that

$$\begin{aligned} \pi_i : \quad & [0,1] \to \mathcal{F}_{\sigma_i} \\ & \pi_i(0) = q_i \\ & \pi_i(1) = q_{i+1} \end{aligned} \quad . \tag{6}$$

Finally, the planning problem can be expressed as an optimization task to determine a solution $\mathcal{P}$ of the minimal cost; however, it can be expected that even finding a feasible solution would be computationally demanding. Therefore, in this paper, we focus on finding a feasible solution satisfying all the considered constraints.

IV. Proposed Gait-Free Planning Method

The proposed gait-free planning method is based on a decomposition of the planning into two parts. First, a candidate sequence of stances is found with intermediate configurations. In the second part, the intermediate configurations are connected to a smooth path that satisfies the requested motion constraints. This decomposition is motivated by the high computational cost of finding the smooth motions that would be too demanding in a solution of the sequencing part.

The existence of a candidate sequence is an indication that a full path can most likely be found [4]. The stance is a partial assignment of the robot's legs to discrete footholds on the terrain. The assignment is not defined by a gait, such as a tripod gait with three legs in the support phase and three legs in the swing phase. The gait-free motion is realized by examining possible assignments that fulfill the desired motion and satisfy the robot's motion constraints. The proposed method is detailed in the following parts of this section.

A. Candidate Sequence

The candidate sequence of the planning task is found using an approximation of the structure of $\mathcal{C}_{\text{valid}}$ by a *stance graph* $G = (\Sigma, E)$, where

$$E = \{(\sigma_1, \sigma_2) \mid \mathcal{F}_{\sigma_1} \cap \mathcal{F}_{\sigma_2} \neq \emptyset \wedge |\sigma_1| = |\sigma_2| \pm 1\}\,. \tag{7}$$

We consider that σ_2 only adds (or removes) a single foothold to σ_1, and there is a configuration in which both stances are achievable. Limiting ourselves to adding/removing a single foothold does not sacrifice generality because any walking motion can be described by a sequence of stances with such a property.

The stance graph can be searched using any graph search algorithm like A*. However, during the search, the existence of edges in accordance with (7) needs to be validated for the feasibility of transition between the stances. Hence, we need to determine a configuration in the intersection of $\mathcal{F}_{\sigma_1}$ and $\mathcal{F}_{\sigma_2}$ to validate edge (σ_1, σ_2) [4]. The existence of an edge (σ_1, σ_2) is equivalent to existence of such an intermediate configuration that is a part of both $\mathcal{F}_{\sigma_1}$ and $\mathcal{F}_{\sigma_2}$. Note that the configuration space is most constrained around these intermediate configurations as they need to satisfy the constraints of both stances.

Finding intermediate configurations is the narrowest bottleneck of the planning procedure. We follow [4] and sample configurations close to $\mathcal{F}_{\sigma_1} \cap \mathcal{F}_{\sigma_2}$ and project them into the subset using the Newton-Raphson method, i.e., solving the related set of equations $f_{\sigma 1}(q) = \mathbf{0}$, $f_{\sigma 2}(q) = \mathbf{0}$. More specifically, the "close" configurations are configurations normally distributed around a nominal configuration of the robot shifted and rotated in the way to minimize the sum of square distances of the foot tips to their footholds assigned by a target stance ($\|\mathbf{d}_\sigma(q)\|_2$).

The sampling procedure is summarized in Algorithm 1. The procedure is also employed to test if $\mathcal{F}_\sigma$ contains a goal configuration $g \in \mathcal{G}$. The constraint function is extended by the condition of the goal set $\mathcal{G}$. An example is an equality constraint on the position and orientation of the robot's body.

For simplicity, but without the loss of generality, we consider the cost of the edge in the search graph to be 1. Then, the distance of the centroid of the footholds to the goal scaled by a factor γ is used as the heuristic for A*. If γ is the inverse value of the longest possible shift of the centroid, the heuristic is admissible. However, an admissible heuristic is not necessary for finding a feasible, not necessarily optimal, solution. Thus, the value of γ can be used to tune the optimality–search-time trade-off. An example of the candidate sequence with intermediate configurations is depicted in Fig. 4.

Algorithm 1: Sample Intermediate Configuration

SampleIntermediate(σ_1, σ_2)
1: $\sigma_{\text{target}} \leftarrow \text{argmax}_{\sigma \in \{\sigma_1, \sigma_2\}} |\sigma|$
2: $\sigma_{\text{suppport}} \leftarrow \text{argmin}_{\sigma \in \{\sigma_1, \sigma_2\}} |\sigma|$
3: **for** $i \in \{1 \dots \text{MaxIteration}\}$ **do**
4: $q \leftarrow$ SampleNeighbourhood(σ_1, σ_2)
5: $q \leftarrow$ SolveNewtonRaphson(q, $f_{\mathcal{F}_{\sigma\,\text{support}} \cap \mathcal{F}_{\sigma\,\text{target}}}$)
6: **if** $f_{\mathcal{F}_{\sigma\,\text{support}} \cap \mathcal{F}_{\sigma\,\text{target}}}(q) \leq \epsilon$ **then**
7: RETURN q
8: **end if**
9: **end for**
10: END failure

SolveNewtonRaphson(q, f)
1: **for** $i \in \{1 \dots \text{MaxIterationNR}\}$ **do**
2: $\mathbf{v} \leftarrow f(q)$
3: $\boldsymbol{\delta} \leftarrow {\mathbf{J}_f}^{\dagger} \cdot \mathbf{v}$
4: $\alpha \leftarrow 1$
5: **while** $\|f(q - \alpha\boldsymbol{\delta})\|_2 > f(q)$ **do**
6: $\alpha \leftarrow \alpha/2$
7: **end while**
8: $q \leftarrow q - \alpha\boldsymbol{\delta}$
9: **if** $f(q) \leq \epsilon$ **then**
10: RETURN q
11: **end if**
12: **end for**
13: END failure

SampleNeighbourhood(σ_1, σ_2)
1: $\sigma_{\text{target}} \leftarrow \text{argmax}_{\sigma \in \{\sigma_1, \sigma_2\}} |\sigma|$
2: $\boldsymbol{\theta}_0 \leftarrow$ NominalJointPositions
3: $\mathbf{p}_0 \leftarrow \text{argmin}_{p \in SE(3)} \left\|\mathbf{d}_{\sigma_{\text{target}}}((p, \boldsymbol{\theta}_0))\right\|_2$
4: $\boldsymbol{\nu} \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{W})$
5: RETURN $(\mathbf{p}_0, \boldsymbol{\theta}_0) + \boldsymbol{\nu}$

B. Connecting Intermediate Configurations

Having a candidate sequence, a smooth path is being determined to validate the sequence; or the sequence is abandoned if a feasible path is not found. Given the intermediate configurations $q_i \in \mathcal{F}_{\sigma_{i-1}} \cap \mathcal{F}_{\sigma_i}$ and $q_{i+1} \in \mathcal{F}_{\sigma_i} \cap \mathcal{F}_{\sigma_{i+1}}$, a smooth path between them has to satisfy the constraint function of $\mathcal{F}_{\sigma_i}$.

Finding a smooth path can be formulated as a continuous optimization problem, and we propose parameterizing the path $\pi : [0, 1] \rightarrow \mathcal{C}_{\text{valid}}$ as the Bézier curve in the configuration space. The Bézier curve of the degree d is defined by $d + 1$ *control points* $\mathbf{Q} = [q_{c,0}, q_{c,1}, \dots q_{c,d}]$ as

$$\pi_{\mathbf{Q}}(t) = \sum_{i=0}^{d} \binom{d}{i} (1-t)^{d-i} t^i q_{c,i}. \tag{8}$$

The degree of the curve can be used to control the path complexity. Here, we assume that a low degree can avoid unnatural motion of the robot, similarly to how a low capacity model can help avoid overfitting in machine learning. Therefore, we first attempt to construct a path using a low degree, the degree of three is sufficient for the simplest motions, and restart the procedure with more control points if it fails, but up to the defined maximum degree.

The path constraint function is defined using the constraint function of $\mathcal{F}_{\sigma_0}$ as

$$f_{\pi_{\mathbf{Q}}, \sigma}(\mathbf{Q}) = \max_t \|f_{\sigma,0}(\pi_{\mathbf{Q}}(t))\|_2 \tag{9}$$

where the control points are initialized to a linear interpolation of q_i and q_{i+1} such that $q_{c,0} = q_i$ and $q_{c,d} = q_{i+1}$. The position of the control points $q_{c,1}, \dots q_{c,d-1}$ are adjusted by using the Newton-Raphson method to solve $f_{\pi,\sigma}(\mathbf{Q}) = 0$.

Algorithm 2: Find Path

FindPath(σ,q_1,q_2, d)
1: $\mathbf{Q} \leftarrow \left[q_1 + \frac{i}{d}(q_2 - q_1) \mid \text{for } i \in \{0, \dots, d\}\right]$
2: **for** i in MaxIteration **do**
3: $t_{worst} \leftarrow \text{argmax}_{t \in [0,1]} \|f_{\mathcal{F}_\sigma}(\pi_{\mathbf{Q}}(t))\|_2$
4: $q_{worst} \leftarrow \pi_{\mathbf{P}}(t_{worst})$
5: $\mathbf{v} \leftarrow f(q_{worst})$
6: **if** $\|\mathbf{v}\|_2 \leq \epsilon$ **then**
7: RETURN $\mathbf{Q}$
8: **end if**
9: $\boldsymbol{\delta} \leftarrow {\mathbf{J}_f}^{\dagger} \cdot \mathbf{v}$
10: $\alpha \leftarrow 1$
11: **while** $\|f(q_{worst} - \alpha\boldsymbol{\delta})\|_2 > f(q_{worst})$ **do**
12: $\alpha \leftarrow \alpha/2$
13: **end while**
14: $\mathbf{Q} \leftarrow \mathbf{Q} - \alpha\boldsymbol{\delta} \cdot \mathbf{b}_{d,0}(t_{worst})^T / \|\mathbf{b}_{d,0}(t_{worst})\|_2$
15: **end for**
16: **if** $d =$ MaxDegree **then**
17: END failure
18: **else**
19: **FindPath**(σ, q_1, q_2, $d + 1$)
20: **end if**

In each iteration of the optimization, the curve is sampled with the desired granularity, and the value for t_{worst} with the most severe constraint violation is picked. A single iteration of Newton-Raphson method is performed for $\pi_{\mathbf{Q}}(t_{\text{worst}})$. The control points are shifted by the resulting step weighted by the coefficients of $\mathbf{b}_{d,0}(t_{\text{worst}})$, where

$$\mathbf{b}_{d,0}(t) = \begin{bmatrix} 0 \\ \binom{d}{1}(1-t)^{d-1}t^i \\ \vdots \\ \binom{d}{d-1}(1-t)t^{d-1} \\ 0 \end{bmatrix}, \tag{10}$$

and normalized such that the coefficients sum to 1. The path finding procedure is summarized in Algorithm 2.

C. Collision Constraint

We do not use mesh interference methods to detect collisions with the environment as in [4]. We consider mesh interference relatively slow, but more importantly, it does not measure how much the collision-free constraint is violated or how close we are to violating it. Furthermore, it also does

not provide any hints on how to move the robot out of the collision. Therefore, we use a method based on geometric primitive and a distance-to-terrain look-up table to avoid those shortcomings.

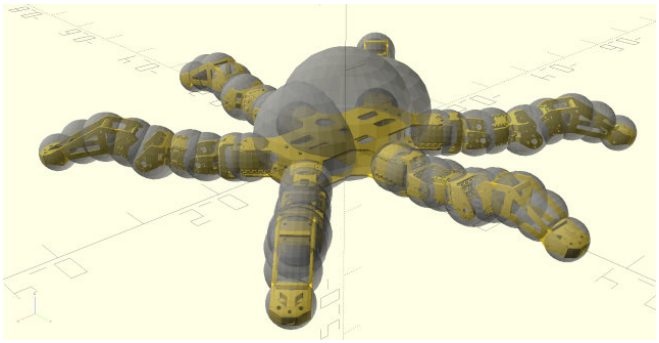


Fig. 2. Sphere-based robot collision model approximation.

The shape of the robot is approximated by enveloping its links with a small set of spheres, see Fig. 2. *Signed Distance Field* (SDF), a 3D look-up table, is obtained using the ANYbotics `grid_map` package [23]. The SDF can be efficiently computed using a "distance transform of a sampled function" [6]. Using the SDF, it is possible to compute the direction and distance to the nearest obstacle quickly. If the distance of the sphere's center to the terrain is shorter than its radius, it is in a collision. The information about the direction to the obstacle allows us to compute Jacobian of the collision constraint used in the Newton-Raphson method.

Let the robot be covered by a set of m collision spheres $(r_i, \mathbf{c}_i(q)), i \in \{1, 2, \ldots, m\}$, where r_i is the radius of the i-th sphere and $\mathbf{c}_i(q)$ is the position of its center given the configuration q. The collision margin function $\tilde{\mathbf{c}}_\sigma(q)$ is defined as

$$\tilde{\mathbf{c}}_\sigma(q) = \begin{bmatrix} \mathrm{SDF}_\sigma\left(c_1(q)\right) - r_1 \\ \mathrm{SDF}_\sigma\left(c_2(q)\right) - r_2 \\ \vdots \\ \mathrm{SDF}_\sigma\left(c_m(q)\right) - r_m \end{bmatrix} \tag{11}$$

and the collision constraint function $\mathbf{c}_\sigma(q)$ can be then defined similarly to (3) as

$$\left[\ c_{\sigma,i}(q)\ \right] = \left[\ \max\left(0, -\tilde{c}_{\sigma,i}(q)\right)\ \right]. \tag{12}$$

The function $\mathrm{SDF}_\sigma : \mathbb{R}^3 \to \mathbb{R}$ is the SDF with the collision threshold relaxed near the footholds of the stance σ. It returns a distance to the terrain reduced by the collision threshold. It gives a negative distance if the query point is inside the collision threshold. Note that when planning the step in $\mathcal{F}_{\sigma i}$ (using Algorithm 2) the collision threshold also needs to be relaxed near the footholds of σ_{i-1} and σ_{i+1}.

V. Empirical Evaluation Results

The empirical evaluation of the proposed gait-free planning for hexapod walking robots has been performed in a testing scenario motivated by [9] that is depicted in Fig. 3 for a comparison. In that setup, the robot has to cross a gap using two "beams" with "holes" in them. However, the scenario is solvable using a "terrain-aware" crawling gait.

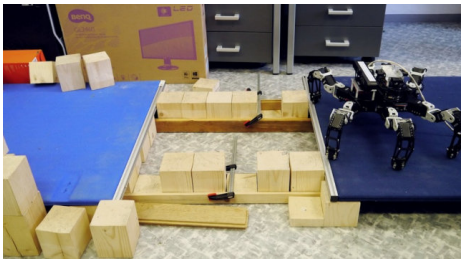
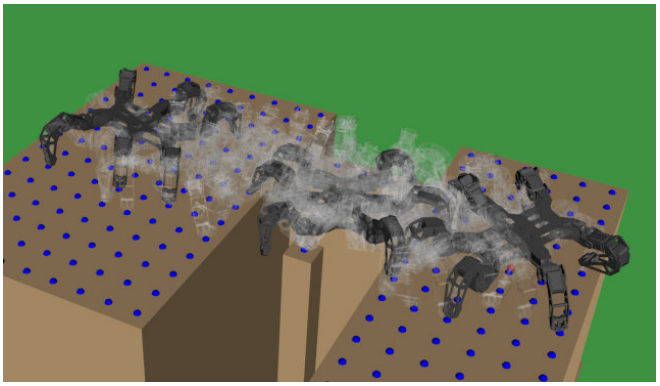
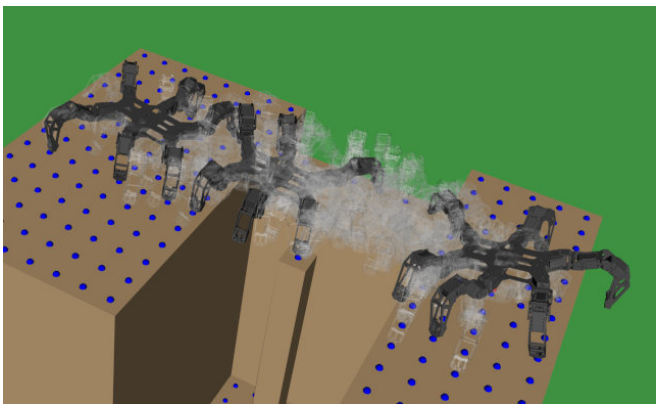


Fig. 3. The setup used in [9]

The proposed planner targets more challenging scenarios where not all legs can find a foothold in the gait-defined stance and support phases. Therefore, we create a more challenging virtual experimental setup, where we replace one of the beams with a "stepping stone" with only a single foothold. The created problem requires precise motion planning and a solution of the sequencing part to determine the suitable sequence of stance and swing phases. Furthermore, we modify the scenario to create an even more challenging planning problem with a wider gap to be passed. Thus, the proposed planner has been examined in two evaluation scenarios that are visualized in Fig. 4 with a set of footholds distributed in a grid pattern for simplicity. Besides, the performed planned motion is also shown.



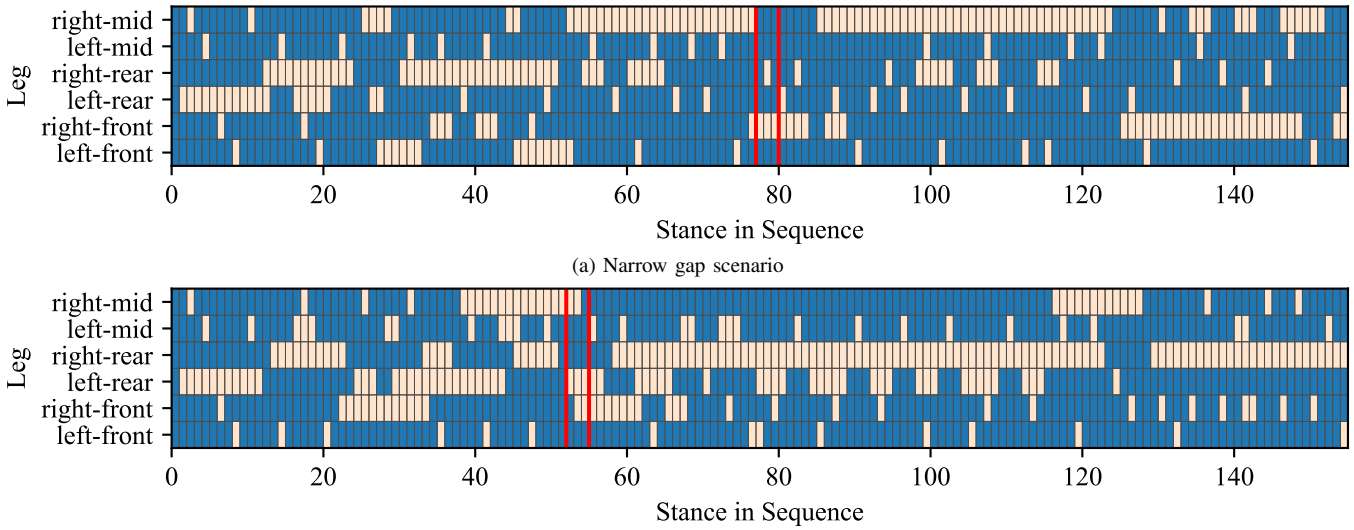
(a) Narrow gap scenario



(b) Wide gap scenario

Fig. 4. Visualization of the evaluation scenarios and the respective candidate sequences.

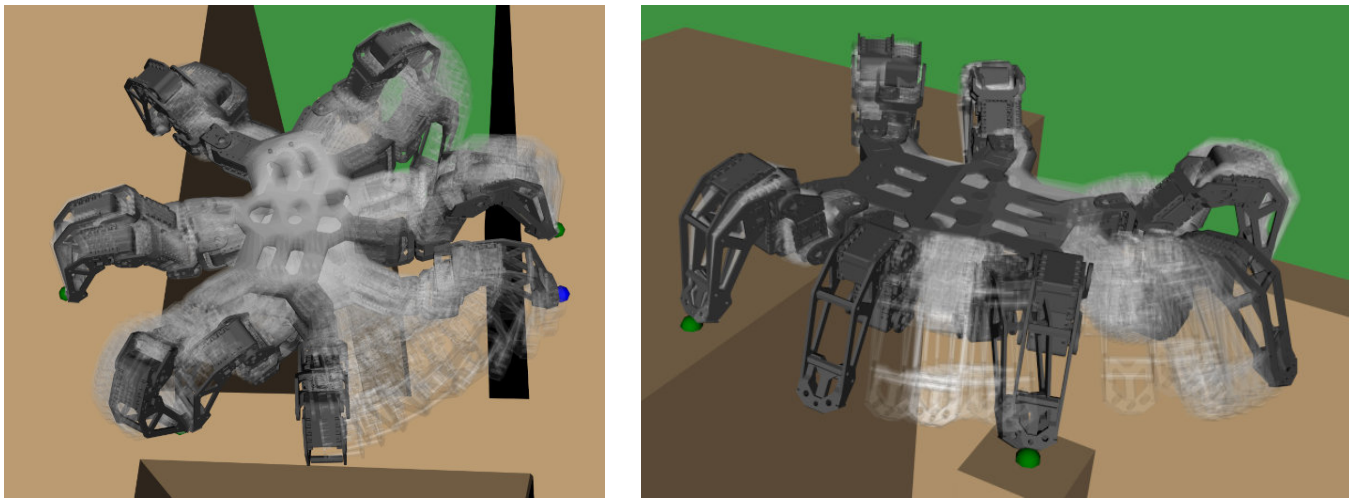
The computational kinematic model is of the hexapod



(a) Narrow gap scenario

(b) Wide gap scenario

Fig. 5. Leg contact diagram in narrow and wide gap scenarios. Blue cells signify the leg is in the stance phase.



(a) Single step in the narrow gap scenario

(b) Single step in the wide gap scenario

Fig. 6. Single step parametrized as the Bézier curve in the configuration (joint) space.

walking robot deployed in [24]. The planner has been implemented in C++, and the computational environment consists of the Intel i7-8565U running at 4.6 GHz accompanied by 32 GB RAM.

TABLE I

SEQUENCING RESULTS

Scenario	γ	**Time** [min]	**No. of Stance Expansions**	**Sequence Length**
Narrow	1000	55	4677	155
Wide	1000	31	3375	155

A summary of the sequencing part of the planning is depicted in Table I; it lists the planning time, the number of expanded stances, and the length of the resulting sequence. The heuristic scaling factor $\gamma = 1000$ is used in both cases. The contact diagrams representing which legs are assigned to a foothold by individual stances in the found sequences are visualized in Fig. 5.

One "interesting" step is selected in each sequence to demonstrate the proposed step planning with the Bézier curve parametrization. The stances at which the step is taken and the two adjacent stances are marked in the contact diagram in Fig. 5. In the narrow gap scenario, the step is chosen for the situation where the right-rear leg is moved between two distant footholds, see Fig. 6a. In the wide gap scenario, the chosen step is when the robot switches the right-front leg for the right-middle leg on the stepping stone, shown in Fig. 6b. The planning results for the two described steps are summarized in Table II showing the maximal degree of the Bézier curve, the number of iterations of Algorithm 2, the planning time, and the used tolerance ϵ.

Discussion: The feasibility of the proposed gait-free planning has been validated in the evaluation scenarios. The gait-free property can be observed in the diagram of the individual

TABLE II
STEP PLANNING RESULTS FOR SINGLE STEP

Scenario	Max Degree	No. of Iterations	Time [sec]	ϵ [m]
Narrow	5	747	4.0	0.001
Wide	6	1465	8.3	0.001

legs' contact with the ground. Notably, in the scenario with the wide gap, the right-middle leg is placed on the "stepping stone" for a significant portion of the sequence. The right-rear leg is not in any foothold for most of the sequence, demonstrating the gait-free capability of the hexapod walking robot. The sequencing part is computationally demanding because each sequence induces expensive sampling of the intermediate configurations needed to validate the graph edges. However, we expect the performance can be improved by a more informed heuristic and employing fast rejection of infeasible edges. Early results support that idea, but a detailed description is out of the scope of the paper. On the other hand, the single-step planning using low-capacity path parametrization shows to be a suitable choice. Even the steps with a substantial difference between the initial and the final configurations have been successfully determined up to the specified precision.

VI. CONCLUSION

In this paper, we present a gait-free planning framework for hexapod walking robots based on decomposition into the sequence determination and sequence validation parts. In the sequencing part, a candidate sequence of stances with intermediate configurations provides a necessary condition heuristic on feasible paths connecting the stances in the sequence. In the sequence validation part, finding smooth paths that connect the intermediate configurations is attempted considering the robot's motion constraints. A feasible solution is provided if a complete path is found. The sequencing procedure is relatively slow in the current state; however, it provides the candidate sequences needed for planning while considering all the constraints – even collisions with the terrain. We believe it can be further accelerated by a more refined heuristic function and a fast test for rejecting infeasible edges. The proposed low-capacity parametrization in single-step planning showed to be a viable approach; however, we plan to validate the generalizability of the parametrization to platforms with different kinematics. Besides, we aim to validate the approach on real hexapod crawlers.

REFERENCES

[1] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, and A.-a. Agha-Mohammadi, "Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2518–2525.

[2] R. Buchanan, L. Wellhausen, M. Bjelonic, T. Bandyopadhyay, N. Kottege, and M. Hutter, "Perceptive whole-body planning for multilegged robots in confined spaces," *Journal of Field Robotics*, vol. 38, no. 1, pp. 68–84, 2021.

[3] G. Wiedebach, S. Bertrand, T. Wu, L. Fiorio, S. McCrory, R. Griffin, F. Nori, and J. Pratt, "Walking on partial footholds including line contacts with the humanoid robot atlas," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 1312–1319.

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

[5] 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. 1–8.

[6] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[7] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[8] 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.

[9] 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.

[10] 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.

[11] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 995–1001.

[12] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[13] J. Norby and A. M. Johnson, "Fast global motion planning for dynamic legged robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3829–3836.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, "Footstep planning for autonomous walking over rough terrain," in *IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 9–16.

[19] 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.

[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] E. Tennakoon, T. Peynot, J. Roberts, and N. Kottege, "Probe-before-step walking strategy for multi-legged robots on terrain with risk of collapse," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5530–5536.

[22] D. Belter and P. Skrzypczynski, "Rough terrain mapping and classification for foothold selection in a walking robot," *Journal of Field Robotics*, vol. 28, no. 4, pp. 497–528, 2011.

[23] 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.

[24] 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.