Robust Rough-Terrain Locomotion with a Quadrupedal Robot

Péter Fankhauser, Marko Bjelonic, C. Dario Bellicoso, Takahiro Miki and Marco Hutter

Abstract— **Robots working in natural, urban, and industrial settings need to be able to navigate challenging environments. In this paper, we present a motion planner for the perceptive rough-terrain locomotion with quadrupedal robots. The planner finds safe footholds along with collision-free swing-leg motions by leveraging an acquired terrain map. To this end, we present a novel pose optimization approach that enables the robot to climb over significant obstacles. We experimentally validate our approach with the quadrupedal robot ANYmal by autonomously traversing obstacles such steps, inclines, and stairs. The locomotion planner re-plans the motion at every step to cope with disturbances and dynamic environments. The robot has no prior knowledge of the scene, and all mapping, state estimation, control, and planning is performed in real-time onboard the robot.**

I. INTRODUCTION

In the quest to get robots working on real-world tasks, many challenges have to be overcome. One of the fundamental requirement is the robot's ability to move in the designated environment safely. Our work focusses on quadrupedal robots to advance to places which traditional wheel- and track-based robots cannot reach. With legs, a robot can climb over difficult terrain by selectively choosing safe footholds and using all joints to move the limbs around obstacles and to adapt the posture to the terrain. To plan these climbing motions, the robot needs to understand its environment to select footholds and to avoid collisions. Once planned, the motions need to be executed robustly, and disturbances need to be compensated to prevent divergence from the planned motion. Finally, while moving, the robot needs to keep track of its pose relative to the environment continuously.

In this work, our focus lies on the planning and execution of climbing motions over rough terrain with a quadrupedal robot. As rough terrain, we refer to ground with various obstacles with a height up to 40 % of the robot's leg length. Fig. 1 shows a set of stairs as one example of this type of terrain. Building up on our previous work on control [1], state estimation [2] and terrain mapping [3], we combine these elements to a fully integrated system, capable of traversing difficult terrain. We emphasize the robustness and mean that the task is achieved in a realistic scenario without any previous knowledge or external equipment:

- The environment is previously unseen, and the robot has to perceive, plan, and adapt its motion as it travels. The robot might be disturbed, and the environment might change during locomotion.
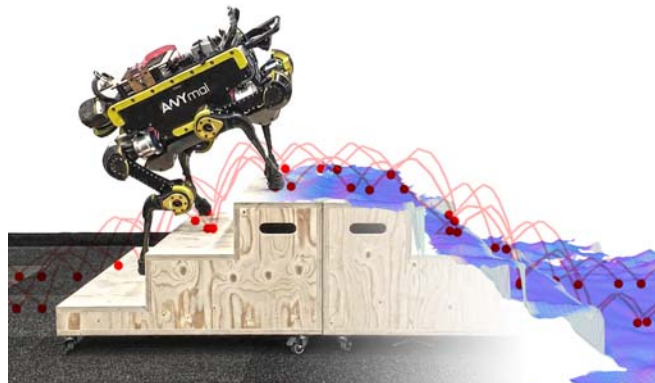
Fig. 1. The robot ANYmal [4] climbing a set of standardized stairs (17 cm × 29 cm). With help of onboard sensors, a terrain map is generated to plan the footholds (red dots) and base and leg motions (red lines). A video is available at `https://youtu.be/CpzQu25iLa0`.

- All sensing and computation is onboard, and the robot runs fully self-contained. Sensors and actuators are corrupted by noise, drift, and delay and we take no assumptions on any 'perfect' data.
- All algorithms run real-time in the sense that the robot never stops to 'think' about the next steps.

Under these requirements, only a few systems have been presented. Most prominently are Boston Dynamics' Spot and SpotMini robots, which have been shown to traverse difficult terrain including stairs at a high pace. However, very little information about this technology and the experimental setup is available. A few research groups have been working on the control and planning for the LittleDog robot (e.g., [5, 6, 7]). While impressive performances were shown, the approaches relied on an external motion capture system and high resolution pre-scanned terrain models. Recent advances towards a fully integrated system were published for the Messor II robot [8]. This work uses an A* guided Rapidly-exploring Random Tree (RRT)-based planner to find a kinematic plan, integrating foothold selection and motion planning in one sampling-based planner. While some results have been shown in combination with online state estimation and mapping, the robot is tethered to an external computation and power source, and no information about obstacle height and execution speed is given. Closest to our approach is the work on the hydraulic quadruped HyQ (90 kg, 1 m height) by [9, 10, 11]. While still tethered, the robot has shown autonomous climbing over previously unseen obstacles. The work of [10] builds upon the architecture of [5] and extends it among others with online state estimation and terrain mapping. With this, the authors report a max. achieved step
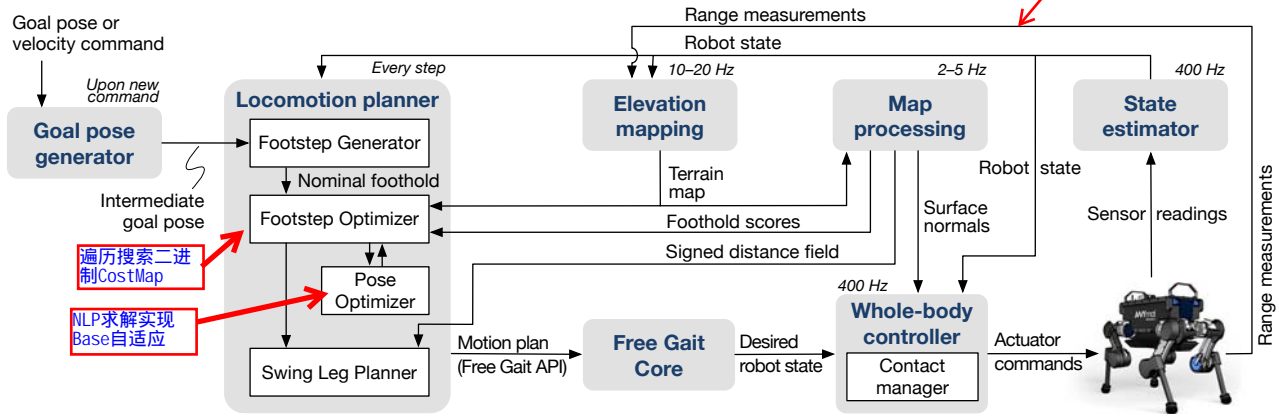
Fig. 2. Based on a goal pose or velocity, the locomotion planning framework generates a new motion plan which is tracked by the whole-body controller. Sensory information is used to estimate the robot's pose and to build a map of the surrounding terrain. The terrain map is processed, and valid footholds are identified. At every step, new foothold locations and swing leg trajectories are planned to account for disturbances and changing environments.

height of 15 cm (20 % leg length) at a speed of 7–13 cm/s.

While our work is similar in structure to [5] and [10], we contribute a novel robot pose generation based on non-linear optimization. During locomotion planning, the pose optimization is coupled with the foothold selection to exploit the full range of motion of the robot. Additionally, a leg motion planner ensures safe swing leg motions over and around obstacles. With this, our robot platform ANYmal (30 kg, 0.5 m height) [4] reaches obstacles up to 20 cm, which is an increase of 100 % in comparison to [10] regarding the different leg lengths. At the same time, we achieve robust behavior and high success rates (∼90 %) with the help of our control framework [1] even under disturbances and changing environments. In our experiments, we have achieved average velocities of 8–15 cm/s in rough terrain, which is an increase of 50 % over [10] regarding the different leg lengths. Our approach is built on our robot-centric elevation mapping approach with which consistent terrain maps are obtained solely based on proprioceptive localization, such as legged odometry with kinematic and inertial sensing. This alleviates the need for an accurate exteroceptive localization which can be problematic in environments with insufficient geometrical or visual features [8, 9]. Furthermore, our approach makes use of the torque-controllable actuation of ANYmal to apply the appropriate contact forces on inclined terrain based on the surface normal information from the terrain map.

In the following, we give an overview of our approach in Section II. We introduce the map processing steps in Section III and present the footstep planning and pose and swing leg motion planning in Section IV. The results for various experiments are given in Section V and the work is concluded in Section VI.

II. OVERVIEW OF THE METHOD

Our proposed method consists of several modules that are all interconnected but run in parallel at different update rates. Fig. 2 shows an overview of the modules and the information flow. Starting with a desired final goal pose (defined by a position and a yaw-orientation angle) or desired velocity

command, a next intermediate goal pose is computed. This is to ensure that the robot does not walk long distances in a direction in which no range sensor is mounted. The range sensor continuously measures the distance to the terrain and is used together with the robot pose estimate to create an elevation map around the robot at 10–20 Hz depending on the type of range sensor. This terrain map is shifted along with the motion of the robot and stores the height information beneath and around the robot and fuses existing data with new measurements. At a lower rate of 2–5 Hz, the elevation map is processed to compute foot score values, surface normal vectors, and a three-dimensional Signed Distance Field (SDF). The foot score values reflect the suitability of an area in the map to place a foot on.

The terrain information is used in the locomotion planner to find a feasible motion of the robot over the perceived terrain. The final output of the planner is the target foothold together with the swing leg motion for the next step. First, based on the goal pose and the current stance (position of the legs), a sequence of (planar) footsteps is generated. These footsteps do not consider the terrain but represent the nominal gait sequence and step lengths. To find a feasible footstep, the area around the nominal footstep position is searched for the closest valid foothold. Since valid footholds are not necessarily reachable, a pose optimizer is used to check if a stable robot configuration can be found which respects stability and kinematic constraints. Once a candidate foothold is verified by the pose optimizer, the swing trajectory can be planned. Based on the terrain's SDF, a collision-free swing trajectory is planned which connects the current foot position and the optimized foothold target. As the map is updated continuously and because the robot might not reach previously planned footholds (due to slip or compliant ground), this sequence of steps in the locomotion planner is performed at every footstep to adapt to these changes.

Based on the target foothold and the swing leg motion plan, the Free Gait Core module synthesizes a whole-body motion. Internally, it uses the same pose optimizer for the generation of the base motion as previously in the locomotion

5762

planning process, ensuring that the planned leg motions are feasible. Given the full motion plan, the whole-body controller executes and stabilizes the robot's behavior around the desired trajectories. The controller runs at 400 Hz to generate the actuator commands from a closed-loop with the state estimator. It uses the terrain surface normal information from the map to generate contact forces that respect the boundaries of the friction cones. Rather than executing the trajectories regardless of the current state, the controller ensures that the desired contact situations are reached (contact manager). Because the motion plans always contain errors, this is a crucial step in the robust tracking of the motion.

III. MAP PROCESSING

In our previous work [3], we presented an approach to create 2.5 D elevation maps from the combination of pose estimate and a range measurement sensor. In this representation, each cell of the grid holds a height and a lower and upper confidence bound. Our approach takes the uncertainty of the pose estimate and the range measurements explicitly into account to obtain consistent maps even without requiring an additional localization system such as GPS or vision- or laser-based registration. The elevation mapping framework works with different range sensors such as laser range sensors, time-of-flight, structured light, and (assisted) stereo cameras. Given the continuously updated robot-centric elevation map, the terrain is processed to compute surface normals, find valid footholds, and build a three-dimensional signed distance field (SDF) used for planning of collision-free swing trajectories.

A. Surface Normal Vectors

We use a standard way to extract the surface normals from the noisy surface representation based on the Principal Component Analysis (PCA) [12]. Given the surface points in a circular region and subtracting their mean position, the surface normal can be estimated from the Eigen vector associated to the smallest Eigen value of the subset. As the nearest neighbors for each cell in the grid can be accessed directly, the surface normals can be computed efficiently for the entire map.

B. Foothold Scores

Each cell in the elevation map is associated with a foothold score $s(x,y) \in [0,1]$. This value represents the quality of the foothold, where a cell with $s = 1$ is a safe foothold, and a cell with $s = 0$ might cause the foot to slip. Similarly to other work [10, 13], we have found that a linear combination of different quality measures

$$s(x,y) = \max\left(1 - \sum_i w_i \frac{v_i(x,y)}{v_{\text{crit}}}, 0\right) \quad (1)$$

provides a simple but reliable foothold score. The variable w_i represents the weight factor for each quality measure i with value $v_i(x,y)$ and max. allowed/critical value v_{crit}. As quality measures, we evaluate the slope and curvature (derived from the surface normals) and the roughness (height standard
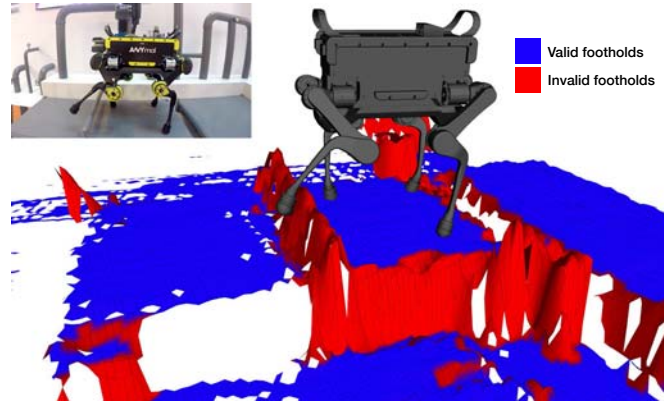

Fig. 3. The elevation map is processed to differentiate regions with valid (blue) and invalid footholds (red). The foothold classifications is based on local terrain features such as slope, curvature, roughness, and uncertainty of the map data.

deviation) from the neighboring cells. We additionally use the uncertainty interval (the difference between lower and upper confidence bound) of the cell as a quality measure to steer the robot away from potentially hazardous regions such as steps and gaps. This approach enables robust foothold selection even under the influence of drift from the state estimation of the robot. To further simplify the foothold selection process, we create a binary foothold score $\hat{s}$ based on a threshold to map s to either 0 or 1. Fig. 3 shows an elevation map where the color represents the binary foothold score $\hat{s}$ for valid and invalid footholds.

C. Signed Distance Field

Planning collision-free trajectories can be solved with the help of an SDF map representation. We generate a three-dimensional SDF from the elevation map, where each voxel holds the distance to the nearest surface. The SDF can be efficiently computed with the approach presented in [14].

IV. MAP-BASED MOTION PLANNING

Based on the current state of the robot, the goal pose, and the terrain information (Section III), the goal is to find a feasible sequence of contacts and whole-body motions to let the robot overcome the rough terrain. Our approach is a motion planner that, starting from a nominal footstep, finds the next feasible foothold (one-step planning horizon) with the help of the combination of the binary foothold score map and a robot pose optimizer. Once a foothold is verified to be feasible from both the foothold score map and the pose optimizer, the swing leg is planned in a second step. Finally, the desired motion plan is executed while accounting for errors in sensing and actuation.

A. Footstep Generation

For generating the nominal footsteps, we employ a geometrical approach which is recomputed at each step (Fig. 4): First, the current stance position and orientation (*start pose*) is determined based on the current foot positions. The start position is given by the geometrical center of the feet and the
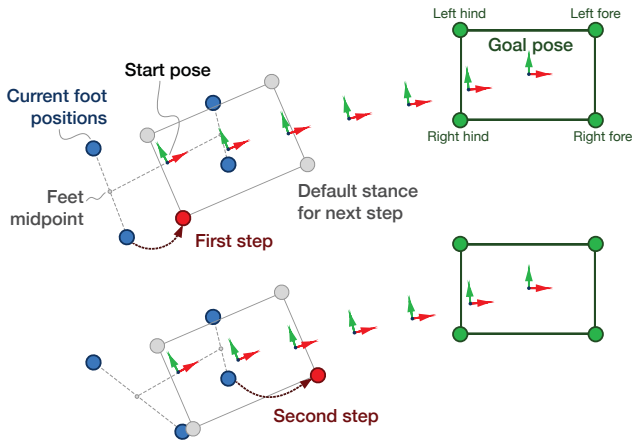
Fig. 4. Illustration of the two first steps of the nominal footstep generation. At every step, a sequence of default stances is (re-)computed that guide the robot from the start pose to the goal. The foot position in the default stance of the subsequent pose is selected as the target for the current swing leg. This procedure converges to a skew between left and right legs of the robot, which is important for stability and speed.
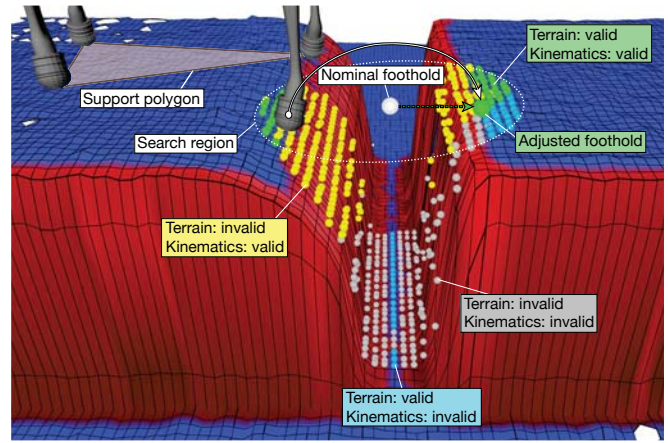

Fig. 5. Starting with the nominal foothold (Section IV-A), candidate footholds in the search region are checked for their validity. Terrain validity is indicated by the foothold scores based on the local geometric features (Section III-B). Kinematic validity is checked with the solution of a pose optimization (Section IV-C). The nominal foothold is adjusted to the closest candidate foothold which is valid for both terrain and kinematics.

start orientation is given by the orientation of the connecting line between the midpoints between hind and forefeet. Based on the difference between the start and goal pose, the main direction (longitudinal, lateral, and yaw rotation) is determined. Given the main direction and the default step length, the path between the start and goal pose is linearly sampled with *n* equal *default stances*. Here we use a priori knowledge of a default step length for longitudinal, lateral, and yaw orientation steps. Each stance is a representation of how much the robot travels for a full cycle in the main direction with the default step length. The leg sequence is determined with the main direction and is given by the optimality analysis by [15]. For the forward direction, the leg sequence is right-hind, right-fore, left-hind, left-fore. The new footstep location is given by the position of the corresponding foot in the first default stance after the start pose. If the main direction changes, the leg with the largest first step is chosen as the new start of the sequence.

This entire procedure is repeated after every step and provides a suitable default foothold pattern. Apart from being simple and extremely quick to compute, there are several important properties of the approach: The algorithm can start from any configuration of the feet and always ends up in the default stance at the goal pose. Since the algorithm is recomputed every step, this method converges to a foothold pattern with a skew between left and right legs. This skew is essential for speed and stability and reduces the required base motion [7]. Finally, this approach is robust against any deviation from the original plan and allows the robot to recompute the next step towards the goal at each step.

B. Footstep Optimization

The footstep determined by the nominal footstep generator is verified for kinematic and terrain feasibility. If the nominal footstep is not a feasible foothold location, the nearest neighbor valid foothold location is found as illustrated in

Fig. 5. The positions around the nominal foothold are iterated through with increasing Euclidian distance from the nominal foothold. Each position is checked for its validity by checking the binary foothold score for the cells in a patch around the candidate position. We choose a circular patch with a diameter of 8 cm, which is slightly larger than the foot size of ANYmal to account for foot rolling during locomotion. If all binary foothold scores in the patch are valid, a candidate foothold is found.

Although the candidate foothold might be a feasible position based on the terrain, the robot might not be able to reach it due to kinematic constraints. For this reason, every candidate foothold is checked for its reachability by running the pose optimizer with the current and the candidate foothold as input. If the pose optimizer does not find a solution, the iteration continuous. Once the search region is exhausted, the procedure is stopped, and the method signalizes that no valid footstep could be found.

C. Pose Optimization

The pose optimization is a crucial part of our approach to enable climbing up and down high obstacles. When reaching for places that are far apart from the previous foot location, the robot needs to adapt its pose such that the new foothold can be reached without sacrificing stability or running into joint position limits. This task can be formulated as an optimization problem as follows: Given N foot locations r_{F_i}, the goal is to find a robot base position r_{IB} and orientation Φ_{IB} which maximizes reachability and the stability margin under the constraint of static stability and allowed joint limits. We encode the reachability in the cost function $f(r_{IB}, \Phi_{IB})$ by penalizing the difference between a default foot position $\hat{r}_{F_i}$ and the foot position resulting from the optimized posed $r_{F_i}(r_{IB}, \Phi_{IB})$. Additionally, we maximize the static stability margin by adding a term that penalizes the distance between the planar projection of the centroid of the Support Polygon
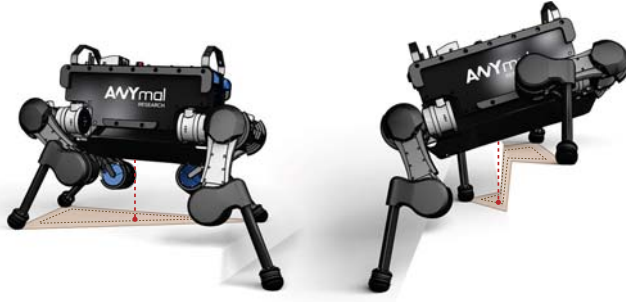
5764

Fig. 6. Our pose optimization formulation efficiently finds the full base pose (6D position and orientation) for given contact situations.
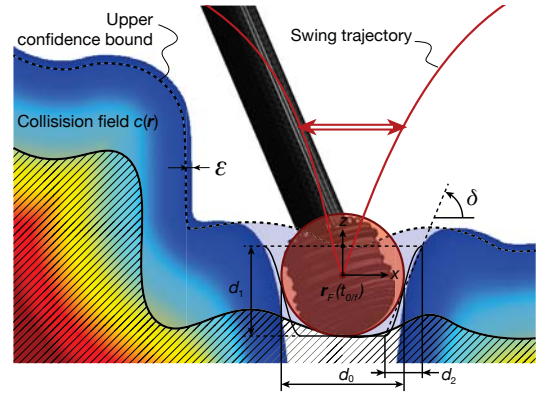

Fig. 7. Collision-free swing leg trajectories are planned with help of the 3D collision fields (here shown in 2D). The collisions field represents the upper confidence bound of the map, to take the uncertainty of the mapping process into account.

(SP) $\bar{r}_{\text{SP}}$ and the planar projection of the position of the Center Of Mass (COM) $\bar{r}_{\text{COM}}(r_{IB}, \Phi_{IB})$:

$$\text{minimize} \quad f(r_{IB}, \Phi_{IB}) = \sum_{i=0}^{N} \left\| \hat{r}_{F_i} - r_{F_i}(r_{IB}, \Phi_{IB}) \right\|_2^2 \quad (2)$$
$$+ w_{\text{COM}} \left\| \bar{r}_{\text{SP}} - \bar{r}_{\text{COM}}(r_{IB}, \Phi_{IB}) \right\|_2^2$$
$$\text{subject to} \quad A_{\text{SP}} \, r_{\text{COM}}(r_{IB}, \Phi_{IB}) \leq b_{\text{SP}} \quad (3)$$
$$l_{\text{min},i} \leq \left\| l_i(r_{IB}, \Phi_{IB}) \right\|_2 \leq l_{\text{max},i} \quad \forall i \in N. \quad (4)$$

We use the support polygon as the area defined by the convex hull of the support feet positions projected along the direction of gravity. The robot is statically stable if the projection along gravity of the COM $\bar{r}_{\text{COM}}$ lies with the support polygon. As the COM position $_B r_{\text{COM}}$ is a function of the joint state vector q, we employ inverse kinematics to find q from $_B r_{F_i}$ for each leg i. We encode the static stability in (3) where A_{SP} and b_{SP} are the linear constraints spanned by the support polygon. Optionally, we add a support margin which offsets the support polygon inwards to increase robustness (as illustrated in Fig. 6). The joint limits are expressed with the inequality constraint (4) with l_i the leg length given by the distance from the hip joint to the foot for leg i. By adjusting the weighting term w_{COM} in (2), we can achieve a suitable trade-off between maximizing the kinematic reachability and the stability margin (we choose $w_{\text{COM}} = 2$ for ANYmal).

We can solve this non-linear optimization problem with Sequential Quadratic Programming (SQP) with an analytical derivation of the gradients and Hessians of the cost function (as shown in [16]). Due to the non-linearity of this problem, the success of finding a solution is dependent on the initial guess. To this end, we use an initial guess based on a least square optimization of the kinematic error for leg i

$$e_i(r_{IB}, \Phi_{IB}) = {}_I r_{IB} + \Phi_{IB}({}_B \hat{r}_{F_i}) - {}_I r_{F_i}. \quad (5)$$

This error can be minimized by transforming it to an Eigen value problem when parametrizing the rotation Φ_{IB} as a unit quaternion [17].

Fig. 6 depicts the solution of the pose optimization formulation for different contact situations. Depending on the situation, the robot's base is lowered and pitched to satisfy the kinematic and stability constraints. With our implementation (quadratic problem solved with the QuadProg++ active set implementation, hardware details see Section V), the pose

optimizer yields a solution to all practical situations within 0.5–2.5 ms with maximal 30 iterations of the SQP.

D. Swing Leg Planning

Given the start and target foothold position, the trajectory of the swing leg needs to be planned to find the shortest collision-free trajectory. We reduce this task to the problem of finding a trajectory of the foot only and limit the collision checking with the terrain to a sphere (diameter 6 cm) around the foot. Inspired by the CHOMP trajectory planner [18], we formulate the planning problem as an optimization. We parametrize the swing leg trajectories as cubic Hermite splines and optimize directly over the knot points (except for the start and target). Herewith, the optimization yields continuous paths that can be smoothly tracked by the robot. We stack the 3D knot positions as a vector u and the position on the spline for time t is given by $r(u, t)$. The optimization cost function is expressed as a weighted sum of length and collisions costs, $f_l(u)$ and $f_c(u)$, respectively:

$$\text{minimize} \quad f(u) = w_l f_l(u) + w_c f_c(u) \quad (6)$$
$$= w_l \int_{t=t_0}^{t_f} \left\| \dot{r}_F(u, t) \right\|_2 dt + w_c \int_{t=t_0}^{t_f} c(r_F(u, t)) \left\| \dot{r}_F(u, t) \right\|_2 dt$$
$$\text{subject to} \quad z_{\text{min}} \leq u_{i,z} \leq z_{\text{max}} \quad \forall u_i \in u. \quad (7)$$

We constrain the knot points z-coordinate $u_{i,z}$ to lie within a minimum clearance z_{min} and maximum height z_{max}. The function $c(r_F)$ represents a collision field based on the terrain map

$$c(r) = c_t(r) c_{t_0}(r) c_{t_f}(r). \quad (8)$$

The function $c_t(r)$ is a smooth potential function for distance to an obstacle based on the distance to an obstacle $d(r)$ acquired from the SDF [18]. Note that we use the upper confidence bound of the elevation map for the generation of the SDF as illustrated in Fig. 7, where ε is the allowable distance threshold. This ensures that the foot does not collide with the terrain even if the knowledge of the terrain is coupled with uncertainties. The functions $c_{t_0}(r)$ and $c_{t_f}(r)$

5765

create funnels in the collision function to allow the foot to create collision-free contact at the start t_0 and end t_f of the swing motion. They are expressed as a smooth function based on the dead zone d_0 and the desired funnel angle δ (Fig. 7):

$$c_{t_{0,f}}(\boldsymbol{r}_F(t)) = \frac{1}{2}\left(1 + \tanh\left(\frac{d_{0,f}(\boldsymbol{r}_F(t))}{d_2}\right)\right), \quad (9)$$

$$d_{0,f}(\boldsymbol{r}_F(t)) = \left\|\overline{\boldsymbol{r}}_F(t) - \overline{\boldsymbol{r}}_F(t_s, t_f)\right\|_2 - d_0, \quad (10)$$

$$d_2 = \frac{d_1}{\tan\delta}. \quad (11)$$

We solve the swing motion optimization problem with SQP and derive the gradients and Hessians numerically. For the integrals in (6) we use a time resolution of $\Delta t = 50\,\text{ms}$ to provide a dense resolution for collision checking. As this optimization might get stuck in a local minimum, we restart the optimization with an increased number of knots until a solution is found.

E. Integration with Motion Control

In the presented approach, the generated motion plan is based on a geometrical representation of the terrain. However, the ground might be compliant (e.g., sand or grass) and objects might move or be deformable (e.g., a plastic bag) in reality. For the successful execution the generated motion plan, we build upon our Free Gait controller framework [1]. The locomotion planner outputs the desired footsteps as foot swing trajectory knot points in the elevation map frame. With this information, the controller can track the motion relative to the terrain and any slip or error of the robot base control is compensated for. During execution, the Free Gait controller uses the presented pose optimization (Section IV-C) with which it interpolates between robot poses that comply with desired motion plan. At every step, the controller has to ensure that the contact between foot and terrain is achieved. To this end, the controller (contact manager) handles pre- and post-mature contact switches with a strategy described in [16].

V. RESULTS

We have implemented the approach described in this paper on the quadrupedal robot ANYmal [4]. The robot weights $30\,\text{kg}$ and has a leg length of $0.5\,\text{m}$. ANYmal's has three joints per leg which are driven by electrically driven series elastic actuators (ANYdrive), enabling accurate torque control. All the presented algorithms run on two onboard PCs[1] which are powered by a battery which allows for a power autonomy of up to $3\,\text{h}$. As range sensors, we either use a rotating laser range sensor (Hokuyo UTM-30LX-EW, update rate $0.5\,\text{Hz}$ for a full rotation) or an assisted stereo camera (Intel RealSense ZR300, update rate at $20\,\text{Hz}$ for a full depth image) which are mounted in the front and back of the robot. The robot's underlying motion controller consists of a virtual model controller for the base and a contact force

[1]Intel Core i5 2.3 GHz, 16 Gb for control and planning / Intel Core i7 3.1 GHz, 16 Gb for terrain mapping

Step height		Avg. velocity	w/o map	w/ map
Step up 7 cm	13 % leg length	15 cm/s	100 %	100 %
Step down 7 cm			100 %	100 %
Step up 14 cm	25 % leg length	11 cm/s	70 %	100 %
Step down -14 cm			90 %	100 %
Step up 21 cm	38 % leg length	8 cm/s	0 %	90 %
Step down -21 cm			80 %	80 %

distribution optimization [16]. A video[2] showing the results is accompanied to this paper.

A. Step Up and Down

We have commanded ANYmal to walk over a flat step at various heights, which includes a step up and a step down maneuver. For comparison, we have conducted the experiments for the pure reactive walking without exteroceptive perception (*without map*) and with active range sensing (*with map*). At each trial, the robot started each from different positions. For the experiments without mapping, we have globally adjusted the step height to match the obstacle. The success rate out of ten trials for each type of combination is summarized in Table I. The average velocity is measured as the total covered horizontal distance divided the duration that the robot takes to clear the obstacle.

For a step height of $7\,\text{cm}$ ($13\,\%$ of leg length), both approaches never failed during our trails. For the step with $14\,\text{cm}$ ($25\,\%$ of leg length), stepping up without perception led to some failure cases, where the footstep would be placed to close to the step and experienced a collision during the following step. This was successfully prevented with active perception. For the $21\,\text{cm}$ step ($38\,\%$ of leg length), stepping up without perception was not possible, due to the fact that reaching this height without tilting the pose cannot be achieved. However, stepping down was still mostly successful due to the step recovery approach of the motion controller [16]. Stepping up to $21\,\text{cm}$ was again significantly improved with active perception. Fig. 8 shows snapshots of one trial on the step with $21\,\text{cm}$ height with active mapping. One can observe how the base is tilted to facilitate stepping up (Fig. 8a, b). At later stages (Fig. 8c–e), the leg length constraint in the pose optimizer prevents over-extension of the legs.

B. Inclined Surfaces

With the help of the torque-control capability of ANYmal, we make use of the knowledge of the surface normals from the elevation map (Section III-A). The controller takes the tilted friction cones in the force distribution as constraints into account. This extends ANYmal capability to overcome difficult terrain with inclined surfaces. Fig. 9 shows an image from an experiment on a concave and a convex structure. In this situation, the contact forces are regulated such that

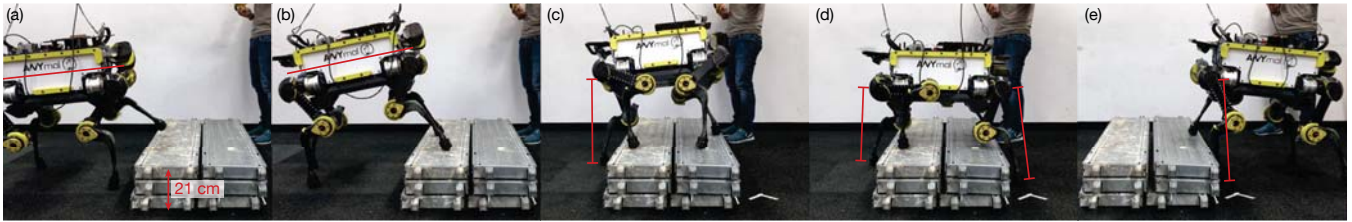[2]Available at https://youtu.be/CpzQu25iLa0

Fig. 8. The locomotion planner is used to climbing over a step of 21 cm with active terrain mapping with the frontal laser range sensor. The planning framework finds a sequence of safe footholds, base poses, and swing trajectories that guide the robot over the obstacle with an average velocity of 8 cm/s.
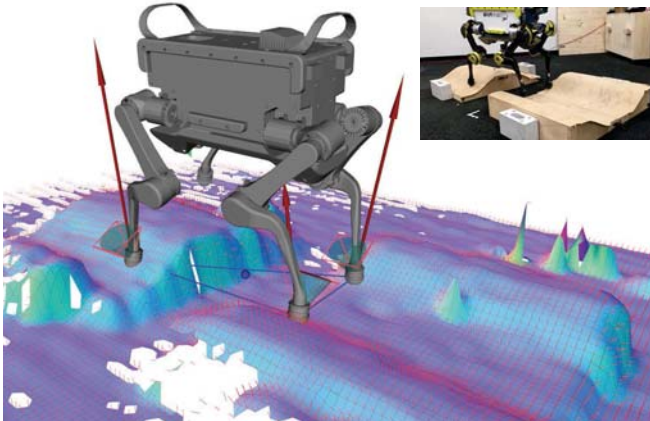

Fig. 9. Surface normals are computed from the terrain map to enhance the controller performance on inclined surfaces. Based on the orientation of the friction cones, the contact forces are regulated to respect the friction constraints preventing the feet from slipping.

they push 'inwards' or 'outwards' to prevent slippage. In our experiments, walking over the same obstacles without taking the surface normals into account was not possible.

C. Changing Environment

Demonstrating the reactivity of our approach, we expose ANYmal to a quickly changing environment. As depicted in Fig. 10, we have commanded forward velocity while placing cinder blocks in front of the robot. The continuous map updates and step re-planning allowed successful planning of the footholds and trajectories around the obstacles.

D. Stairs

Stairs can be seen as a concatenation of multiple steps in both upward and downward direction. However, they are significantly more demanding because the front and hind legs are often multiple steps apart which makes it difficult to find a motion that fulfills the kinematic, stability, and joint torque limitations. We have evaluated our approach on a standardized set of stairs with a rise of 17 cm and a run of 29 cm as shown in Fig. 1. The system has no prior knowledge about the shape of the stairs and the planning framework deals with it like with any other obstacle. ANYmal was able to climb the stairs upwards and downwards multiple times. However, climbing the stairs up turned out to be more robust than climbing down because of the limited view on the lower stairs when heading downwards.

E. Robustness and Versatility

We have conducted several experiments to illustrate the robustness of our approach. In multiple trials we have disturbed

the execution of approaching a goal, requiring the robot to adapt and replan the motion. For the experiment shown in Fig. 11a, we have moved the whiteboard ∼1 m while the robot was walking over it. In this case, the robot localized with respect to the environment with Iterative Closest Point (ICP) [19] such that the navigation goal could be tracked in a global frame. While moving the board, the robot successfully kept balance, updated the terrain map, and replanned the footsteps such that it could reach the goal. Other experiments further demonstrate the robustness of the framework such as walking over a thin ledge and climbing over a seesaw and a human (Fig. 11b–d).

VI. CONCLUSIONS

We presented a motion planning method for the perceptive rough-terrain locomotion with a quadrupedal robot. All sensing and computation run onboard the robot in real-time without any prior knowledge of the terrain. Based on the acquired elevation map, the framework finds a sequence of suitable footholds and generates base and swing leg trajectories that guide the motion of the robot over the obstacles. A core element of the framework is the proposed pose optimization which efficiently finds full robot poses for given foot placements. The planner is tightly coupled with the terrain mapping, robot controller, and localization, allowing it to continuously re-plan the motion while robustly accommodating disturbances and changes in the environment. We have evaluated the effectiveness of our approach in experiments including climbing over steps, inclined surfaces, stairs, and various disturbance tests with changing environments.

In this work, the primary assumption for foothold planning is that we can find a feasible motion by planning the next step while executing the previously planned step. This approach is limited by the greedy behavior of the one-step planning horizon. However, we have found that this approach works reliably for the tested terrains described in Section V including stairs. A similar experience was reported by [5, 20], where multi-step look-aheads were only necessary for obstacles larger than ∼50% of the leg length. Another assumption is that we can apply a sequential approach in which the swing leg trajectory is planned after the foothold selection similar to other works [7, 10]. Thereby, it is not guaranteed that the swing leg planner can find a feasible solution. However, we have found that with the help of the base motion towards to next stance during the swing motion and the large range of motion of our hardware, this is almost never an issue. We also reduce the collision checking to the

Fig. 10. As the robot is commanded to walk forward, obstacles are placed in front of it. The mapping framework continuously updates the map which allows the locomotion planner to adapt to the changing terrain.

Fig. 11. (a) The robot replans the motion to adapt to the disturbance which has been introduced by relocating the board during execution. In this demonstration, the robot localizes itself with the rotating laser range sensor and can robustly track and reach the desired goal location. (b)–(d) Various experiments demonstrating the versatility and robustness of our approach.

geometry of the foot, and we neglect other parts of the leg. This simplification still yields satisfactory results in practice, as the unique form of ANYmal's shank avoids most of the shin collisions as can be observed for example in Fig. 8d.

In future work, we are focussing on extending the maximal obstacle height to over 50 % of the leg length by extending our motion planner with a multi-step planning horizon. Furthermore, an increase of the locomotion speed is targeted by employing fully dynamic gaits such as a trot.

REFERENCES

[1] P. Fankhauser, C. D. Bellicoso, C. Gehring, R. Dubé, A. Gawel, and M. Hutter, "Free Gait – An Architecture for the Versatile Control of Legged Robots," in *IEEE-RAS International Conference on Humanoid Robots*, 2016.

[2] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State Estimation for Legged Robots on Unstable and Slippery Terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6058–6064, 2013.

[3] 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.

[4] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - a Highly Mobile and Dynamic Quadrupedal Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[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, pp. 236–258, nov 2010.

[6] 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, pp. 175–191, jan 2011.

[7] P. D. Neuhaus, J. E. Pratt, and M. J. Johnson, "Comprehensive summary of the Institute for Human and Machine Cognition's experience with LittleDog," *The International Journal of Robotics Research*, vol. 30, pp. 216–235, jan 2011.

[8] D. Belter, P. Łabęcki, and P. Skrzypczyński, "Adaptive Motion Planning for Autonomous Rough Terrain Traversal with a Walking Robot," *Journal of Field Robotics*, vol. 33, pp. 337–370, may 2016.

[9] I. Havoutis, J. Ortiz, S. Bazeille, V. Barasuol, C. Semini, and D. G. Caldwell, "Onboard Perception-Based Trotting and Crawling with the Hydraulic Quadruped Robot (HyQ)," in *IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, 2013.

[10] A. W. Winkler, C. Mastalli, M. Focchi, D. G. Caldwell, and I. Havoutis, "Planning and Execution of Dynamic Whole-Body Locomotion for a Hydraulic Quadruped on Challenging Terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[11] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini, "Trajectory and Foothold Optimization using Low-Dimensional Models for Rough Terrain Locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1096–1103, 2017.

[12] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, "Comparison of surface normal estimation methods for range sensing applications," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3206–3211, may 2009.

[13] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, "A control architecture for quadruped locomotion over rough terrain," *IEEE International Conference on Robotics and Automation (ICRA)*, may 2008.

[14] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance Transforms of Sampled Functions," *Theory of Computing*, vol. 8, no. 1, 2012.

[15] R. B. McGhee and A. Frank, "On the stability properties of quadruped creeping gaits," *Mathematical Biosciences*, vol. 351, 1968.

[16] P. Fankhauser, *Perceptive Locomotion for Legged Robots in Rough Terrain*. Doctoral thesis, ETH Zurich, 2018.

[17] M. Bloesch and M. Hutter, "Technical Implementations of the Sense of Balance," in *Humanoid Robotics: A Reference* (P. Vadakkepat and A. Goswami, eds.), Springer Netherlands, 2017.

[18] 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.

[19] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[20] B. Mitchell, A. G. Hofmann, and B. C. Williams, "Search-based Foot Placement for Quadrupedal Traversal of Challenging Terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1461–1466, apr 2007.