IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada, September 2017.
Finalist for Best Paper Award on Safety Security and Rescue Robotics.

Anytime Hybrid Driving-Stepping Locomotion Planning

Tobias Klamt and Sven Behnke

***Abstract*— Hybrid driving-stepping locomotion is an effective approach for navigating in a variety of environments. Long, sufficiently even distances can be quickly covered by driving while obstacles can be overcome by stepping. Our quadruped robot Momaro, with steerable pairs of wheels located at the end of each of its compliant legs, allows such locomotion. Planning respective paths attracted only little attention so far.**

We propose a navigation planning method which generates hybrid locomotion paths. The planner chooses driving mode whenever possible and takes into account the detailed robot footprint. If steps are required, the planner includes those. To accelerate planning, steps are planned first as abstract manoeuvres and are expanded afterwards into detailed motion sequences. Our method ensures at all times that the robot stays stable. Experiments show that the proposed planner is capable of providing paths in feasible time, even for challenging terrain.

I. Introduction

Hybrid driving-stepping locomotion enables robots to traverse a wide variety of terrain types. Application domains, such as search and rescue and delivery services, pose considerable navigation challenges for robots due to non-flat grounds. Sufficiently flat terrain can be traversed by driving, which is fast, efficient and safe, regarding the robot stability. However, driving traversability is limited to moderate slopes and height differences and obstacle-free paths. Stepping locomotion requires only adequate footholds and, hence, enables mobility in cases where driving is unfeasible. But stepping is also slower and decreases the robot stability.

Most mobile ground robots use either driving locomotion or stepping locomotion, and there exist path planning methods for both such locomotion modes independently [1]–[7]. Our mobile manipulation robot Momaro [8] (see Fig. 1), however, supports both locomotion types due to its four legs ending in steerable pairs of wheels. This unique design allows omnidirectional driving on sufficiently flat terrain and stepping to overcome obstacles. In contrast to purely walking robots, Momaro is able to change its configuration of ground contact points (which we will refer to as its footprint) under load without lifting a foot. This enables motion sequences for stepping that have large stability margins. Multiple platforms that are capable of driving-stepping locomotion have been developed [9]–[13], but planning which combines the advantages of both locomotion types was addressed for none of these.

All authors are with Rheinische Friedrich-Wilhelms-Universität Bonn, Computer Science Institute VI, Autonomous Intelligent Systems, Friedrich-Ebert-Allee 144, 53113 Bonn, Germany klamt@ais.uni-bonn.de, behnke@cs.uni-bonn.de.
This work was supported by the European Union's Horizon 2020 Programme under Grant Agreement 644839 (CENTAURO).

Fig. 1. Our hybrid wheeled-legged mobile manipulation robot Momaro is capable of omnidirectional driving (left) and stepping (right).

In our previous work with Momaro [14], we demonstrated semi-autonomous driving, 2D (x, y) path planning and execution in a Mars-like environment accompanied by manipulation tasks. In this work, we extend the driving path planning method to incorporate the robot orientation θ and its detailed footprint in order to increase driving flexibility. We improve the path quality by introducing an orientation cost term.

In addition, we demonstrated stepping over a wooden bar obstacle, climbing stairs, and egressing a car with Momaro at the DARPA Robotics Challenge (DRC) [15]. All of the DRC tasks were performed via teleoperation based on pre-defined motion sequences. Teleoperation depends on a good data connection between the operator station and robot and generates a high cognitive load for the operators. Autonomous locomotion is desirable to relieve the operators and to increase speed and safety.

We extend the locomotion planner to generate stepping motions. Driving in difficult terrain and stepping require a high planning resolution which increases planning times. To keep the search space feasible, we first generate abstract steps that we later expand to detailed motion sequences.
To summarize, the main contributions in this paper are:

- a three-dimensional (x, y, θ) driving path planning method allowing driving in constrained uneven environments by consideration of the detailed robot footprint,
- the introduction of orientation costs, favoring a preferred driving direction to align the robot with the path,
- a hierarchical step planner, which generates detailed manoeuvres to perform individual steps under the constraint to always keep the robot statically stable, and
- application of anytime planning to quickly find paths with bounded suboptimality.

We demonstrate our approach in simulation and with the real robot and systematically evaluate the effect of our acceleration methods. The results indicate that our planner provides paths in feasible time even for challenging tasks.

II. Related Work

Path planning in unstructured terrain has been addressed by many works. The considered systems provide either purely wheeled/tracked locomotion or are able to traverse terrain by walking. Planning is often done with either grid-based searches, such as A* [16], or sampling-based approaches, such as RRT [17]. Despite the application of similar planning methods, these two locomotion modes differ in many aspects.

Driving is fast and energy efficient on sufficiently flat terrain, which makes it suitable for traversing longer distances. When supported by three or more wheels, the robot is generally statically stable. Planning of drivable paths in unstructured environments is heavily dependent on the degrees of freedom (DoF) of the platform. Simple robot designs offer longitudinal and rotational movements with a constant robot shape [18], [19]. For search and rescue scenarios, some robots were extended by tracked flippers [1], [2], [3]. These allow the robots to climb stairs and thus increase capabilities but also planning complexity due to additional shape shifting DoFs. Flipper positions are often not considered by the initial navigation path planning and are adjusted to the terrain in a second planning step. Platforms which offer omnidirectional locomotion increase the path planning search space by another dimension [4]. Driving is restricted, however, by terrain characteristics such as height differences and slopes which makes it not suitable for very rough terrain and for overcoming obstacles.

Legged locomotion is capable of traversing more difficult terrain since it only requires isolated feasible footholds. The drawback of this locomotion mode is, that motion planning is much more complex. Since legs are lifted from the ground repeatedly, the robot also has to constantly ensure that it remains stable. Due to the high motion complexity of stepping, path planning is often performed in at least two hierarchical levels [5], [6], [7]. A coarse planning algorithm identifies feasible footholds or areas for feasible steps. Detailed motion planning is done in a second step to connect these footholds. Navigation towards the goal is either included in the coarse planning or realized in a higher-level planner.

Since both locomotion modes have complementary advantages, it is promising to combine those. Halme et al. [9] and Takahaashi et al. [10] developed quadruped robots with legs ending in wheels. Control mechanisms to overcome obstacles are presented, but locomotion planning is not addressed. The hybrid locomotion robots HUBO [11] and CHIMP [12] were used by the winning and the third best teams at the DARPA Robotics Challenge. HUBO provides legged and wheeled locomotion, but needs to shift its shape to switch between those. Thus, hybrid locomotion, which combines advantages of both locomotion types, is not possible. CHIMP provides bipedal and quadruped hybrid locomotion. However, hybrid locomotion planning is not presented. Finally, a bipedal robot, capable of driving and walking, and a respective planning algorithm is presented by Hashimoto et al. [13]. Depending on the terrain, it either chooses walking or driving mode. A combination of both, which might bring further advantages, is not considered. Recently, Boston Dynamics introduced its biped platform Handle[1] with legs ending in wheels. It demonstrated manoeuvres which require very good dynamic control but path planning was not presented.

Our approach combines both locomotion modes in a single planning algorithm and thus has many benefits of both.

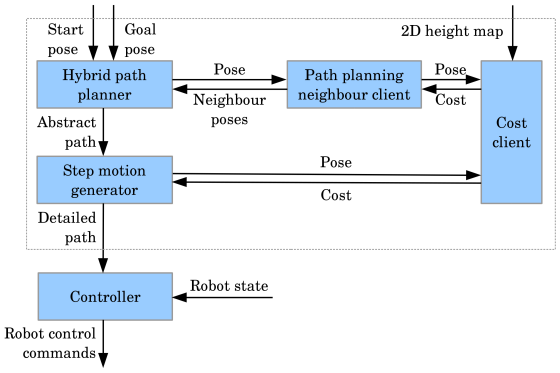

Fig. 2. Hybrid locomotion system structure: The hybrid path planner searches an abstract path from start to goal pose. The step motion generator expands this abstract path to a detailed path which can be executed by the robot. A neighbour client provides neighbour states to the planner. Both, this neighbour client and the step motion generator, request pose costs from the cost client which generates costs out of the 2D height map. The resulting path is executed by a controller.

III. Hardware

We use our quadruped robot platform Momaro [8] (see Fig. 1) with articulated legs ending in directly-driven 360° steerable pairs of wheels. Those offer omnidirectional driving and the possibility to change the robot footprint under load which neither can be done by pure driving nor by pure walking robots and enables novel movement strategies.

Each leg consists of three pitch joints which allow leg movements in the sagittal plane. Lateral leg movements are possible only passively. Legs show compliant behaviour due to their elastic carbon composite links, which work as a passive suspension system on rough terrain. Moreover, soft foam-filled wheels compensate small terrain irregularities.

A continuously rotating Velodyne Puck 3D laser scanner with spherical field-of-view at the robot head and an IMU provide measurements for terrain perception.

IV. Environment Representation

An overview of the planning system structure is given in Fig. 2. Range measurements from the 3D laser scanner are used for mapping and localization by utilizing a multiresolution surfel map [20]. Input to the planner are a 2D height map and the start and goal robot poses. A robot pose $\vec{r} = (r_x, r_y, r_\theta, \vec{f_1}, ..., \vec{f_4})$ includes the robot base position r_x, r_y and orientation r_θ and the foot positions $\vec{f_j} = (f_{j,x}, f_{j,y})$ in map coordinates. The grid resolution is set to 2.5 cm with 64 possible orientations at each position.

[1] https://youtu.be/-7xvqQeoA8c

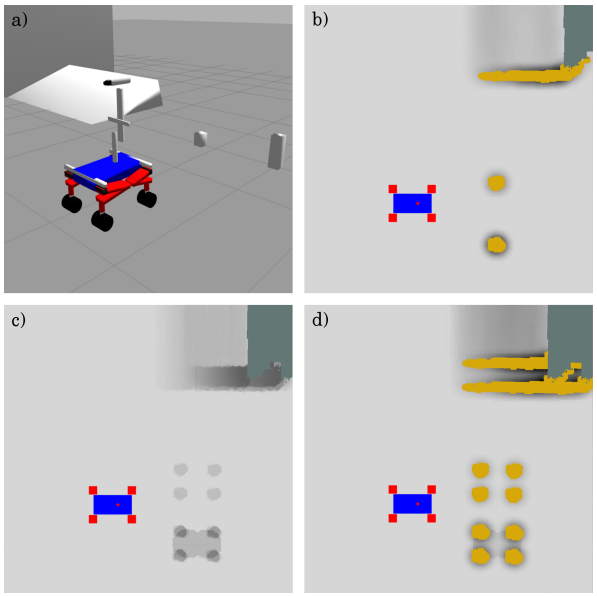

Fig. 3. Driving cost computation: a) Simulated scenario in which the robot stands in front of a ramp, a small and a tall pole. b) Foot costs. Yellow areas are not traversable by driving, olive areas are unknown. c) Body costs. The robot can take the small pole between its legs while the tall pole generates costs for lifting the robot body. Costs are shown for the current robot orientation. d) Pose costs combine body costs and foot costs at their respective positions.

The cost client computes pose cost values from the height map for a robot pose as follows: From the height map, local unsigned height differences $\triangle H$ are computed to generate the foot specific cost

$$C_{\mathrm{F}}(c_j) = 1 + k_1 \cdot \sum\nolimits_{c_\mathrm{i} \in \mathrm{map}} \triangle H(c_\mathrm{i}) \cdot w(c_\mathrm{i}) \qquad (1)$$

where $k_1 = 100$ and

$$w(c_i) = \begin{cases} \infty & \text{if } \|c_\mathrm{i} - c_j\| < r_\mathrm{F} \wedge \triangle H(c_\mathrm{i}) > 0.05, \\ 1 - \frac{\|c_\mathrm{i} - c_j\|}{r_N} & \text{if } \|c_\mathrm{i} - c_j\| < r_\mathrm{N}, \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

for a map cell c_j in which the foot $\vec{f}_j$ is located. Foot costs are assigned an infinite value if untraversable height differences $> 0.05\,\mathrm{m}$ occur in a surrounding of the size of a foot ($r_\mathrm{F} = 0.12\,\mathrm{m}$). In a neighbourhood of greater size ($r_\mathrm{N} = 0.3\,\mathrm{m}$), height differences are accumulated weighted by their distance to c_j. Foot costs are defined to be 1 in flat surroundings and increase if challenging terrain occurs. C_F includes traversability information and describes the surrounding of each foot position (see Fig. 3).

The robot shape allows obstacles to pass between the robot legs. However, if obstacles are too high they might collide with the robot base. The base cost

$$C_B(\vec{r}) = 1 + k_2 \cdot \max(H_{\mathrm{max,uB}} - H_\mathrm{B}, 0) + k_3 \cdot \triangle H_{\mathrm{max,F}}, \quad (3)$$

where $k_2 = 1$ and $k_3 = 0.5$ compares the maximum terrain height under the robot body $H_{\mathrm{max,uB}}$ with the body height H_B and assigns additional costs if the space is not sufficient. In addition, the height difference between the lowest and highest foot $\triangle H_{\mathrm{max,F}}$ generates costs since this is a measure for the terrain slope under the robot. Again, the basic cost is 1 which increases for challenging terrain. The robot base is estimated by two circles of 0.25 m radius to avoid expensive detailed collision checking.

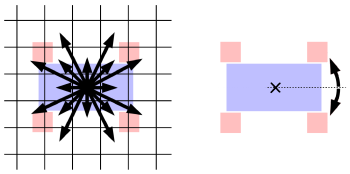

Fig. 4. Driving locomotion neighbour states can either be found by straight moves with fixed orientation within a 16-neighbourhood (l.) or by orientation changes on a fixed position (r.). Grid and orientation resolution are enlarged for better visualization.

All cost values are combined into the pose cost

$$C(\vec{r}) = k_4 \cdot \max_j(C_\mathrm{F}(\vec{f}_j)) + k_5 \cdot \sum_{j=1}^{4} C_\mathrm{F}(\vec{f}_j) + k_6 \cdot C_B(\vec{r}), \quad (4)$$

where $k_4 = 0.1$, $k_5 = 0.1$ and $k_6 = 0.5$. Pose costs are defined to be 1 on flat terrain where both, foot and body costs, induce 50% of the pose costs. We want to consider the terrain under all four wheels but want to prefer a pose with four slightly challenging contact points over a pose with three non-challenging and one very difficult contact point. Hence, it is neither sufficient to sum up all individual foot costs nor to just take the maximum. A weighted sum of both, however, achieves the desired functionality.

V. Path Planning

Path planning is done in a hybrid planner, which prefers the driving mode and considers steps only if necessary. It is realized through an A*-search on a pose grid. The used heuristic combines the Euclidean distance with orientation differences. For each pose, the path planning neighbour client provides feasible neighbouring poses (see Fig. 2). Driving neighbours can be found within a 16-neighbourhood and by turning on the spot to the next discrete orientation (Fig. 4).

As illustrated in Fig. 5, additional stepping manoeuvres are added, if a foot $\vec{f}_j$ is

- close to an obstacle $\left(\exists c \in map \ \left(C_\mathrm{F}(c) = \infty \wedge \left\|c - \vec{f}_j\right\| < 0.1\ \mathrm{m}\right)\right)$,
- a feasible foothold c_h with $C_\mathrm{F}(c_\mathrm{h}) < \infty$ can be found in front of the foot in its sagittal plane that respects a maximum leg length,
- the height difference to the foothold is small ($|H(\vec{f}_j) - H(c_\mathrm{h})| \leq 0.3\,\mathrm{m}$), and
- the distance between the two feet on the "non-stepping" robot side is $> 0.5\,\mathrm{m}$ to guarantee a safe stand while stepping.

A step is represented as an additional possible neighbouring pose for the planner.

The step which is considered by the planner at this level is an abstract step. We define an abstract step to be the direct transition from a pre-step pose to an after-step pose. It does

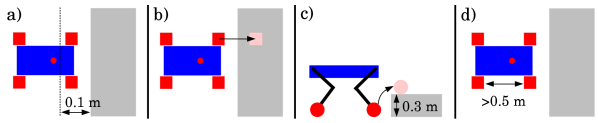

Fig. 5. Criteria to add steps to the hybrid planner: a) A foot is close to an obstacle, b) a feasible foothold can be found, c) the height difference to overcome is $\leq$ 0.3 m and d) the distance between the feet on the "non-stepping" robot side is $>$0.5 m. Grey areas show an elevated platform.

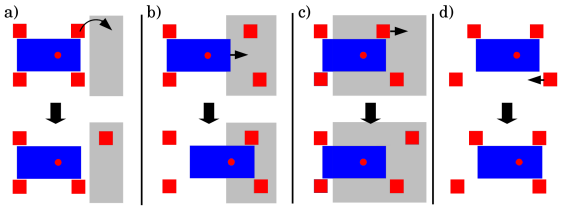

Fig. 6. Manoeuvres which extend the driving planner to a hybrid locomotion planner, visualized on a height map: a) Abstract steps, b) longitudinal base shifts, c) driving a pair of wheels at one front foot forward, and d) driving a pair of wheels at any foot back to its neutral position.

not describe the motion sequence to perform the step and needs to be expanded to be stable and executable by the robot. An abstract step is visualized in Fig. 6a.

Each step is assigned a cost value

$$C_S = k_7 \cdot L_{\text{step}} + k_8 \cdot (C_F(c_{\text{h}}) - 1) + k_9 \cdot \triangle H_{\text{step}}, \quad (5)$$

where $k_7 = 0.5$, $k_8 = 0.1$ and $k_9 = 2.3$ which includes the step length L_{step}, the foot specific terrain difficulty costs of the foothold to step in c_{h}, and the maximum height difference $\triangle H_{\text{step}}$. If multiple end positions for a step exist, only the cheapest solution is returned to the search algorithm.

Further manoeuvres are required to navigate in cluttered environments. We define the footprint shown in Fig. 5a to be the neutral footprint. It provides high robot stability at a small footprint size, and is the preferred configuration for driving.

If both front feet are positioned in front of their neutral position, the robot may perform a longitudinal base shift manoeuvre. The base is shifted forward relatively to the feet until one of the front feet reaches its neutral position or a maximum leg length is reached for one of the rear legs (see Fig. 6b). Base shifts of length L_{BS} using the average discovered base costs $C_{\text{B,avg}}$ and $k_{10} = 0.5$ carry the cost

$$C_{\text{BS}} = k_{10} \cdot L_{\text{BS}} \cdot C_{\text{B,avg}}. \quad (6)$$

If a rear foot is close to an obstacle, the pair of wheels at each front foot may be driven forward while keeping ground contact (Fig. 6c) which is a preparation for a rear foot step. If the robot footprint is not neutral, it may drive a single pair of wheels to their neutral position (Fig. 6d). A single foot movement of length L_{FM} using the average discovered foot costs $C_{\text{F,avg}}$ and $k_{11} = 0.125$ carries the cost

$$C_{\text{FM}} = k_{11} \cdot L_{\text{FM}} \cdot C_{\text{F,avg}}. \quad (7)$$

Since driving is faster and safer than stepping, we want the planner to consider drivable detours of acceptable length

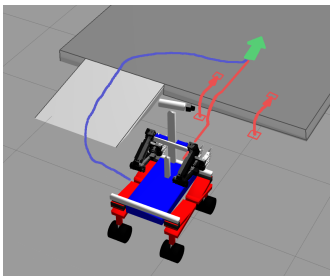

Fig. 7. Step-related manoeuvre costs are weighted such that the planner just prefers taking a 1.5 m detour over a ramp (blue path) instead of stepping up (red path) to an elevated platform.

instead of including steps in the plan. We define that, when the robot stands in front of an 0.2 m elevated platform, it should prefer a 1.5 m long detour over a ramp instead of stepping up to this platform (Fig. 7). This can be achieved by increasing the costs of stepping-related manoeuvres by a certain factor.

VI. Step Motion Generation

The result of the A* search is a cost-optimal abstract path which lacks executable motion sequences for steps and information about foot heights. The resulting path is expanded during step motion generation. It finds stable robot positions for steps and adds leg height information to the path. Again, costs are obtained from the cost client.

A. Robot Stabilization

Abstract steps only describe the start and goal poses for a stepping manoeuvre. An executable transition between these poses is a motion sequence which keeps the robot stable at all times. Such a motion sequence is generated for each abstract step in the path. Due to the compliant leg design of our robot, we have no information about the exact position of the feet but have to estimate it from actuator feedback. Hence, we limit stability considerations to static stability. Since actuator speeds are slow, dynamic effects can be neglected. Stability estimation while stepping is done on the support triangle which is spanned by the horizontal position of the remaining three feet with ground contact (Fig. 8). If the horizontal robot center of mass (CoM) projection is inside the support triangle, the robot pose is stable. The closer the CoM is to the support triangle centroid (STC), the greater the stability.

Lateral alignment of the CoM and STC is done by base roll motions (Fig. 8), which are rotations around the longitudinal axis. These are achieved by changing the leg lengths on one side of the robot. The resulting angle between the wheel axes and ground is compensated by the compliant legs and the soft-foam filled wheels. Roll manoeuvre parametrization is described in Sec. VI-B.

Longitudinal alignment of the CoM and STC is done by driving the remaining pair of wheels on the stepping side (e.g., the rear left foot if the front left foot is stepping) towards the robot center (Fig. 8). If this does not suffice because the motion is hindered by obstacles, the remaining longitudinal alignment is done by shifting the

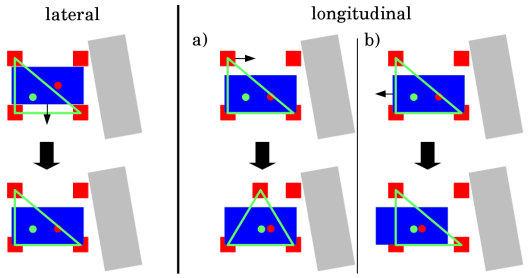

Fig. 8. To find a stable position for stepping, the robot CoM (red dot) needs to be aligned with the STC (green dot). Lateral alignment is done by base rolling. Longitudinal alignment is either done by a) driving the remaining pair of wheels on the stepping side towards the center or b) by shifting the robot base.

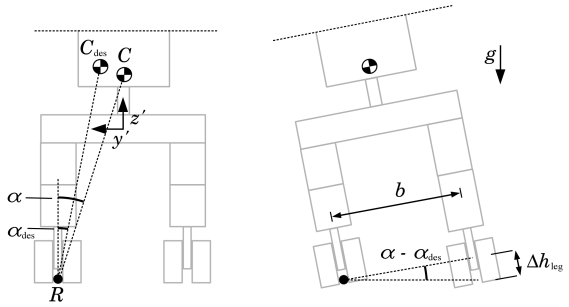

Fig. 9. Momaro's lower body in back view. Lateral CoM shifts can be achieved by changing leg length on one side which rolls the robot.

robot base. The longitudinal CoM position is also affected by the robot base pitch angle which is described in the next subsection. The presented motions generate a stable robot configuration which allows the desired step to be performed. After stepping, the robot reverses its base roll, foot displacement, and base shift manoeuvres to get back to its nominal configuration.

B. Leg Heights

Each robot pose is assigned an individual height for each leg, which describes the vertical position of a foot relative to the robot base. When driving with neutral footprint, a low leg height of 0.27 m is chosen, which provides a low CoM and good controllability through reduced leg compliance (see Fig. 1 left). A larger ground clearance is chosen for manoeuvres other than driving to provide great freedom for leg movements (see Fig. 1 right). In this case, the base height is determined by the highest foot. A soft constraint is applied that the leg height should be at least 0.45 m. At the same time, a hard constraint from the mechanical system is that none of the legs exceeds its maximum leg length. The height of each individual foot can be read from the 2D height map. The robot base pitching angle is set to be 70% of the ground slope, as can be seen in Fig. 19. This pitching value provides a good compromise between sufficient ground clearance for all four legs and a good CoM position.

As described before, base roll manoeuvres are used to shift the robot CoM laterally. Due to the soft-foam filled wheels, we can estimate the center of rotation $R(y'_{\text{rot}}, z'_{\text{rot}})$ between the two wheels (Fig. 9). In addition, the center of mass position $C(y'_{\text{CoM}}, z'_{\text{CoM}})$, the angle

$$\alpha = \arctan\left(\frac{y'_{\text{rot}} - y'_{\text{CoM}}}{z'_{\text{CoM}} - z'_{\text{rot}}}\right) \tag{8}$$

between $\vec{RC}$ and the vertical axis and the desired lateral center of mass position $y'_{\text{CoM,des}}$ are given. Using $\left\|\vec{RC}\right\| = \left\|\vec{RC}_{\text{des}}\right\|$ we compute the desired angle between $\vec{RC}_{\text{des}}$ and the vertical axis

$$\alpha_{\text{des}} = \arcsin\left(\frac{y'_{\text{rot}} - y'_{\text{CoM,des}}}{\left\|\vec{RC}\right\|}\right) \tag{9}$$

and consequently using the footprint width b we compute the desired leg height difference

$$\triangle h_{\text{leg}} = b \cdot \tan(\alpha - \alpha_{\text{des}}). \tag{10}$$

This leg height difference is added to both legs on the respective side to induce a base roll manoeuvre. Fig. 10 shows how Momaro steps up an elevated platform, using the described motion sequences.

VII. Path Planning Extensions

Due to the fine position and orientation resolution, the search space which is considered for path planning is large. Moreover, we want the planner to consider several detours before taking a step, which further increases planning times. We present methods to accelerate planning and to improve the resulting path quality. Their individual effects are investigated during evaluation.

A. Robot Orientation Cost

Although our robot is capable of omnidirectional driving, there are multiple reasons to prefer driving forward. Since the sensor setup is not only used for navigation, but also for manipulation, it is designed to provide best measurement results for the area in front of the robot. The required width clearance is minimal when driving in a longitudinal direction, which is helpful in narrow sections such as doors. The same applies to driving backwards. Driving straight backwards requires a smaller clearance than driving diagonal backwards. Finally, our leg design restricts us to perform steps in the longitudinal direction. Thus, when approaching areas where stepping is required, a suitable orientation is helpful. We address this desire of preferring special orientations by multiplying neighbour costs during A* search by the individual factor $k_{\triangle\theta}$, as described in Fig. 11.

B. ARA*

To obtain feasible paths quickly, we extend the A* algorithm to an Anytime Repairing A* (ARA*) [21]. Its initial search provides solutions with bounded suboptimality by giving the heuristic a weight >1. The result quality is then improved by decreasing the heuristic weight stepwise down to 1, if the given planning time is not exhausted yet. ARA* recycles the representations it generated previously to accelerate replanning.

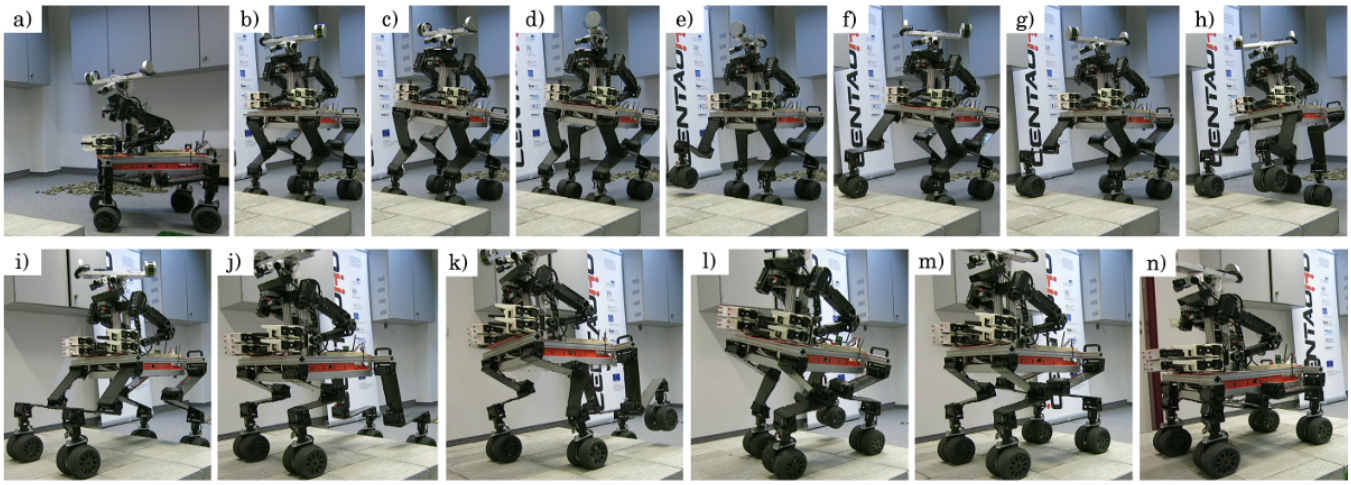

Fig. 10. Momaro stepping up an elevated platform. a) It arrives at the platform in low driving position, b) stands up, c) rolls its base to the left to shift its CoM laterally, d) drives its rear right pair of wheels forward to reach a stable stepping configuration. e) It then steps with its front right foot, f) drives its rear right pair of wheels back and g) rolls back its base to reach the configuration it had before the step. The remaining steps follow a similar motion sequence which is shown in less details. Subsequently, h) Momaro steps with its front left foot. i) It then drives forward and j) shifts its base forward before k) doing a step with the rear left and l) rear right foot. n) When the robot stands on the platform, o) it lowers its base to continue driving.

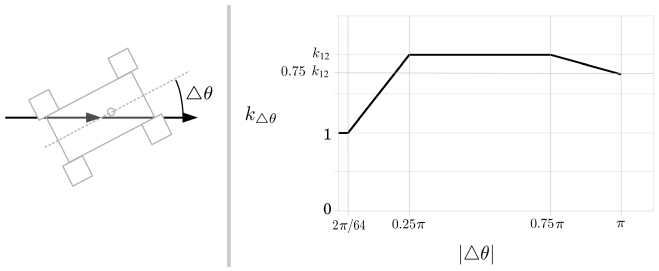

Fig. 11. For a difference between robot orientation and driving direction $\triangle\theta$ the cost factor $k_{\triangle\theta}$ is computed which expresses the desire to drive forward. It is 1 within an orientation step of $2\,\pi/60$ and increases up to k_{12}. When driving backward, there is a desire to drive straight backwards since the required clearance is smaller.

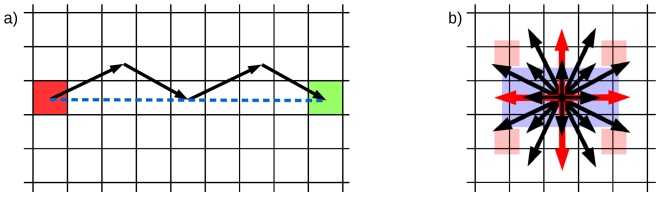

Fig. 12. Adressing ARA* preferences of long moves. a) For larger heuristic weights, the ARA* algorithm prefers those driving manoeuvres which bring the robot as close to the goal as possible which leads to undesired paths (black arrows). b) To obtain the desired behaviour (blue line) we extend the driving neighbourhood by the four red manoeuvres.

When planning with higher weighted heuristics, the planner prefers those driving manoeuvres that bring it as close to the goal as possible. This leads to the undesired effect that resulting paths mainly consist of those driving manoeuvres which go two cells in one direction and one cell in an orthogonal direction, as can be seen in Fig. 12 a. To prevent this behaviour, we extend the driving neighbourhood size from 16 to 20, as shown in Fig. 12 b.

VIII. Plan Execution

We utilize the control framework described by Schwarz et al. [15]. Input for omnidirectional driving is a velocity command $\vec{w} = (v_{x'}, v_{y'}, \omega)$ with horizontal linear velocities $v_{x'}$ and $v_{y'}$ in robot coordinates and a rotational velocity ω around the vertical robot axis. We obtain $\vec{w}$ by computing a B-spline through the next five driving poses and aim towards a pose $\vec{p} = (p_x, p_y, p_\theta)$ on this B-spline in front of the robot.

For manoeuvres which require leg movement, the input to the control framework are 2D (x', z') foot poses which can be directly derived from the resulting path.

IX. Experiments

We evaluate our path planning method and the presented extensions in the Gazebo simulation environment[2]. Experiments are done on one core of a 2.6 GHz Intel i7-6700HQ processor using 16 GB of memory. A video of the experiments is available online[3].

In a first scenario, the robot stands in a corridor in front of an elevated platform and some cluttered obstacles. It needs to find a way to a goal pose on this platform as can be seen in Fig. 13. We compare the performance of the planner for different values of k_{12} in Fig. 14. The parameter k_{12} is defined in Fig. 11. All shown path costs are the costs the path would have in the plain A* planner to keep them comparable. It can be seen that an increasing robot orientation cost factor decreases the difference between robot orientation and driving direction. This can also be observed in Fig. 16. Moreover, planning is accelerated for higher values (≥ 2) of k_{12}, while path costs increase only slightly.

In addition, we evaluate the ARA* approach in the same scenario. We choose exponentially decaying heuristic

[2] http://www.gazebosim.org

[3] https://www.ais.uni-bonn.de/videos/IROS_2017_Klamt/

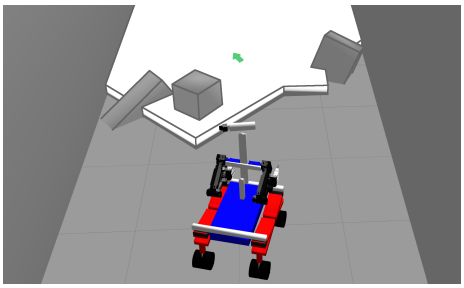

Fig. 13. Gazebo scenario to compare planner variants. Momaro stands in front of an elevated platform, cluttered with obstacles and has to reach a pose on this platform.

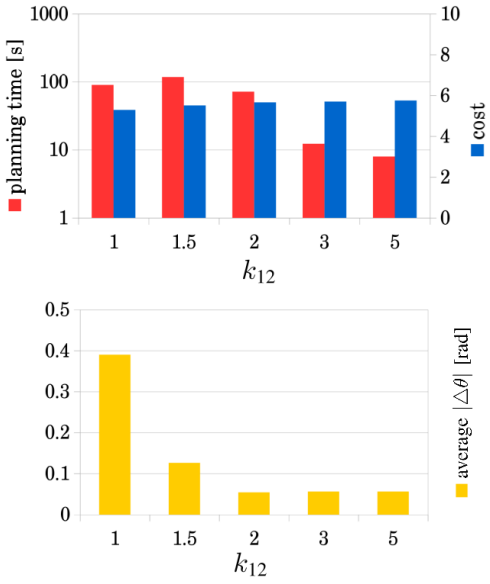

Fig. 14. Comparison between the original A* planner ($k_{12} = 1$) and the modification with robot orientation cost factor.

weights, starting at 3.0 while k_{12} = 2. The performance results are shown in Fig. 15 and a path can be seen in Fig. 16. ARA* provides its first result in 32 ms, and this is 31% more costly than the optimal solution. A solution with only 2% higher costs is found in ~10 s which is sufficiently fast compared to the required execution times. Planning paths using an optimal heuristic weight takes infeasibly long ($> 100\,s$). Searching optimal results takes longer than in the plain A* variant because higher heuristic weights are considered first and the neighbourhood size changed from 16 to 20. It can be seen that the effect of the robot orientation cost factor increases with decreasing heuristic weights.

To demonstrate the capabilities of our planner, we present a second experiment in which Momaro has to climb a staircase which is blocked by obstacles (Fig. 17). Tracked vehicles would have great difficulties to overcome this. Our robot climbs the stairs and then drives sideways while taking the obstacle between its legs. Our planner finds a first path with heuristic weight of 3.0 in 1.02 s. Fig. 19 shows Momaro on the staircase and visualizes how the robot base adapts its pitch angle to the terrain slope.

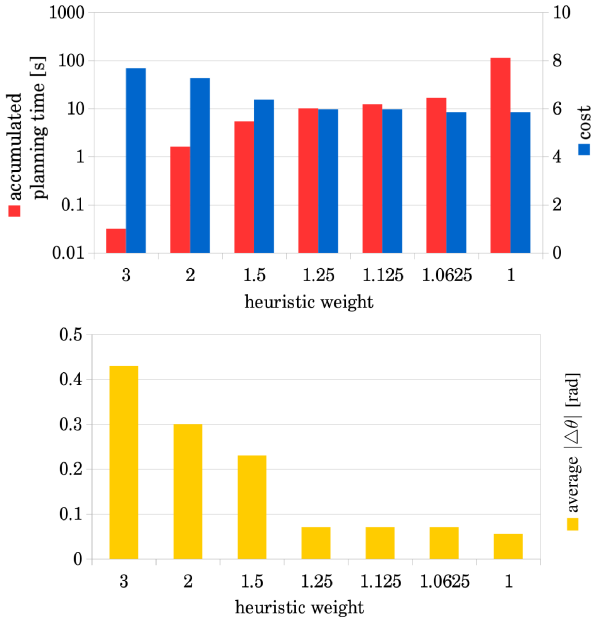

Fig. 15. Performance and result quality of the ARA* algorithm where $k_{12} = 2$.

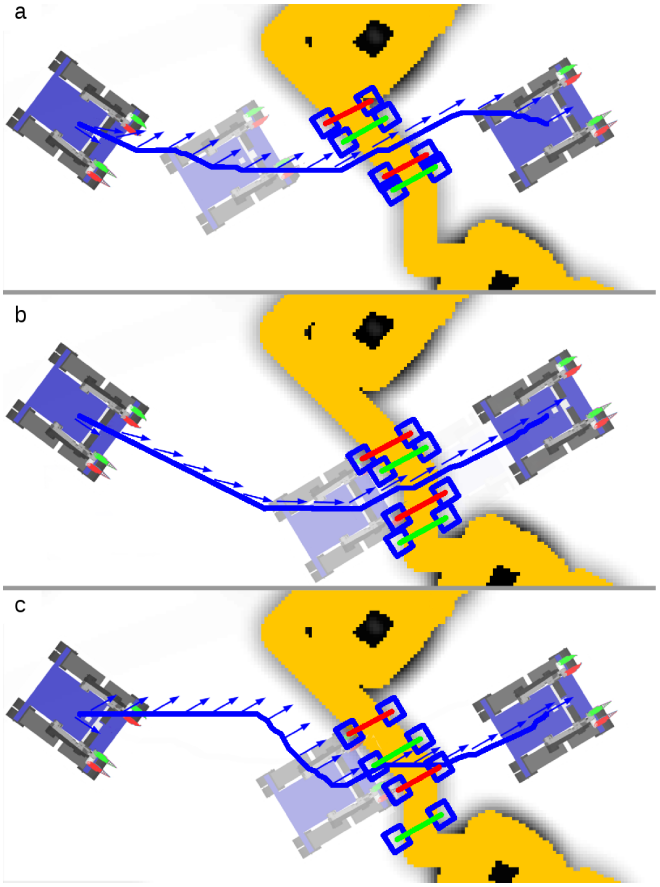

Fig. 16. Resulting paths of our planner in different settings on a foot cost map. Yellow areas are not traversable by driving. Blue paths show the robot center position, arrows show the orientation. Blue rectangles show used footholds. Red lines represent front foot steps, green lines represent rear foot steps. a) Result of the plain A* algorithm, b) orientation differences are considered ($k_{12} = 2$), c) first result of the ARA* algorithm using a heuristic weight of 3.0.

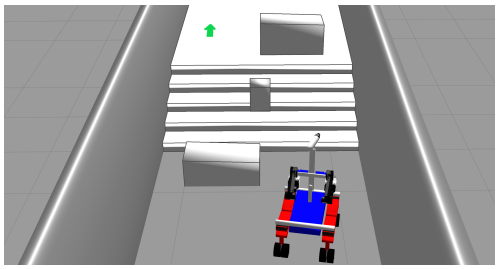

Fig. 17. Challenging scenario to demonstrate the planner capabilities. A staircase with obstacles on it requires a combination of stepping and driving sideways.

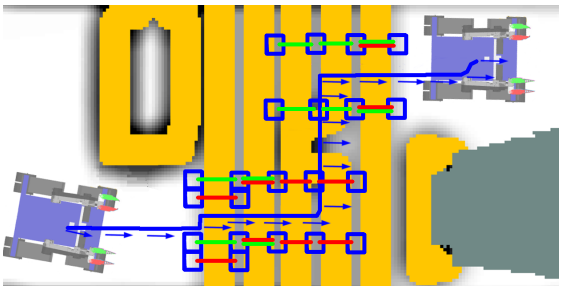

Fig. 18. Planner output for the staircase scenario on a foot cost map using a heuristic weight of 3. The blue path shows the robot center position. Arrows show the robot orientation. Blue squares show used footholds. Red lines represent front foot steps, green lines represent rear foot steps.

X. Conclusion

In this paper, we presented a hybrid locomotion planning approach which combines driving and stepping in a single planner. It provides paths with bounded suboptimality in feasible time and is capable of path planning in challenging environments. Due to the high dimensionality of the search space and the desire to consider detours instead of stepping, finding optimal solutions may take considerable time. We address this by using an anytime approach with larger heuristic weights. The planned paths are executed by our mobile manipulation robot Momaro. Experiments demonstrated that our method generates paths for challenging terrain, which could not be traversed by driving or stepping alone.

References

[1] F. Colas, S. Mahesh, F. Pomerleau, M. Liu, and R. Siegwart, "3D path planning and execution for search and rescue ground robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[2] M. Menna, M. Gianni, F. Ferri, and F. Pirri, "Real-time autonomous 3D navigation for tracked vehicles in rescue environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[3] M. Brunner, B. Brüggemann, and D. Schulz, "Motion planning for actively reconfigurable mobile robots in search and rescue scenarios," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2012.

[4] Z. Ziaei, R. Oftadeh, and J. Mattila, "Global path planning with obstacle avoidance for omnidirectional mobile robot using overhead camera," in *IEEE International Conference on Mechatronics and Automation*, 2014.

[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.

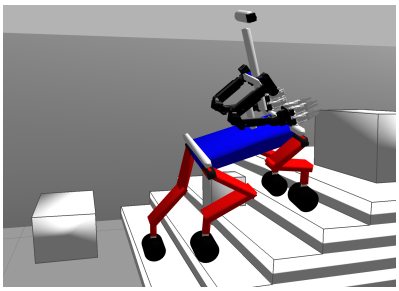

Fig. 19. Momaro climbing stairs. The robot base pitch angle adapts to 70% of the terrain slope.

[6] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[7] 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.

[8] M. Schwarz, T. Rodehutskors, M. Schreiber, and S. Behnke, "Hybrid driving-stepping locomotion with the wheeled-legged robot Momaro," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[9] A. Halme, I. Leppänen, J. Suomela, S. Ylönen, and I. Kettunen, "Workpartner: Interactive human-like service robot for outdoor applications," *The international journal of robotics Research*, vol. 22, no. 7-8, pp. 627–640, 2003.

[10] M Takahaashi, K. Yoneda, and S. Hirose, "Rough terrain locomotion of a leg-wheel hybrid quadruped robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[11] H. Bae, I. Lee, T. Jung, and J.-H. Oh, "Walking-wheeling dual mode strategy for humanoid robot, DRC-HUBO+," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[12] A. Stentz, H. Herman, A. Kelly, E. Meyhofer, G. C. Haynes, D. Stager, B. Zajac, J. A. Bagnell, J. Brindza, C. Dellin, *et al.*, "Chimp, the CMU highly intelligent mobile platform," *Journal of Field Robotics*, vol. 32, no. 2, pp. 209–228, 2015.

[13] K. Hashimoto, T. Hosobata, Y. Sugahara, Y. Mikuriya, H. Sunazuka, M. Kawase, H.-o. Lim, and A. Takanishi, "Realization by biped leg-wheeled robot of biped walking and wheel-driven locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[14] M. Schwarz, M. Beul, D. Droeschel, S. Schüller, A. S. Periyasamy, C. Lenz, M. Schreiber, and S. Behnke, "Supervised autonomy for exploration and mobile manipulation in rough terrain with a centaur-like robot," *Frontiers in Robotics and AI*, vol. 3, 2016.

[15] M. Schwarz, T. Rodehutskors, D. Droeschel, M. Beul, M. Schreiber, N. Araslanov, I. Ivanov, C. Lenz, J. Razlaw, S. Schüller, D. Schwarz, A. Topalidou-Kyniazopoulou, and S. Behnke, "NimbRo Rescue: Solving disaster-response tasks through mobile manipulation robot Momaro," *Journal of Field Robotics*, vol. 34, no. 2, pp. 400–425, 2016.

[16] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[17] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[18] B. P. Gerkey and K. Konolige, "Planning and control in unstructured terrain," in *ICRA Workshop on Path Planning on Costmaps*, 2008.

[19] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 141–166, 2007.

[20] D. Droeschel, M. Schwarz, and S. Behnke, "Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner," *Robotics and Autonomous Systems*, vol. 88, pp. 104–115, 2017.

[21] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara*: Anytime A* with provable bounds on sub-optimality.," in *NIPS*, 2003.