

TAMOLS: Terrain-Aware Motion Optimization for Legged Systems

Fabian Jenelten , Ruben Grandia , Farbod Farshidian , and Marco Hutter 

Abstract—Terrain geometry is, in general, nonsmooth, nonlinear, nonconvex, and, if perceived through a robot-centric visual unit, appears partially occluded and noisy. This article presents the complete control pipeline capable of handling the aforementioned problems in real-time. We formulate a trajectory optimization problem that jointly optimizes over the base pose and footholds, subject to a height map. To avoid converging into undesirable local optima, we deploy a graduated optimization technique. We embed a compact, contact-force free stability criterion that is compatible with the nonflat ground formulation. Direct collocation is used as transcription method, resulting in a nonlinear optimization problem that can be solved online in less than ten milliseconds. To increase robustness in the presence of external disturbances, we close the tracking loop with a momentum observer. Our experiments demonstrate stair climbing, walking on stepping stones, and over gaps, utilizing various dynamic gaits.

Index Terms—Legged robots, motion control, optimization and optimal control, perceptive locomotion.

I. INTRODUCTION

LEGGED locomotion has been studied and designed for the last couple of decades. Recent advances in both software and hardware have triggered the transition from experimental platforms used under laboratory conditions to (semi)autonomous machines deployed in real-world scenarios, e.g., on industrial sites for inspection [1] or in underground mines for exploration and mapping [2]. Yet, assumptions made in “classical” control approaches limit applications to flat or mildly rough ground or restrict locomotion to static stability. However, the true potential of legged locomotion is undeniably rooted in a combination of rough environments and dynamic agility. More recent control approaches have eliminated both the flat-ground and static-gait restrictions but struggle to match the computational overhead with the onboard compute budget.

Manuscript received 10 March 2022; revised 1 June 2022; accepted 23 June 2022. This work was supported in part by the Swiss National Science Foundation (SNSF) under Project 188596, in part by the European Union’s Horizon 2020 Research and Innovation Programme under Grant 780883 and Grant 101016970, and in part by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics (NCCR Robotics). This article was recommended for publication by Associate Editor C. Semini and Editor E. Yoshida upon evaluation of the reviewers’ comments. (*Corresponding author: Fabian Jenelten.*)

The authors are with Robotic Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland (e-mail: fabian.jenelten@mavt.ethz.ch; rgrandia@ethz.ch; farshidian@mavt.ethz.ch; mahutter@ethz.ch).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TRO.2022.3186804>.

Digital Object Identifier 10.1109/TRO.2022.3186804

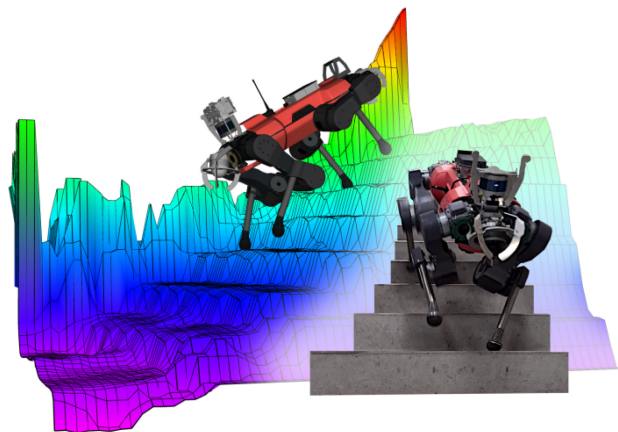


Fig. 1. Right: ANYmal [3] is walking down a staircase using an ambling gait. Left: Corresponding gazebo model visualization with the elevation map.

When it comes to rough terrain locomotion, we can isolate the following three major challenges.

- 1) Low computation time and generalizability over various terrains are contradicting requirements. Therefore, it is essential to strike a balance between simplicity and complexity to fully exploit perceptive feedback.
- 2) When embedding the true terrain geometry, many local optima might appear, rendering a motion optimization problem sensitive to the initial guess. If the initializing motion is located on the “wrong” side of the cost function valley, the solution might converge into an undesirable local optimum.
- 3) Sensors used to generate perceptive data are mounted onboard, and large parts of the field of view might appear occluded. These occluded regions can significantly shrink the feasible space or lead to unreliable foot placement.

In this work, we address each of the three points individually, aiming to further close the gap between blind and perceptive locomotion.

A. Contributions

The main contributions of this work can be split into the following four independent parts.

- 1) First, we derive and analyze a contact-force free and fully differentiable dynamic stability metric. Compared to single rigid body dynamics (SRBD), our model eliminates the contact forces and has thus a smaller problem size; as opposed to contact wrench models, we do not depend on algorithmic construction of the feasible set; and compared

to zero-moment point (ZMP) constraints, our model generalizes to uneven ground with vertical contact planes.

- 2) We present a trajectory optimization (TO) framework, capable of simultaneously optimizing over footholds and base pose, that can be solved at frequencies larger than 100 Hz. Instead of handcrafting a reasonable initial guess, we rely on graduated optimization, a heuristic tool primarily used in computer vision for dealing with nonconvex and nonsmooth problems [4].
- 3) We introduce a mapping pipeline and show how to extrapolate missing data in the map and how to remove artifacts and noise.
- 4) Finally, we improve the inverse dynamics-based tracking loop by embedding a disturbance observer.

B. Outline

In the remainder of this section, we review different model-based TO methods and categorize them according to complexity and decomposition of structure. Guided by this discussion, we introduce and justify design choices and model assumptions in Section II, forming the basis of our control pipeline. What follows in Section III is the development of a dynamic stability criterion. Mapping and map processing are carried out all onboard. Related details can be found in Section IV. In the following Section V, we present our approach to terrain-aware motion optimization for legged system, hereinafter referred to as TAMOLS. In Section VI, we present changes to the widely used whole body control scheme. The overall locomotion controller is evaluated on the quadrupedal robot ANYmal (see Fig. 1) in Section VII. Finally, Section VIII concludes this article.

C. Literature Review

1) *Decomposition of Structure*: A first class of locomotion controllers, in this work termed *modular motion optimization (MMO)*, decomposes motion optimization into several modules. The number of modules and the module hierarchy may vary, but the separation of foothold and base pose planning is a necessary characteristics [5]–[15]. The decoupling allows to simplify foothold selection in mapped environments, e.g., by iterating over a foot-score map and checking each grid cell for optimality. An often encountered concept, called foothold adaptation, limits the search space to the local vicinity of a nominal foothold, which is selected under blind conditions [12]–[15]. Computation durations of such planners are low, but the lack of a true kinematic model can lead to infeasible base motions.

As opposed to foothold adaptation, other work has focused on pattern generation. Similar approaches lay out a pattern of footholds along a height map, predicting touch-down locations over several strides. Scanning and map generation are either outsourced as offline tasks [16]–[18], or the maps are generated onboard [19], [20]. Optimization duration ranges from several tenths of a second to several seconds, rendering continuous replanning difficult on fast walking robots.

Locomotion planners characterized by a *combined motion optimization (CMO)* can guarantee the existence of kinematically consistent trajectories. In combination with automatic gait discovery, this second class of locomotion controllers typically suffers from large computation times of several seconds

to minutes [21]–[24]. When fixing the gait, these TO problems have shown to be solvable in less than a second [25] or less than 100 ms [26], [27]. There exist also CMO frameworks that optimize over the gait and terminate in several hundreds of milliseconds [28] or even less [29], [30]. Extending these formulations with perceptive information while keeping the optimization fast and reliable is still ongoing research.

2) *Perceptive Locomotion*: Most work in the field of perceptive locomotion is inspired by the early development of Kolter *et al.* [16]. Footsteps are found using a greedy search over a foot-cost map along an approximate body path. The latter is subsequently refined offline by planning a statically stable center of mass (COM) trajectory. The controller's performance was evaluated with *LittleDog* on four different terrains of varying difficulty. Kalakrishnan *et al.* [17] proposed several extensions to that approach; the most notable one is online COM planning subject to ZMP stability. Both methods rely on accurately prescanned terrains and preprocessed height maps.

Throughout the following years, related MMO approaches have been pushed toward full onboard motion planning. Mastalli *et al.* [19] achieved a replanning frequency of 0.5 Hz for the foothold computation. The closed-loop pipeline was evaluated with *HyQ*, demonstrating crawling over stacked pallets and stepping stones. The key to fast convergence is to restrict the search space to an approximate body path plan. Griffin *et al.* [20] suggested a convex decomposition of the environment and introduced an A^* -footstep planner. It also accounts for yaw orientation of the feet and converges in less than 2 s. Performance was verified with the biped robots *Atlas* and *Valkyrie* on a set of different terrains.

Following the MMO structure, Fankhauser *et al.* [12] introduced a terrain-aware locomotion pipeline, where footholds are selected from a binary foothold-score map. The search space is limited to the local surroundings of a priori computed nominal foothold. Using *ANYmal*, the authors showed statically stable rough terrain locomotion.

Magana *et al.* [13] presented a self-supervised foothold classifier. The convolutional neural network learns a correction step based on a nominal foothold. The approach was verified on *HyQ*, demonstrating a trot over gaps and pallets. This approach was further extended by Jenelten *et al.* [14] using a batch search, showing dynamic stair-climbing with *ANYmal*.

Gangapurwala *et al.* [31] employed a learning-based policy to plan perceptive footholds, tracked within a model-based MMO environment. Robustness of the tracking controller has been improved by learning corrective joint torques. The combined framework has been tested with *ANYmal* on slopes and bricks, utilizing a trotting and crawling gait.

The more complex CMO formulations are on a transition to online-perceptive locomotion. Mastalli *et al.* [32] presented a method where the motion is subject to ZMP-stability, and the footholds are constrained to a terrain cost map. The stochastic search leads to optimization durations of several minutes, and hence, the TO has to be carried out offline.

First successful results in generalizing kino-dynamic model predictive control (MPC) has been shown by Grandia *et al.* [33]. *ANYmal* was demonstrated to trot over dense stepping stones with the help of a premapped and presegmented terrain.

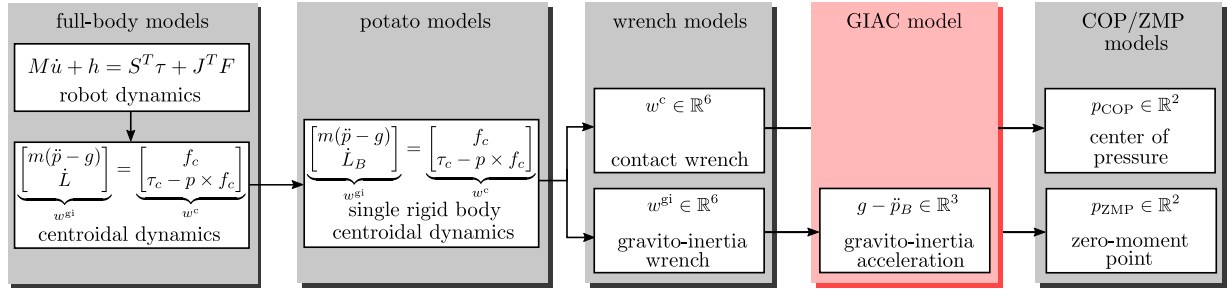


Fig. 2. Illustration of dynamic models following the order of regressive complexity (or dimensionality) from left to right. The whole-body dynamics represent the robot as a fully articulated structure. The governing equations of motion (EOM) are obtained by Lagrangian mechanics. The first six rows are called the centroidal dynamics (CD) and coincide with the Newton–Euler EOM. It relates variations of the centroidal momentum with the sum of contact forces $f_c = \sum f_i$ and momenta $\tau_c = \sum p_i \times f_i$. The SRBD account for only one rigid body, and the angular centroidal momentum L simplifies to the angular momentum of the base L_B . Gravito-inertia/contact wrench models consider only the left/right-hand side of the CD. The GIAC model, which will be presented in this work, only depends on the gravito-inertia acceleration and footholds. ZMP/COP models are the projection of the associated wrench onto the ground or some virtual plane.

In the work of Melon *et al.* [34], the CMO pipeline proposed by Winkler *et al.* [24] has been improved with a learning-based initializer. In the experiments performed with ANYmal, the environment was loaded from a virtual model. Most recently, this approach was shown in combination with online mapping, enabling ANYmal to walk over 20 cm high steps [28]. Despite promising results, the replanning frequency was reported as not larger than 3.3 Hz in the most favorable case.

3) *Dynamic Model*: The formulation of a TO problem requires a physical model at hand. We have categorized these models based on their complexity as in the following and as summarized in Fig. 2.

TO has been used in combination with the *whole-body dynamics* (WBD) of legged robots. These methods can directly constrain joint torques. Due to excessive model complexity, optimization times usually peak at several minutes [22], [23] or seconds [21], [25], but can be as low as several tenths of a millisecond [29], [30].

The *centroidal dynamics* (CD) are the projection of the WBD to the COM [35]. It has been used in TO methods for bipedal [36], [37] and quadrupedal systems [38], [39].

The joint space states can be eliminated by assuming that the robot’s inertia remains constant. These models are commonly referred to as *SRBD*. Related methods can integrate kinematic constraints formulated in joint [26] or task space [24], [28], [34]. A popular approach is to linearize the dynamics around a nominal operation point, e.g., the measured state [40], steady state [8], [27], [41], or desired state [42].

The CD or SRBD can be further decomposed into two parts, the *contact wrench* (CW) and the *gravito-inertia wrench* (GIW) [43], separating contact forces from spatial acceleration. The GIW is typically used for planning where it is subject to centroidal or contact wrench cone (CWC) feasibility. Due to the absence of contact forces, this stability criterion scales better with the number of contacts [44]. To the authors’ best knowledge, methods of this category have only been used in combination with prefixed footholds [9]–[11], [45], [46], for which the CWC can be precomputed as a conic polyhedron.

Projecting the GIW or the CW onto the ground [47] or a virtual plane [43] leads to the definition of the so-called *zero-moment point* (ZMP) or *center of pressure* (COP), respectively. Both have different meanings but are equivalent on a physical basis.

Under the same transformation, the CWC projects to a support polygon, defined as the convex hull of (virtual) contact positions. These reduced models represent some kind of inverted pendulum with a single virtual contact, from which the most popular one is the linear inverted pendulum mode [5], [6], [17]. If we further assume zero acceleration, the ZMP/center of pressure (COP) simplifies to the projected COM [12], [16]. The associated stability criterion is of pure static nature.

II. OVERVIEW

A. Design Choices and Model Assumptions

1) *Decomposition of Structure*: In rough terrains, the optimal motion is often found at kinematic limits. The absence of kinematic constraints in the foothold selection step renders MMO susceptible to leg over-extension. Footholds might be located too far from each other, potentially reducing the feasible space of the base pose trajectory to the empty set. On the other hand, if kinematic constraints are not taken into account, the limbs may be exposed to singular configurations where tracking performance degrades [14]. When combining base pose and footholds into one single TO problem, we can directly constrain limb extension and thereby enforce a feasible task space configuration without the need for an exact kinematic model. In this work, the focus is put on such a CMO method.

The gait pattern defines lift-off and touch-down timings of the feet over a periodic time interval. As a second design choice, our approach uses a fixed gait pattern. This is not a necessary assumption, but it reduces the problem size and thus the expected optimization time.

2) *Dynamic Stability Criterion*: The literature review suggests a correlation between model complexity and optimization duration. In favor of the latter, we prefer the model to be located on the far right side of Fig. 2.

In the case of ANYmal, the four limbs make up for half of the robot’s total weight. Two third of the limb mass is accumulated in the hip abduction/adduction (HAA) and hip flexion/extension (HFE) joints, forming an approximate revolute joint. Its mass is located close to the torso, which makes the SRBD a plausible model for our robot.

As a next simplification, we might consider only the total wrench generated by the contact forces and the CWC that it

belongs to. Mathematically, such a cone is formulated as the Minkowski sum of translated friction cones, for which there exists no closed-form solution. Numerical tools exist, such as the double-descriptor method (DDM) [48]. Unfortunately, the algorithm finds an expression that is nondifferentiable w.r.t. footholds, and thereby, wrench models can not be used in our case. A brief introduction into CWC stability is provided in Appendix A-C.

The simpler ZMP/COP models assume flat ground and are by design not representative in rough environments.

We conclude that the least complex design that suffices the requirements for perceptive locomotion is based on SRBD. In the subsequent section, we will introduce model assumptions that allow us to eliminate contact forces while maintaining differentiability w.r.t. all unknowns.

B. Nomenclature

If not stated differently, vectors are expressed in *world frame* W , defined by the orthogonal unit vectors $\{e_x, e_y, e_z\}$ and the origin $\mathcal{O}_W$. The robot-centric *base frame* B is attached to the geometric center of the base. We use x^B notation to refer to a vector given in local coordinates B .

We denote a foothold and a contact force of leg i by p_i and f_i , respectively, with $i = 1, \dots, N$ and N the number of grounded legs. A difference between two footholds i and j is written as $p_{ij} = p_j - p_i$. In case of a quadrupedal robot, the four legs are labeled left front (LF) ($i = 1$), right front (RF) ($i = 2$), left hind (LH) ($i = 3$), and right hind (RH) ($i = 4$).

For the base pose, we use the expression $\Pi_B = \begin{bmatrix} p_B & \phi_B \end{bmatrix}$

where $p_B = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is the base position in world frame and $\phi_B = \begin{bmatrix} \psi & \theta & \varphi \end{bmatrix}^T$ are the ZYX-Euler angles of the base w.r.t. world. We write $R_B = R_B(\phi_B)$ to indicate a rotation matrix that rotates a vector expressed in base coordinates to world frame. For reducing index complicity, we assume that the base center is identical to the COM of the base.

We compute linear and angular momentum of the base as $P_B = m\dot{p}_B$ and $L_B = R_B(I_B\omega_B^B)$. The vector $\omega_B^B = \omega_B^B(\phi_B, \dot{\phi}_B)$ is the angular velocity of the base around the base frame, m is the total mass, and I_B the inertia of the robot in base frame computed at nominal configuration.

The gravity vector is written as $g = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$ with $g = 9.81 \text{ m/s}^2$. The symbol μ is used to denote a friction coefficient. We define a wrench $w := \{f, \tau\}$ as the tuple consisting of a force and moment. The *gravito-inertia acceleration (GIA)* vector is defined as $a_B := g - \ddot{p}_B$. Finally, we use the determinant notation $\det(a, b, c)$ to express the scalar triple product $a \cdot (b \times c)$.

C. Control Structure

The control architecture used in this work is outlined in Fig. 3. Generalized positions $\hat{q}$, generalized velocities $\hat{u}$, and torques $\hat{\tau}$ are estimated/measured at 400 Hz. The point clouds gathered from two onboard LiDARs are processed to different elevation maps $\{h, h_{s1}, h_{s2}\}$ at a rate of 20 Hz. The three elevation layers are tightly embedded in the TO problem and the

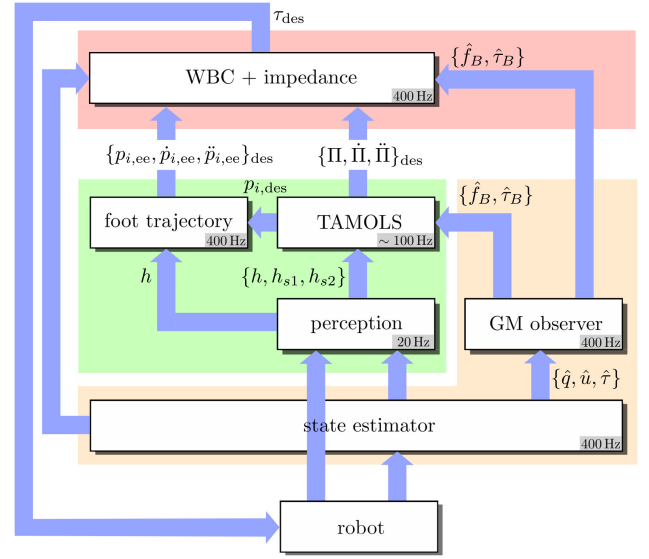


Fig. 3. Overview of the control structure, consisting of trajectory generation (green), tracking (red), and state/disturbance observer (orange).

swing trajectory generation. These two modules output desired reference signals for the base $\{\Pi_B, \dot{\Pi}_B, \ddot{\Pi}_B\}_{des}$ and swinging feet $\{p_{i,ee}, \dot{p}_{i,ee}, \ddot{p}_{i,ee}\}_{des}$, which are tracked by a whole-body controller (WBC) in task space at 400 Hz. External disturbances of the base $\{\hat{f}_B, \hat{\tau}_B\}$, estimated by a generalized momentum (GM) observer, are compensated in the tracking controller and in the motion optimizer.

III. GIAC MODEL

We first recapitulate the SRBD in Section III-A, and show how to eliminate the contact forces in Section III-C, given the assumptions to be introduced in Section III-B. The resulting model depends on the contact state and is summarized in Section III-D. The remaining Section III-E discusses its inherent geometric and physical properties.

A. Background

A legged robot can be modeled as a floating body that balances contact forces through the rate of change of linear and angular momentum

$$m \underbrace{\begin{bmatrix} \ddot{p}_B - g \\ m p_B \times (\ddot{p}_B - g) \end{bmatrix}}_{:= -w^{gi}} + \underbrace{\begin{bmatrix} 0 \\ \dot{L} \end{bmatrix}}_{:= w^c} = \sum_{i=1}^N \underbrace{\begin{bmatrix} f_i \\ p_i \times f_i \end{bmatrix}}_{:= w^c}. \quad (1)$$

The left/right-hand side is called the gravito-inertia/contact wrench, while the combination of both is known as the CD. If the limb masses are negligible, then the angular momentum L simplifies to the angular momentum of the base L_B , and the resulting equation is called SRBD.

We are not only interested in motions that are consistent with physics, but that also support nonsliding contacts. If friction is modeled using Coulomb's friction law, such a requirement can be formulated as

$$f_i \in \mathcal{F}_i, \quad \forall i = 1, \dots, N. \quad (2)$$

The set $\mathcal{F}_i$ is called the *friction cone* and is defined by the plane normal $\mathbf{n}_i$ and friction coefficient μ_i at contact i

$$\mathcal{F}_i = \{\mathbf{f}_i \mid \mu_i \mathbf{n}_i^T \cdot \mathbf{f}_i \geq \|(\mathbb{I}_{3 \times 3} - \mathbf{n}_i \mathbf{n}_i^T) \mathbf{f}_i\|\}. \quad (3)$$

B. Model Assumptions

We start the mathematical derivation by introducing the following set of assumptions.

Assumption 1 (Single body): The limbs have zero mass.

Assumption 2: The rate of change of the angular momentum has a negligible effect on the contact forces.

Assumption 3: Contact forces can only push on the ground.

Assumption 4 (Coulomb friction): Friction coefficients are constant over time and identical for all legs.

Assumption 5: Contacts are established on horizontal planes.

Assumption 6 (Overhanging torso): The base position is located above all the grounded feet, i.e., $\mathbf{e}_z^T \cdot \mathbf{p}_B > \mathbf{e}_z^T \cdot \mathbf{p}_i \forall$ grounded i .

C. Derivation

By summing up the contact forces in (2), we obtain a condition on the GIA vector

$$\sum_{i=1}^N \mathbf{f}_i = m(\ddot{\mathbf{p}}_B - \mathbf{g}) \Rightarrow \ddot{\mathbf{p}}_B - \mathbf{g} \in \mathcal{F}. \quad (4)$$

The set $\mathcal{F}$ is obtained as the Minkowski sum of local friction cones $\{\mathcal{F}_1, \dots, \mathcal{F}_N\}$. Assumptions 3–5 lead to a relaxed scenario, in which an analytical expression of the friction cone can be found as

$$\mu \mathbf{e}_z^T \cdot (\mathbf{g} - \ddot{\mathbf{p}}_B) \geq \|(\mathbb{I}_{3 \times 3} - \mathbf{e}_z \mathbf{e}_z^T)(\mathbf{g} - \ddot{\mathbf{p}}_B)\|. \quad (5)$$

Assumption 2 guarantees that friction constraints on the linear momentum are sufficient for nonsliding contacts.

Without loss of generality, we exploit assumption 1, i.e., $\dot{\mathbf{L}} = \dot{\mathbf{L}}_B$, and rewrite the system dynamics as

$$m \begin{bmatrix} \ddot{\mathbf{p}}_B - \mathbf{g} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{L}}_B \end{bmatrix} = \sum_{i=1}^N \begin{bmatrix} \mathbf{f}_i \\ (\mathbf{p}_i - \mathbf{p}_B) \times \mathbf{f}_i \end{bmatrix} \quad (6a)$$

$$\mathbf{f}_i \in \mathcal{F}_i \quad \forall i \in \{1, \dots, N\}. \quad (6b)$$

In the following, we show how to eliminate the contact forces from (6a) and (6b) by using Assumptions 5 and 6.

1) *Three or More Grounded Legs:* If the number of contact locations is $N > 1$, then we can solve the first row of (6a) for $\mathbf{f}_1$ and substitute into the second row

$$m(\mathbf{p}_1 - \mathbf{p}_B) \times (\ddot{\mathbf{p}}_B - \mathbf{g}) + \sum_{i \neq 1} (\mathbf{p}_i - \mathbf{p}_1) \times \mathbf{f}_i = \dot{\mathbf{L}}_B. \quad (7)$$

Next, we multiply (7) from the left with $\mathbf{p}_{12} = \mathbf{p}_2 - \mathbf{p}_1$

$$m \mathbf{p}_{12}^T \cdot [(\mathbf{p}_1 - \mathbf{p}_B)^T \times (\ddot{\mathbf{p}}_B - \mathbf{g})] + \mathbf{p}_{12}^T \cdot \sum_{i \neq 1, 2} (\mathbf{p}_i - \mathbf{p}_1) \times \mathbf{f}_i = \mathbf{p}_{12}^T \cdot \dot{\mathbf{L}}_B. \quad (8)$$

Using the determinant notation and the circular shift property of the scalar triple product, the equation can be simplified to

$$\det(\mathbf{p}_{12}, \mathbf{p}_B - \mathbf{p}_1, \mathbf{g} - \ddot{\mathbf{p}}_B) - \mathbf{p}_{12}^T \cdot \frac{\dot{\mathbf{L}}_B}{m} = -\frac{1}{m} \sum_{i \neq 1, 2} \mathbf{f}_i^T \cdot [\mathbf{p}_{12} \times (\mathbf{p}_i - \mathbf{p}_1)]. \quad (9)$$

We seek for a condition under which the left-hand side is smaller than zero, i.e., for which

$$\mathbf{f}_i^T \cdot [\mathbf{p}_{12} \times (\mathbf{p}_i - \mathbf{p}_1)] \geq 0 \quad \forall i \neq 1, 2. \quad (10)$$

The three contact locations $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_i\}$ span a local terrain plane Π_{12i} , whose normal is parallel to $\mathbf{p}_{12} \times (\mathbf{p}_i - \mathbf{p}_1)$. In the most restrictive case, there is no friction, $\mu_i = 0$, implying that (10) needs to be satisfied for $\mathbf{f}_i = \mathbf{n}_i$. We write this condition as $\angle\{\Pi_{12i}, \mathbf{n}_i\} \geq 0$. It is satisfied if the ground can be represented by a single plane (with arbitrary inclination) or segmented into multiple planes perpendicular to gravity.

With these considerations at hand, we can conclude the following.

Proposition 1: There exists a set of contact forces $\{\mathbf{f}_i\}_{i=1, \dots, N}$ satisfying (6a) and (6b), if:

1) the GIA vector is bounded by

$$m \det(\mathbf{p}_{ij}, \mathbf{p}_B - \mathbf{p}_i, \mathbf{g} - \ddot{\mathbf{p}}_B) \leq \mathbf{p}_{ij}^T \cdot \dot{\mathbf{L}}_B; \quad (11)$$

2) the GIA vector is bounded by the friction cone (5);

3) assumption 2 is satisfied;

4) $\angle\{\vec{\Pi}_{jki}, \vec{n}_i\} \geq 0, j \neq k \neq i$ and $j \neq k$.

2) *Two Grounded Legs:* We consider a double support phase where legs 1 and 2 are grounded. Equation (7) becomes

$$m(\mathbf{p}_1 - \mathbf{p}_B) \times (\ddot{\mathbf{p}}_B - \mathbf{g}) + (\mathbf{p}_2 - \mathbf{p}_1) \times \mathbf{f}_2 = \dot{\mathbf{L}}_B. \quad (12)$$

The projection along $\mathbf{p}_{12}$ eliminates $\mathbf{f}_2$

$$m \det(\mathbf{p}_{12}, \mathbf{p}_B - \mathbf{p}_1, \mathbf{g} - \ddot{\mathbf{p}}_B) - \mathbf{p}_{12}^T \cdot \dot{\mathbf{L}}_B = 0. \quad (13)$$

Due to the projection axis, we also need to limit the GIA vector perpendicular to $\mathbf{p}_{12}$. If we look at the moment $\mathbf{M}_1 = (\mathbf{p}_B - \mathbf{p}_1) \times (\mathbf{g} - \ddot{\mathbf{p}}_B) - \dot{\mathbf{L}}_B/m$ induced about foothold $\mathbf{p}_1$, perpendicular to $\mathbf{p}_{12}$, then $\mathbf{e}_z^T \cdot (\mathbf{p}_{12} \times \mathbf{M}_1) \geq 0$, or

$$\det(\mathbf{e}_z, \mathbf{p}_{12}, \mathbf{M}_1) \geq 0. \quad (14)$$

Notice that the formulation requires an overhanging configuration (assumption 6).¹ For the second foothold, Condition (14) can only be satisfied if $\mathbf{e}_z^T \cdot \mathbf{p}_{12} \times (\mathbf{p}_{12} \times \mathbf{f}_2) \leq 0$. Considering again the worst case scenario, i.e., $\mu_i = 0$, the inequality constraint can be simplified to

$$\begin{aligned} \mathbf{e}_z^T \cdot \mathbf{p}_{12} \cdot (\mathbf{p}_{12} \mathbf{n}_2) &\leq \mathbf{e}_z^T \cdot \mathbf{n}_2 \cdot \|\mathbf{p}_{12}\|^2 \\ \mathbf{e}_z^T \cdot \mathbf{p}_{12} \cdot \|\mathbf{n}_2\| \cos \theta &\leq \mathbf{e}_z^T \cdot \mathbf{n}_2 \cdot \|\mathbf{p}_{12}\| \\ \mathbf{e}_z^T \cdot \mathbf{e}_p \cos \theta &\leq \mathbf{e}_z^T \cdot \mathbf{e}_n \\ \mathbf{e}_{p,z} \cos \theta &\leq \mathbf{e}_{n,z} \end{aligned} \quad (15)$$

¹This requirement comes from the fact that the projection axis is chosen as $\mathbf{e}_z$. A more general formulation is possible by replacing $\mathbf{e}_z$ with $\mathbf{p}_B - \mathbf{p}_1$, but would lead to a higher order of nonlinearity.

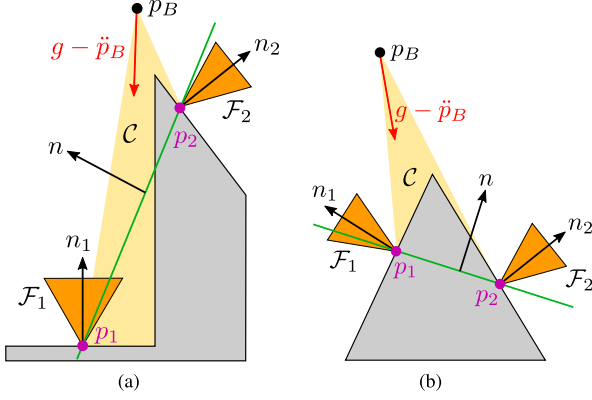


Fig. 4. a) We can find $\mathbf{g} - \ddot{\mathbf{p}}_B$ s.t. it is bounded by the gravito-inertia acceleration cone (GIAC) $\mathcal{C}$ and by the convex hull of friction cones $\{\mathcal{F}_1, \mathcal{F}_2\}$. But we cannot find a contact force $\mathbf{f}_2 \in \mathcal{F}_2$ that is lower bounded by the vector $(\mathbf{p}_2 - \mathbf{p}_1)$. b) We can place the footholds $\mathbf{p}_1$ and $\mathbf{p}_2$ anywhere on the pyramid without ever violating proposition 2.

where we have used $\theta = \angle\{\mathbf{p}_{12}, \mathbf{n}_2\}$, $\mathbf{e}_p = \mathbf{p}_{12}/\|\mathbf{p}_{12}\|$ and $\mathbf{e}_n = \mathbf{n}_2/\|\mathbf{n}_2\|$. Since $e_{n,z} \geq 0$ for a push contact (Assumption 3), the condition further simplifies to $\angle\{\mathbf{p}_{12}, \mathbf{n}_2\} \geq 0$ and we can conclude that

Proposition 2: There exists a set of contact forces $\{\mathbf{f}_1, \mathbf{f}_2\}$ satisfying (6a) and (6b), if:

- 1) the GIA vector is bounded by

$$m \det(\mathbf{p}_{12}, \mathbf{p}_B - \mathbf{p}_1, \mathbf{g} - \ddot{\mathbf{p}}_B) - \mathbf{p}_{12}^T \cdot \dot{\mathbf{L}}_B = 0 \quad (16a)$$

$$\det(\mathbf{e}_z, \mathbf{p}_{12}, \mathbf{M}_1) \geq 0; \quad (16b)$$

- 2) the GIA vector is bounded by the friction cone (5);
- 3) assumptions 2 and 6 are satisfied;
- 4) $\angle\{\vec{p}_{ij}, \vec{n}_j\} \geq 0, \forall i \neq j$.

Fig. 4(a) demonstrates an exotic example where Proposition 2 is violated. It should be noted that Propositions 1 and 2 do not exclude convex nor concave surfaces. For instance, Fig. 4(b) shows a valid example, where horizontal forces cancel each other out.

3) *Single Contact and Full Flight Phase:* In the case of a single contact, the 2-D cone described by (16a) and (16b) simplifies to a line and the induced moment needs to be aligned with it, i.e., $m(\mathbf{p}_B - \mathbf{p}_i) \times \mathbf{a}_B - \dot{\mathbf{L}}_B = \mathbf{0}$. Similarly, the base is constrained to evolve along the ballistic trajectory given by $\ddot{\mathbf{p}}_B = \mathbf{g}$ and $\dot{\mathbf{L}}_B = \mathbf{0}$ during any full flight phase.

D. Model Formulation

Combining Propositions 1 and 2 together with Assumption III-B, we can now state our dynamic stability criterion

$$\mu \mathbf{e}_z^T \cdot \mathbf{a}_B \geq \|(\mathbb{I}_{3 \times 3} - \mathbf{e}_z \mathbf{e}_z^T) \cdot \mathbf{a}_B\| \quad N > 0 \quad (17a)$$

$$m \det(\mathbf{p}_{ij}, \mathbf{p}_B - \mathbf{p}_i, \mathbf{a}_B) \leq \mathbf{p}_{ij}^T \cdot \dot{\mathbf{L}}_B \quad N \geq 3 \quad (17b)$$

$$m \det(\vec{p}_{ij}, \vec{p}_B - \vec{p}_i, \vec{a}_B) = \vec{p}_{ij}^T \cdot \vec{\dot{L}}_B \quad N = 2 \quad (17c)$$

$$\det(\vec{e}_z, \vec{p}_{ij}, \vec{M}_i) \geq 0 \quad N = 2 \quad (17d)$$

$$m(\mathbf{p}_B - \mathbf{p}_i) \times \mathbf{a}_B - \dot{\mathbf{L}}_B = \mathbf{0} \quad N = 1 \quad (17e)$$

$$\mathbf{a}_B = \mathbf{0} \quad \dot{\mathbf{L}}_B = \mathbf{0} \quad N = 0. \quad (17f)$$

Due to its geometric interpretation discussed in Section III-E, we refer to the set of constraints (17a) to (17f) as the *GIAC model*.

In Section VI-B, we recapitulate a method for estimating an external base wrench disturbance $\hat{\mathbf{w}}_{\text{ext},B} = [\hat{\mathbf{f}}_B, \hat{\boldsymbol{\tau}}_B]$. By assuming the wrench to be constant over the prediction horizon, we can complete the dynamic constraints, e.g., (17b) can be written as

$$\det\left(\mathbf{p}_{ij}, \mathbf{p}_B - \mathbf{p}_i, \mathbf{a}_B + \frac{\hat{\mathbf{f}}_B}{m}\right) \leq \mathbf{p}_{ij}^T \left(\frac{\dot{\mathbf{L}}_B}{m} - \hat{\boldsymbol{\tau}}_B\right). \quad (18)$$

E. Properties

In Appendix A, we proof the following.

- 1) The GIA Constraints (17b) to (17f) can be interpreted as the largest inscribing convex cone $\tilde{\mathcal{C}}$ of the so called *gravito-inertia acceleration cone (GIAC)* $\mathcal{C}$, which is defined as the convex hull of rays connecting the base position with the footholds. Rate of change of the angular momentum changes shape and size of these cones (A1).
- 2) The ZMP is the projection of the GIA vector onto the ground. On flat ground, the GIA Constraints (17b) to (17f) simplify to the well known ZMP stability criterion (A2).
- 3) Under Assumption III-B, the GIAC model is *weak contact stable*, a property shared with the closely related CWC models (A3).

IV. PERCEPTION

Next to proprioceptive sensors, which allow the robot to perceive its surroundings by touch, vision is the second most important sense for locomotion. It allows the robot to not only react to the environment (blind) but also to plan its motion ahead w.r.t. the terrain topology (perceptive). The perception pipeline is illustrated in Fig. 5. Two onboard LiDARs are used to generate a detailed *height map* [49], abbreviated by $\tilde{h}$. For performance reasons, projection and visibility clean-up are performed on an onboard GPU [14]. Further map processing algorithms run on CPU and generate three height layers of different smoothness, h , h_{s1} , and h_{s2} .

A. Filtering

We deploy a filter chain consisting of in-painting, outlier rejection, and smoothing.

- 1) First, we iterate over all grid cells in the raw map $\tilde{h}$, replacing empty cells with the minimum found across the occlusion border.
- 2) Reflections, damaged lenses, odometry drift, or a misaligned URDF are known sources of artifacts. These outliers are removed by a sequentially repeated median filter [50] on the inpainted map.
- 3) Given the denoised map h , we compute two additional layers, h_{s1} and h_{s2} .

The former is a slightly Gaussian filtered version of the original map and is used to compute gradients of edges

$$h_{s1} = \text{Gauss}_{\sigma_1}(h). \quad (19)$$

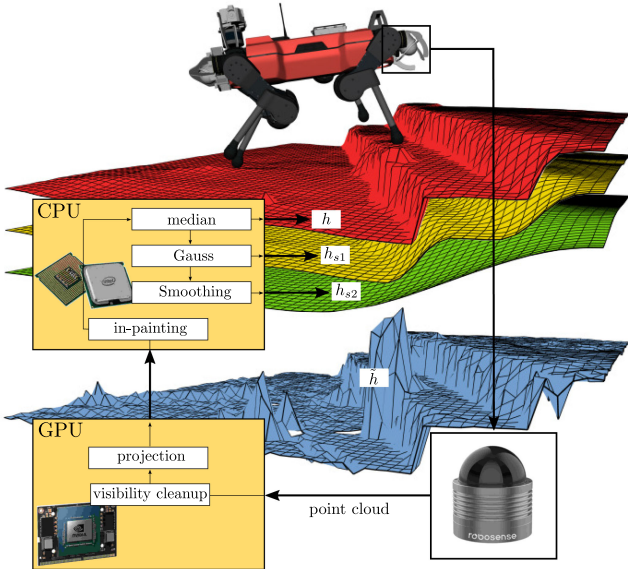


Fig. 5. Illustration of the mapping pipeline using data recorded in a real-world experiment. The raw map $\tilde{h}$ (blue) is generated on GPU and updated at 20 Hz. A filter chain, implemented on CPU, removes noise and artifacts.

The third layer h_{s2} represents a “virtual floor” and is derived in four steps. The map h is aggressively median filtered and then subtracted from itself

$$\Delta h = h - \text{median}(h). \quad (20)$$

The height difference is used to set up a filter mask m

$$m_{ij} = \begin{cases} 1 & \Delta h_{ij} > 0 \text{ (stepping stone)} \\ 1 & \Delta h_{ij} < 0 \text{ (gap)} \\ \infty & \text{otherwise.} \end{cases} \quad (21)$$

In the third step, we widen stepping stones and narrow gaps through masked dilation. This filter replaces height values of h by the maximum found in its close neighborhood. Candidate grid cells must be categorized as either stepping stones or gaps

$$h_{\text{dilated}} = \text{dilate}(h \mid m = 1). \quad (22)$$

Finally, the processed height map is smoothed by a Gaussian filter with a large standard deviation $\sigma_2 > \sigma_1$

$$h_{s2} = \text{Gauss}_{\sigma_2}(h_{\text{dilated}}). \quad (23)$$

Such a four-staged filter hierarchy is, in most cases, just identical to Gaussian smoothing. However, in the presence of positive/negative obstacles, more weight is naturally given to elevated parts, as exemplified in Fig. 6.

Post processing relies on `opencv` [51] filters which are deployed on CPU. For a grid map with dimensions 2.4 m $\times$ 2.4 m and a cell size 4 cm $\times$ 4 cm, the postprocessing step takes about 1.5 ms.

B. Numerical Differentiation and Interpolation

We use a 1-D 5-point central finite difference kernel for computing numerical derivatives

$$\text{1th order} \quad \frac{1}{12\Delta x} \begin{bmatrix} 1 & -8 & 0 & 8 & -1 \end{bmatrix} \quad (24)$$

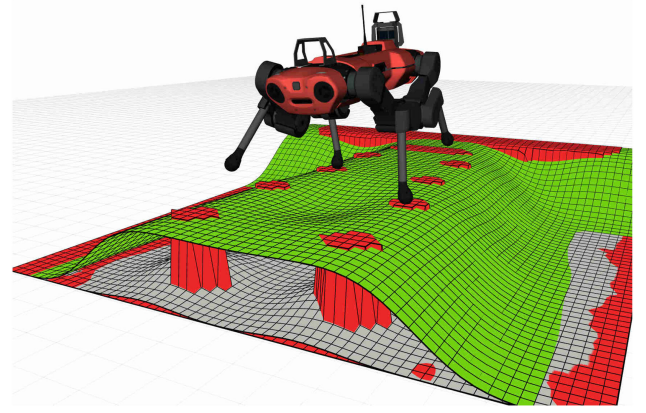


Fig. 6. The red map h , representing geometric features of the terrain, is used for foot placement, while the green map h_{s2} serves as a base pose reference. In contrast to the gray layer, which would be obtained from h by pure Gaussian smoothing, h_{s2} spawns above the stepping stones. This encourages the robot to walk along the path indicated by the cylindrical pillars.

$$\text{2th order} \quad \frac{1}{12\Delta x^2} \begin{bmatrix} -1 & 16 & -30 & 16 & -1 \end{bmatrix} \quad (25)$$

with Δx the grid cell length and width. Smoothing perpendicular to the derivative direction is not necessary, as the height maps $\{h, h_{s1}, h_{s2}\}$ are already relatively smooth. This reduces runtime compared to classical 2-D derivative kernels as the number of multiplications and additions decreases by a factor of five. Gradients and curvatures are computed online whenever needed and stored in a lookup table for potential reuse. As the optimizer typically only visits a tiny fraction of grid cells, this approach is faster than precomputing the convolutions for all layers a priori.

Since the height derivatives are computed on a discrete net, solutions to gradient-based solvers become restricted to the grid map’s spatial resolution. This problem can be most prominently observed as jumps of the footholds between neighboring grid cells across two consecutive optimizations. We tackle this problem using a bilinear interpolation scheme—derivative values are interpolated between four neighboring grid cells.

V. MOTION OPTIMIZATION

We parametrize the base pose trajectory as a 6-D spline of fixed order five. The first three dimensions capture position, and the last three dimensions describe orientation using Euler angles. To increase the feasible space, we allow the base acceleration to evolve discontinuous at contact transitions [5], [17]. We do so by breaking the trajectory into different spline segments connected with each other at the transition times. For the k th spline, we write the l th dimension as

$$\Pi_{B,kl}(t) = a_{0kl} + a_{1kl}t + \dots + a_{4kl}t^4 \quad (26)$$

with $t \in (0, \tau_k)$, $0 \leq k < N_s - 1$, $0 \leq l < 5$, where $a_{0kl}, \dots, a_{4kl}$ are the spline coefficients, N_s is the number of splines, and τ_k is the duration of the k th spline. The method is further explained in Fig. 7.

We stack all unknown variables (which are spline coefficients $a_{0kl}, \dots, a_{4kl}$, positions of grounded feet $\mathbf{p}_{\text{meas}}^T = [\mathbf{p}_{\text{meas},1}^T \ \dots \ \mathbf{p}_{\text{meas},N}^T]$, desired footholds

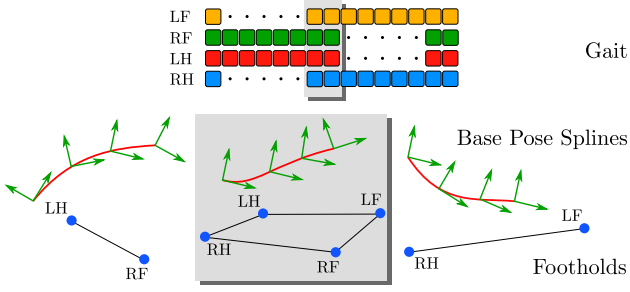


Fig. 7. The gait pattern is fixed and defines lift-off and touch-down timings for each leg. Two adjacent phase events enclose a time interval of constant contact states. Each phase is assigned a 6-D base pose spline and a set of active footholds. The prediction horizon includes one step per leg.

$\mathbf{p}^T = [\mathbf{p}_1^T \cdots \mathbf{p}_4^T]$, and slack variables ε^T together into a state vector $\mathbf{x}$

$$\mathbf{x} = [a_{000} \cdots a_{4(N_s-1)5} \mathbf{p}_{\text{meas}}^T \mathbf{p}^T \varepsilon^T]^T. \quad (27)$$

For the reminder of this section, if not stated differently, we will no more differentiate between foot position and foothold, and refer to both as foothold $\mathbf{p}_i$.

The constraint nonlinear program (NLP) can be formulated as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_i f_i(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{c}_{\text{eq}}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{c}_{\text{ineq}}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \quad (28)$$

where individual objective functions f_i , equality constraint $\mathbf{c}_{\text{eq}}$, and inequality constraints $\mathbf{c}_{\text{ineq}}$ are detailed in the following two sections.

A. Constraints

1) *Initial and Junction Constraints*: Initial constraints are formulated for the base pose $\Pi_B(0)$, base twist $\dot{\Pi}_B(0)$, and foot positions $\mathbf{p}_{\text{meas},i}$ of grounded legs i . Junction constraints are imposed between two adjacent splines to ensure smoothness up to the first derivative, i.e., $\Pi_{B,k+1}(0) = \Pi_{B,k}(\tau_k)$ and $\dot{\Pi}_{B,k+1}(0) = \dot{\Pi}_{B,k}(\tau_k)$.

2) *Dynamic Stability*: The dynamic constraints (17b) to (17f) are implemented as slacked inequalities

$$\min \varepsilon \quad \text{s.t.} \quad \mathbf{c}_{\text{dyn}}(\mathbf{x}) \leq \mathbf{1}\varepsilon, \quad \varepsilon \leq 0. \quad (29)$$

Notice that constraints of type equality can always be transformed to inequalities by $\mathbf{c}_{\text{dyn}}(\mathbf{x}) = [\mathbf{c}_{\text{dyn,eq}}(\mathbf{x}) - \mathbf{c}_{\text{dyn,eq}}(\mathbf{x})]^T$.

The slack variable ε , which is unique to each phase, maximizes the robustness margin, i.e., the smallest angle between the GIA vector and the bounding cone. Choosing a large weight will increase the GIAC volume by pushing the footholds away from each other.

3) *Convex GIAC*: The feasible space imposed by the dynamic stability criterion is given by the cone $\tilde{\mathcal{C}} \subseteq \mathcal{C}$ with $\mathcal{C}$ being the GIAC. If $\mathcal{C}$ is nonconvex, then $\tilde{\mathcal{C}} \subset \mathcal{C}$, and the feasible space shrinks accordingly. Fig. 8 exemplifies three possible scenarios:

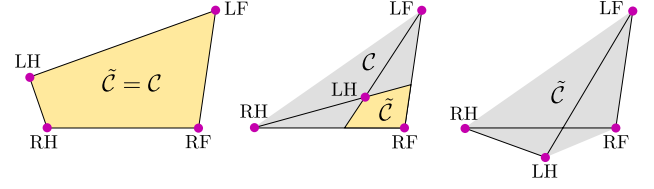


Fig. 8. Gray: GIAC $\mathcal{C}$ as seen from the top. Yellow: Largest inscribing convex cone $\tilde{\mathcal{C}}$, defining the feasible space of the GIA vector. If the cone $\mathcal{C}$ is convex, then $\tilde{\mathcal{C}} = \mathcal{C}$ (left), otherwise $\tilde{\mathcal{C}} \subset \mathcal{C}$ (middle), or $\tilde{\mathcal{C}} = \emptyset$ (right).

No, partial and complete reduction of the support volume. By enforcing convexity of $\mathcal{C}$, we can avoid the reduction of the effective feasible volume, and thereby guarantee that Proposition 3 and Constraints (17b) to (17f) are identical. This can be achieved by constraining the footprint to a convex shape, and the kinematic configuration to an over-hanging torso, i.e., $\mathbf{e}_z^T \cdot \mathbf{p}_B \geq \mathbf{e}_z^T \cdot \mathbf{p}_i$ (Assumption 6). For a phase with four grounded feet $\{1, 2, 3, 4\}$, counterclockwise ordered, the former constraint can be imposed as

$$\begin{aligned} (\mathbf{p}_{13} \times \mathbf{p}_{12}) \cdot \mathbf{e}_z &\leq 0 & (\mathbf{p}_{14} \times \mathbf{p}_{13}) \cdot \mathbf{e}_z &\leq 0 \\ (\mathbf{p}_{24} \times \mathbf{p}_{23}) \cdot \mathbf{e}_z &\leq 0 & (\mathbf{p}_{20} \times \mathbf{p}_{24}) \cdot \mathbf{e}_z &\leq 0. \end{aligned} \quad (30)$$

We also apply (30) in the absence of a full stance phase. We iterate over future phase events and add corresponding footholds until a set of four is complete. This reduces the risk of swing-leg collisions, as explained in Fig. 9(a) and (b).

4) *Friction Cone*: The no-slip condition is embedded in the weak and simplified form (5). An additional constraint is required to ensure that the robot pushes (Assumption 3) on the ground, i.e., $\mathbf{e}_z^T \cdot \dot{\mathbf{p}}_B \geq \mathbf{e}_z^T \cdot \mathbf{g}$.

5) *Kinematics*: We approximate the reachable space of an end-effector i by two balls, centered at the rotation center of the hip²

$$l_{\min}^2 \leq \|\mathbf{p}_B + \mathbf{R}_B \mathbf{r}_i^B - \mathbf{p}_i\|^2 \leq l_{\max}^2. \quad (31)$$

The vector $\mathbf{r}_i^B$ specifies the limb center and $l_{\max}/l_{\min}$ denotes the maximal/minimal limb extension.

B. Objectives

1) *Footholds on Ground*: To ensure that a desired foothold $\mathbf{p}_i$ is located on the height map h , we may incorporate the constraint $h(\mathbf{p}_i) = \mathbf{e}_z^T \cdot \mathbf{p}_i$. As Fig. 10 exemplifies, such a formulation can lead to a suboptimal solution or even local infeasibility. We, therefore, soften the task and write $\min(h(\mathbf{p}_i) - \mathbf{e}_z^T \cdot \mathbf{p}_i)^2$.

2) *Leg Collision Avoidance*: During fast lateral motions, foot collisions with neighboring legs are likely to occur. We wish to enforce a lower bound $\epsilon_{\min}$ on the Euclidean distance in the xy plane between any two feet i and j , for instance

$$z(\mathbf{p}_i, \mathbf{p}_j) = \|(\mathbf{I} - \mathbf{e}_z \mathbf{e}_z^T) \cdot (\mathbf{p}_i - \mathbf{p}_j)\|^2 \geq \epsilon_{\min}. \quad (32)$$

Unfortunately, this constraint is concave and not well suited for gradient-based optimization. Instead, we use a one-sided

²In case of ANYmal, this point can be found where the rotation axis of the HAA and HFE joint intersect.

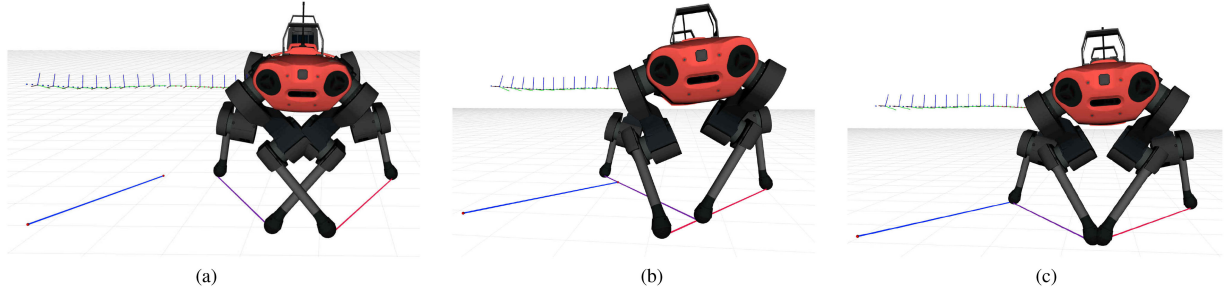


Fig. 9. Illustration of foothold tasks for a trotting gait on flat ground. The lateral reference velocity was 1.5 m/s for all three simulation experiments. For our TO method, we use the second setup, which is less likely to encounter feet or knee joint collisions. (a) Unconstrained footprint: LF overtakes RF, leading to a collision of the two knee joints. Notice how the kinematic configuration is perfectly symmetric during the full stance phase. (b) Enforcing convex footprint and collision avoidance: LF, RF, and LH join on a line. The realized velocity is considerably smaller and the kinematic symmetry is perturbed. (c) Constraint for footprint convexity is active, but leg collision constraints are disabled. The two front legs are likely to step on each other.

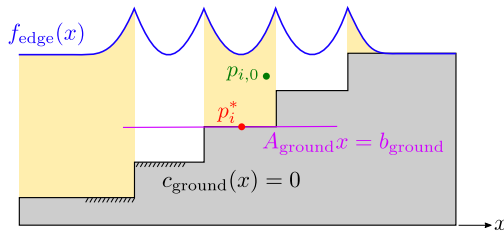


Fig. 10. Given an initial foothold $p_{i,0}$, the optimizer will find a solution p_i^* , which is most likely located on the same tread. The cost function for edge avoidance f_{edge} increases on both ends of the tread, inducing a local optimum at its center. Moreover, the linearization of the height constraint $c_{\text{ground}} = 0$ is only valid on the tread where the constraint was linearized about. This renders exploration beyond the initial tread difficult because either the line search could reject such an update or the update could lead to an infeasible kinematic configuration in the next SQP iteration. A soft task allows finding feasible footholds by exploring unfeasible intermediate solutions.

quadratic barrier approximation [52]

$$\min \begin{cases} (\epsilon_{\min} - z(p_i, p_j))^2 & z < \epsilon_{\min} \\ 0 & z \geq \epsilon_{\min} \end{cases}$$

Fig. 9(b) illustrates the impact of the collision avoidance task on the solution.

3) *Nominal Kinematics*: We want to relate a set of nominal footholds with a nominal base pose. By introducing a desired leg extension vector $\mathbf{l}_{\text{des}} = \begin{bmatrix} 0 & 0 & h_{\text{des}} \end{bmatrix}^T$ with h_{des} the desired height above ground, we can write

$$\min \|\mathbf{p}_B + \mathbf{R}_B \mathbf{r}_i^B - \mathbf{p}_i - \mathbf{l}_{\text{des}}\|^2. \quad (33)$$

For each foothold, the objective is applied once per stance. More specifically, we define this time as midstance, which introduces the symmetric behavior observed in Fig. 9(a).

4) *Base Pose Alignment*: A popular approach used in blind quadrupedal locomotion is to align the torso with some local terrain estimate [5], [6], [53], but similar ideas are also used for perceptive planners [14], [33]. The terrain is typically represented by a plane, obtained by a least-squares fit from current and previous stance foot locations [53].

We choose a more general approach and invoke

$$\min \sum_{i=1}^4 [e_z^T \cdot (\mathbf{p}_B + \mathbf{R}_B \mathbf{r}_i^B - \mathbf{l}_{\text{des}}) - h_{s2}(\mathbf{p}_B)]^2. \quad (34)$$

The local plane is somewhat implicitly given by the spatial locations of limb thighs $\{\mathbf{p}_B + \mathbf{R}_B \mathbf{r}_i^B\}_{i=1,\dots,4}$, and objective (34) resembles the least squares fit. The smooth map h_{s2} is preferred over h , as it “smooths out” nonconvexity.

5) *Edge Avoidance*: The robot shall avoid stepping on edges to minimize the risk of slippage. Methods that rely on plane segmentation avoid this problem naturally for point feet [33], or at least give rise to certain simplifications for humanoid feet [20]. An often encountered idea merges geometric features into a *foothold score* and selects footholds in a way that minimizes this quantity. Such cost functions are widely used in MMO structures [12]–[14], [16], [17].

Plane segmentation and foothold scores are computationally expensive. Moreover, for gradient-based methods, the foothold score needs to be at least locally convex³ and conventional traversability maps or binary scores do not apply. Instead, we suggest to penalize directly height gradient

$$\min \nabla h(\mathbf{p}_i)^T \nabla h(\mathbf{p}_i) + \nabla h_{s1}(\mathbf{p}_i)^T \nabla h_{s1}(\mathbf{p}_i). \quad (35)$$

We utilize the blurred version h_{s1} in the second term to push the optimum away from edges. The larger the standard deviation of the Gaussian kernel is chosen, the farther away the feet will be placed from the edges, but the more details are blurred out. Both cost terms penalize inclination, which is in line with the horizontal contact plane Assumption 5.

6) *Previous Solution*: In the presence of external (e.g., slip) and internal (e.g., noisy elevation map, imperfect tracking) disturbances, desired footholds are likely to jump in between two adjacent optimization steps. And such jumps are particularly large if the ground is cluttered with edges or other infeasible foothold locations. To increase the step confidence, we minimize squared distance to the previous optimal foothold.

7) *Tracking*: We track momentum in base frame by

$$\min \frac{\|\mathbf{P}_B^B - \mathbf{P}_{\text{des}}^B\|^2}{m^2} + \|\mathbf{L}_B^B - \mathbf{L}_{\text{des}}^B\|^2. \quad (36)$$

³There needs to be a gradient that pushes the foothold in the “good” direction.

TABLE I
WEIGHTS FOR SOFT TASKS; CONTINUOUS-TIME TASKS ARE SAMPLED ALONG
THE TRAJECTORY AND MARKED AS “SAMPLED”

Task	Sampled	Weight
Robustness margin		0.007
Footholds on ground		10^4
Leg collision avoidance		0.001
Nominal kinematics		7
Base pose alignment	✓	$100 \cdot T_k$
Edge avoidance		3
Previous solution		0.01
Tracking	✓	$2 \cdot T_k$
Smoothness	✓	$0.001 \cdot T_k$

This task is identical to approach the planned twist toward a reference twist $\{\dot{p}_{B,\text{des}}^B, \omega_{B,\text{des}}^B\}$ and using the weight $\mathbb{I}_{3 \times 3}$ for the linear and the weight $\mathbf{I}_B^T \mathbf{I}_B$ for the angular velocity.

8) *Smoothness*: In order to not diverge too far from Assumption 2, we minimize the rate of change of angular momentum by $\min \|\dot{\mathbf{L}}_B\|^2$.

C. Implementation Details

1) *Solver*: By leveraging the *Gauss–Newton* (GN) method, as summarized in Appendix B, we can extract the positive definite part of a nonconvex objective. We use a custom-made sequential quadratic program (SQP) solver that internally approximates the problem as a sequence of quadratic programs (QPs). The convexity of the resulting QPs is exploited by deploying an efficient implementation of QuadProg++ [54].

At each iteration, a line search globalization approach [55] trade off cost function minimization and constraint violation to compute a search step length. As long as the linearization of the feasible set is nonempty, the SQP solver is guaranteed to find a local solution. Since the tasks can be nonconvex on a global scale, optimality can only be guaranteed locally.

2) *Automatic Differentiation*: Analytical derivatives of the tasks are sometimes difficult to derive by hand. They might also be numerically inefficient due to a large number of trigonometric multiplications. To this end, we deploy CppADCodeGen [56], a C++ library for automatic differentiation and code generation.

3) *Trajectory Sampling and Tuning*: Some tasks appear once per problem or once per spline segment, while others are enforced along the entire prediction horizon. These continuous tasks are sampled on a discrete grid and enforced for each $T_k = \tau_k/6$ s. Table I summarizes the weights used in TAMOLS which are identical for all gaits. The weight for the foothold task as introduced in Section V-B1 is chosen comparably large s.t. the constrained violation of a kinematically feasible problem stays within the mapping accuracy (< 5 mm). Significant constraint violations are very rare and only happen when no feasible solution exists around the initial guess.

D. Initialization

In contrast to most interior point methods, SQP solvers can be efficiently warm started, in our case, using the latest optimal solution. However, at each leg touch-down, a new foothold and base pose spline segment appear, for which there is no

previous solution available. Initializing the new states with some heuristics is prone to drive the solution into a local optimum. This issue is elaborated using the stair example in Fig. 10. Additional problems arise due to the GN approximation, which disregards important second-order derivative information. For instance, the objective for edge avoidance as introduced in Section V-B5 loses the curvature of the map, rendering the classification of extrema impossible. It is crucial to provide an initial guess located sufficiently far from the saddle or maximum points in this particular case.

Even though indisputably important, the available literature often does not give insightful information into the initialization procedure. Some exceptions can be found in the data-driven community, where initial guesses are learned [28], [34], [57].

1) *Batch Search Optimization*: In our previous work [14], we have introduced *batch search*. The algorithm first establishes nominal footholds, based on Raibert’s heuristic, around an approximate base pose trajectory. It then iterates within a circular search space centered about the nominal foothold, selecting a kinematically reachable grid cell with the lowest cost. Since footholds are selected independently from each other, and the base pose trajectory does not take into account the terrain topology, such an initial guess might lead to a suboptimal or even infeasible optimization problem.

2) *Graduated Optimization*: Instead of finding a more sophisticated initial guess capable of avoiding the aforementioned pitfalls, we deploy a global optimization technique. At each leg touch-down, we solve a sequence of TO problems, starting with a greatly simplified problem and progressively approaching toward the original problem. Such a technique is generally known as *graduated optimization* [4].

In the case of TAMOLS, nonconvexity and discontinuity are introduced majorly by the terrain and the base orientation, but most prominently with the two foothold tasks as introduced in Sections V-B1 and V-B5. This observation gives rise to the idea of solving a “nearly convex” TO in the first iteration by replacing the height maps h and h_{s1} with h_{s2} . The robot’s measured state suffices already as an initial guess for this simplified NLP to succeed.

We could now continue in the sense of graduated optimization and introduce several virtual floors of different smoothness that gradually approach to the original map. For each of these virtual floors, a TO problem needs to be solved, progressively adjusting the solution toward the actual terrain. Even though such an approach may be capable of finding the global solution, it would not be tractable due to the exploding number of optimization steps.

3) *Hierarchical Graduated Optimization*: We notice that task as introduced in Section V-B4 keeps the base pose aligned with h_{s2} . We, thus, expect the base trajectory to stay approximately constant across different iterations of the graduating optimization scheme. This leads to the idea of truncating the sequence after solving the first TO problem while replacing the remaining sequence with a foothold refinement step. We do so by leveraging a simplified version of the batch search. The embedded cost functions penalize occluded cells of $\tilde{h}$, gradient and positive curvature of h_{s1} and vertical distance to h_{s2} within a 0.4 m wide search area centered about the previously optimized foothold.

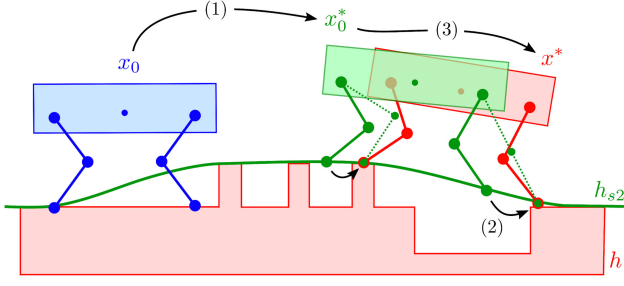


Fig. 11. (a) Given an initial guess x_0 (blue) the motion is optimized over the virtual floor h_{s2} , resulting in the intermediate solution x_0^* (green). (b) The postprocessed intermediate solution (dashed) is obtained by refining the footholds w.r.t. to the height map h . (c) The refined solution is used to warm start the original NLP, constraining the motion to h and producing the final solution x^* (red).

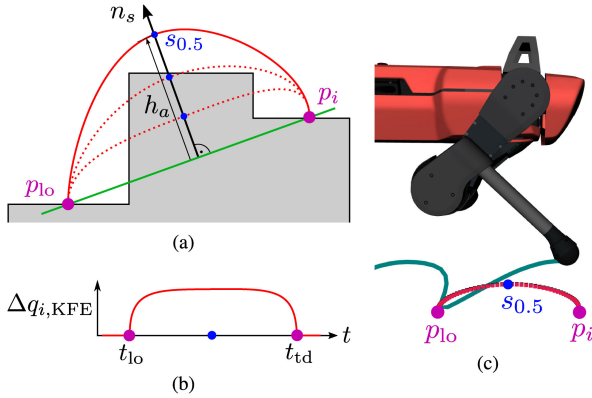


Fig. 12. (a) Geometric construction of the swing trajectory in task space. (b) The knee trajectory flexes the joint between lift-off t_{lo} and touch-down time t_{td} . (c) Overlaid task space trajectory.

In the last stage, the optimized and refined state is subsequently passed over the actual NLP as a new initial guess. The resulting three-staged optimization hierarchy is visualized in Fig. 11 and applied at each leg touch-down.

E. Swing Trajectory

A collision-free swing trajectory $s(t)$ is established that connects the lift-off position p_{lo} with the desired foothold p_i . The gradient-free method is geometrically motivated in Fig. 12(a). The trajectory consists of two quintic splines, smoothly connected at midswing phase, corresponding to spatial location $s_{0.5}$. The parametrization introduces $6 \cdot 2$ spline coefficients, from which $3 \cdot 3$ are given by the initial, final, and junction conditions. We assume that the edge of an obstacle is located in between the lift-off and touch-down position, along the step normal n_s . We place the spline junction onto that edge, i.e., $s_{0.5} = h_a n_s + 0.5(p_{lo} + p_i)$, leaving one degree of freedom (DOF) left. The apex height h_a is found iteratively by lifting $s_{0.5}$ along n_s until the obstacle is cleared.

Due to tracking errors and odometry drift, a swing leg may still collide with the environment. After the collision, the foot travels along the vertical direction of the obstacle until a stable contact is detected. The new touch-down location shrinks the volume of the planned GIAC, and stability of the truncated motion is no more guaranteed. We wish the collision to take place in an early stage of the swing phase, s.t. the GIA vector is

TABLE II
TASK PRIORITIES USED FOR WBC

task	type	priority
equations of motion	=	0
joint torque limits	≤	0
kinematic limits	≤	1
friction pyramid (stance legs)	≤	1
no contact motion (stance legs)	=	1
tracking in task space (swing legs)	=	2
tracking in task space (torso)	=	3
tracking in joint space	=	4
minimize contact forces	=	4

still contained in the stability margin. In other words, we wish the foot to travel in the shortest possible time to the apex. Setting up such trajectories in task space might require accelerating a lot of inertia.⁴ Instead, we overlay the swing trajectory with a joint space offset trajectory, that only flexes joints carrying the smallest inertia, i.e., all knee flexion/extension (KFE) joints. Let q_i be the generalized coordinates of limb i , then the new swing motion becomes

$$\begin{aligned} q_{i,des,new}(t) &= q_{i,des}(t) + \Delta q_i(t) \\ \dot{q}_{i,des,new}(t) &= \dot{q}_{i,des}(t) + \Delta \dot{q}_i(t) \\ \ddot{q}_{i,des,new}(t) &= \ddot{q}_{i,des}(t) + \Delta \ddot{q}_i(t). \end{aligned} \quad (37)$$

A valid offset trajectory Δq_i needs to satisfy

$$\begin{aligned} j \neq \text{KFE}: \quad \Delta q_{ij}(t) &= 0 \quad \forall t \\ j = \text{KFE}: \quad \Delta q_{ij}(t) &= 0 \quad t_{td} < t < t_{lo}. \end{aligned} \quad (38)$$

Fig. 12(b) shows a plot of the KFE element. We first convert the swing trajectory into joint space using analytical inverse kinematics, pseudo-inverse differential kinematics and dynamics, yielding desired joint positions $q_{i,des}(t)$, velocities $\dot{q}_{i,des}(t)$, and accelerations $\ddot{q}_{i,des}(t)$. After adding the offsets $\{\Delta q_i, \Delta \dot{q}_i, \Delta \ddot{q}_i\}$ the signals are converted back to task space by solving the forward problem, leading to a new swing trajectory as shown in Fig. 12(c).

VI. WHOLE BODY CONTROL

A. Tracking

The reference signals are tracked by a WBC [5] in task space, while an impedance control law [14] improves tracking under low loads. Table II outlines the task priorities. The tasks are categorized into three blocks, where the first one contains physical constraints at highest priority. The middle block ensures stable contacts and tracks the reference trajectories in task space. The prioritization is such that a saturation of kinematic limits causes a base pose adaptation in favor of swing leg tracking. This feature is particularly useful in the case of regaining [58], i.e., an event where a leg expects a contact, but it has not been established yet. In this case, the foot is pushing vertically toward the ground with a constant velocity. The motion optimizer will adjust the pose trajectory accordingly, but with a delay of one

⁴The knee joint of ANYMal is relatively heavy.

optimization duration. In this short time period, the WBC sacrifices tracking performance of the base pose to avoid a regaining leg to enter a singular configuration. We use the kinematics limits task introduced in [58], where we constrain only the KFE joints to enforce the X-configuration.⁵ The last block consists of low-priority tasks used to completely resolve the null-space.

B. GM-Observer and Disturbance Rejection

Consider the equations of motion (EOM) of an articulated robot structure with n actuated joints

$$M(q)\dot{u} + h(q, u) = S^T \tau + J(q)^T F_{\text{ext}} \quad (39)$$

with $q \in \mathbb{R}^{(n+6)}$, $u \in \mathbb{R}^{(n+6)}$ the generalized coordinates and velocities, $\tau \in \mathbb{R}^n$ the joint torques, $F_{\text{ext}} = [\mathbf{f}_B \ \tau_B \ \mathbf{f}_1 \ \dots \ \mathbf{f}_N] \in \mathbb{R}^{6+3N}$ a vector stacking external base force, base momentum and contact forces, $M(q)$ the mass matrix, $h(q, u)$ the Coriolis, centrifugal and gravity terms, S^T the selection matrix of the actuated DOFs and $J(q) \in \mathbb{R}^{(n+6) \times (n+6)}$ the stacked contact Jacobians. A GM observer is a momentum integrating structure that estimates the external forces using only the measurements $\{q, u, \tau\}$

$$\hat{\tau}_{\text{ext}} = \text{GM}(q, u, \tau) = J(q)^T \hat{\mathbf{F}}_{\text{ext}} \in \mathbb{R}^{n+6}. \quad (40)$$

The filter does not depend on measured joint accelerations or mass matrix inversion and serves as a virtual sensor for “external joint torques” acting on the base. It is well studied [59] and widely applied in locomotion for disturbance compensation [60] and contact estimation [61].

Instead of discretizing the continuous filter dynamics, we implement a discrete-time version proposed in [61] with a cutoff frequency of 40 Hz.

The external torque estimate is postprocessed using saturation, median, and low pass elements as illustrated in Fig. 13.

A disturbance on a swing leg will cause an equivalent reaction on the base. It is desirable to project these disturbances back into the floating base. Let S_B be a matrix selecting the floating base DOFs, S_i a matrix selecting the actuated DOFs of the i th limb, and $J_{B,i}(q)$ be first six columns of the contact Jacobian associated to the i th limb. Then, the base wrench disturbance becomes

$$\hat{\mathbf{w}}_{\text{ext},B} = S_B \hat{\mathbf{F}}_{\text{ext}} + \sum_{i=\text{swing}} J_{B,i}(q)^T S_i \hat{\mathbf{F}}_{\text{ext}}. \quad (41)$$

The output of the GM observer can be directly used to compensate for the external disturbances. A convenient way to do so is by manipulating the nonlinear term $h(q, u)$

$$\begin{aligned} S_B \tilde{h}(q, u) &= S_B h(q, u) - \hat{\mathbf{w}}_{\text{ext},B} \\ S_i \tilde{h}(q, u) &= S_i h(q, u) - S_i \hat{\tau}_{\text{ext}} \quad \forall \text{ swinging } i. \end{aligned} \quad (42)$$

Replacing h with $\tilde{h}$ in all tasks of the WBC allows compensating directly for disturbances at the base and the swing feet.

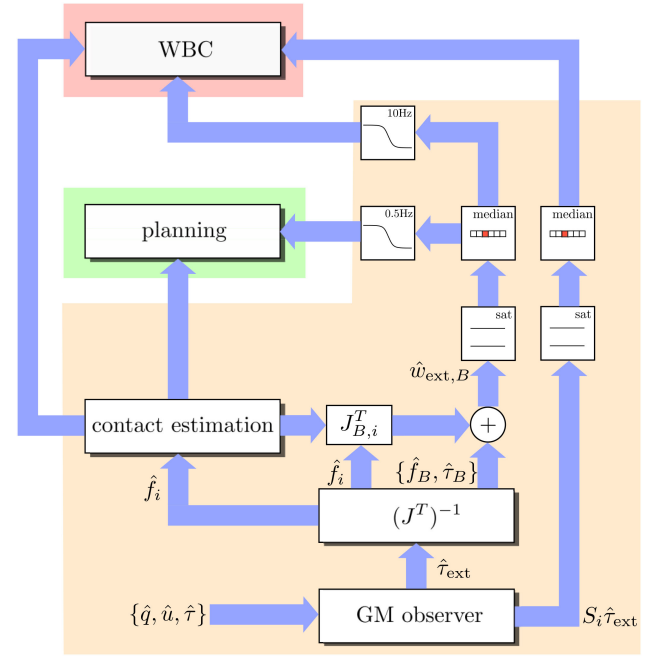


Fig. 13. The GM observer estimates disturbances in joint space. The task space disturbance can be calculated by inverting the contact Jacobian J^T . The vertical components of the estimated contact forces are thresholded to contact states. If a leg is swinging according to this estimate, contact forces are projected back into the floating base through the Jacobian matrix $J_{B,i}^T$. The base wrench and joint torques are further mean and low-pass filtered to allow for smooth compensation on the planner and tracking side.

VII. RESULTS

We evaluate the performance of our perceptive control pipeline using the ANYmal platform. ANYmal is a fully ruggedized quadrupedal robot with 12 actuated DOFs, designed to autonomously operate in challenging environments. Two LiDARs (Robo-Sense bperl) are mounted in the front and back of the torso. Elevation mapping runs at 20 Hz on an onboard GPU (Jetson AGX Xavier) while control and state estimation is updated on a separate onboard CPU (Intel i7-8850H, 2.6 GHz, Hexa-core 64-b) at 400 Hz. TAMOLS runs asynchronously at the maximum possible rate and optimizes trajectories for a prediction horizon of one full gait cycle.

A. Computation Times (Real World)

TAMOLS is able to produce motions for a wide variety of dynamic gaits. We record optimization durations while ANYmal is walking on flat ground with a maximum heading velocity. Mean and median, computed over five strides, are plotted in Fig. 14. Each gait introduces a different number of decision variables, which grows linearly with the number of phases (splines) covered by the prediction horizon. This dependency can also be observed in the computation time.

For the most common gait trot, we achieve an average optimization duration of 6.3 ms, which is 48 times faster than the latest state-of-the-art fully perceptive CMO pipeline [28]. Compared to the blind MMO controller presented in [6], which shares the same parametrization of the base pose, we are only

⁵Depending on the joint index, the angle is upper or lower bounded by 0.

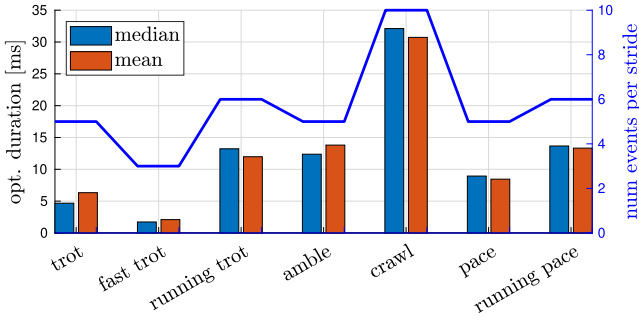


Fig. 14. Optimization duration (elapsed time between initialization to retrieving the solution) for a prediction horizon of one stride. Fast trot eliminates the full stance phase of trot, while the swing phases overlap for running trot. Amble is obtained from crawl by overlapping all swing phases until the triple stance phase vanishes. Pace/running pace has the same gait timing as trot/running trot but has a lateral leg pair alternating.

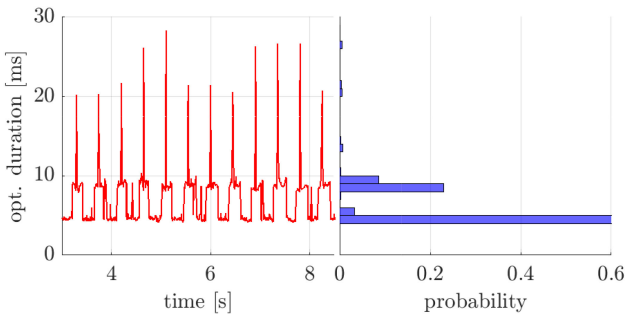


Fig. 15. Left: Computation duration recorded over time for a trotting gait. Right: Corresponding normalized histogram.

2.9 times slower. If we also eliminate the full stance phase (fast trot), the average computation duration drops to 2.1 ms.

Fig. 15 contains the histogram for the trotting gait. The SQP converges most often after one (5 ms) or two (10 ms) iterations. The peak values correspond to touch-down events, for which TAMOLS also optimizes over the initial guess and performs a batch search. The latter takes in average about 0.15 ms. On rough terrain, the peaks are slightly larger (about one SQP iteration) due to a mismatch between the height maps h_{s2} and h , and thus the optimized initial guess and the solution.

B. Staircase (simulation)

In our previous work [14], we have presented an MMO pipeline and discussed its limitations. We found that a significant amount of failure cases on stairs were caused by knee joint collisions with edges (upstairs) or by violating reachability constraints (downstairs). The experimental setup included a simulated staircase of 12 treads, which was passed 18 times in ascending and descending direction using a trotting gait. The success rates are recapitulated in Table III.

We repeat the exact same experiment with TAMOLS. The different limb lengths of the robot versions are taken into account by increasing height and width of each tread by a factor 1.27.⁶

⁶The thigh length increased from 0.25 to 0.3 m and the shin length from 0.3 to 0.32 m. The total relative limb extension is 12.7 % longer.

TABLE III
SUCCESS RATE SCORED WHILE WALKING ON STAIRS

method gait	batch search trot	TAMOLS		
		trot, amble, running	trot, pace	crawl
up	10/18	18/18		16/18
down	14/18	18/18		14/18

We further choose a reference velocity that allows the robot to progress one tread per step.

The statistical results are presented in Table III. While trotting upstairs, the KFE joints occasionally collide with the edges, but the robot is always able to keep its balance. Contrary to our previous control framework, we do not encounter issues with leg over-extensions. This can be explained by the kinematic constraints that are present in the prediction and the tracking level. Moreover, we successfully repeat the experiment with three additional dynamic gaits: running trot, amble, and pace. For the sake of completeness, the experiment is also performed for crawl. Due to its large stride, the robot is very sensitive to contact state mismatches (downstairs) and fails to react to knee joint collisions (upstairs).

C. Staircase (Real World)

We take ANYmal to the real world and walk upstairs using a trot in the building of our lab. Two floors are connected with each other by 20 treads where each platform has the dimensions $29\text{ cm} \times 17\text{ cm}$, forming an inclination of 36° . The operator commands a heading speed of 0.45 m/s whereas the realized average velocity is 0.37 m/s. The tracking error originates mainly from the reduced feasible space imposed by the stair geometry. The robot prefers to place the footholds, s.t. all feet either clear zero, one, or two treads per stride. Depending on the magnitude of the reference velocity, a different optimum is generated, which was found at one tread per step for our experiment. This behavior is further investigated in the subsequent experiment Section VII-D.

In Fig. 16, we visualize the three height maps used for motion optimization along with the raw height map. For a staircase, the computation of the virtual floor h_{s2} equals Gaussian filtering of the height map h . The smooth map h_{s1} attains zero gradient in the middle of the tread, thereby pushing footholds away from edges toward the center line.

D. Velocity Modulation (Simulation)

We want to investigate the relation between commanded and realized velocity when walking 12 steps upstairs. We use the stair parameters from the previous experiment and a fast trot. The relationship between reference velocity and the number of stair treads traversed per step is visualized in Fig. 17. For heading reference speeds larger than 0.9 m/s, the robot always clears two treads per step. Kinematic limits prevent the robot from progressing any faster. If the commanded velocity is reduced below 0.9 m/s, the robot alternates between clearing two and one treads per step. Clearing exactly one tread per step is found as the optimal solution over a wide range of reference velocities $\{0.45, \dots, 0.55\}$ m/s.

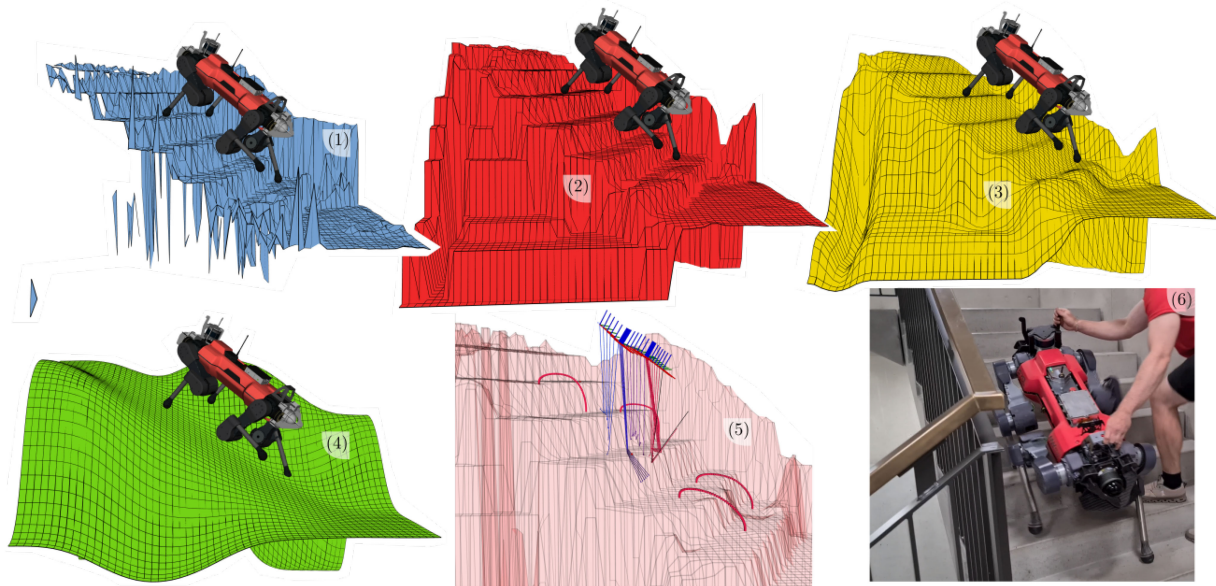


Fig. 16. ANYmal trotting up stairs. (1) Visualization of the robot together with the raw elevation map $\tilde{h}$. (2) The inpainted and denoised elevation map h is used for height constraints on the footholds. (3) The Gaussian-smoothed height map h_{s1} is used for edge avoidance. (4) The robot aligns its torso with the virtual floor map h_{s2} . (5) Visualization of the motion plan in task space. (6) Snap-shot of the experiment.

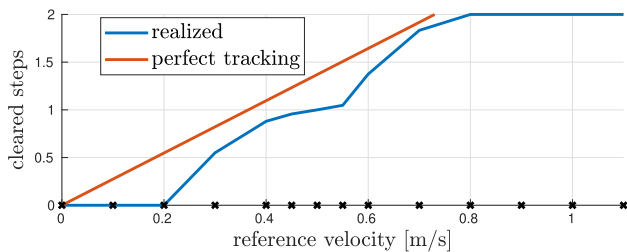


Fig. 17. Average number of stair treads cleared per step and per foot, which is proportional to the realized average velocity. The count starts when the second foot strikes the first tread and stops if the first foot reaches the top platform. Blue: Results for a fast trot. The attractive velocities clear zero, one, or two treads per step. Red: Hypothetical curve that would lead to perfect velocity tracking. Black: Reference velocities for which experiments were conducted.

It can be observed that the realized velocity is always smaller than the reference. TAMOLS trades off robustness against velocity tracking. Increasing the weight for minimizing ε in (29) increases the robustness margin but also degrades velocity tracking.

E. Gap (Real World)

The strength of the proposed TO method lies in the generalization to stairs, gaps, and stepping stones. In the experiment outlined in Fig. 18, we test the controller's behavior in the presence of a gap. The robot is required to traverse over a pallet and slope using an ambling gait. Both obstacles are placed around 30 cm apart from each other.

By penalizing inclinations, the robot successfully avoids the steep part of the pallet. The height layer h_{s2} connects the pallet with the slope, thus, creating an artificial floor. The cost function of the batch search increases quadratically with the vertical foothold distance to h_{s2} , virtually rendering the gap unattractive

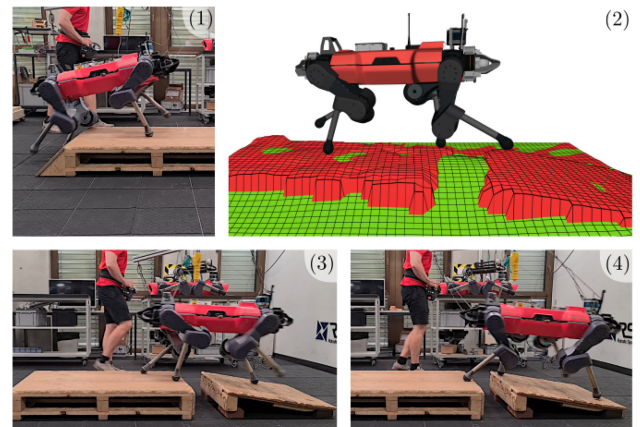


Fig. 18. ANYMal ambling over two obstacles, forming a gap. The commanded heading speed is 0.7 m/s. The gap has a height of 20 cm and width of 27 cm. The second pictures shows the two layers h (red) and h_{s2} (green).

to step. The experiment was repeated five times, from which the robot never stepped on the inclined slope and placed once RH foot into the gap.

F. Stepping Stones (Real World)

We validate the control performance in a stepping stone experiment as described in Fig. 19. By penalizing gradients of the smooth map h_{s1} , the robot centers the footholds in the middle of the stepping stones. Due to odometry drift, the bricks, as seen by the elevation map, may be translated by a few centimeters from their actual locations. This explains why the robot sometimes steps close to the border.⁷

⁷When visualizing the robot together with the elevation map, the footholds appear in the middle of the brick.



Fig. 19. Anymal traversing a stepping stone parkour made of inclined wooden bricks using a trotting gait. The commanded heading velocity is 0.4 m/s. The dimensions of each brick are 20 cm $\times$ 20 cm $\times$ 50 cm and the gap between any two adjacent bricks measures 20 cm. Due to the inclined surfaces, the model assumptions are not perfectly satisfied in this experiment. (7) Visualization of the robot along the raw elevation map.

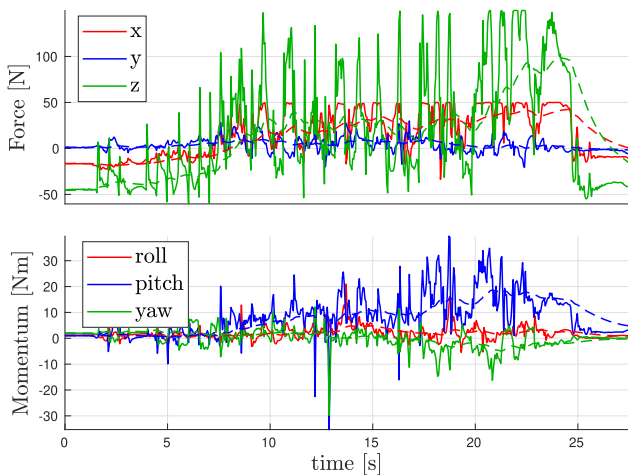


Fig. 20. Estimate of the external force (top) and moment (bottom) acting on the base. Solid: Low-pass filtered at 10.0 Hz, compensated at tracking level. Dashed: Low-pass filtered at 0.5 Hz, compensated at planning level. Horizontal/vertical forces are thresholded at 50 N, 150 N.

The robot is attached to an industrial crane during the experiment, exerting an external force on the base. The wrench estimate of the GM observer is plotted in Fig. 20. It can be seen that the forces do not converge to zero at rest ($t < 2$ s and $t > 25$ s). This is due to modeling errors, in particular inaccurate whole body mass and COM location. Each peak of the unfiltered forces corresponds to one step, which requires the robot to push the crane for the same distance walked. Toward the end of the path, the robot attempts to walk down the slope while the crane is holding back the torso. This disturbance is experienced majorly as a horizontal force and pitch moment, as can be seen in the plots for the time interval $t \in (20, 25$ s).

G. Limitations

The major drawback of TAMOLS originates from the dynamic model approximation: The method is only guaranteed

to find feasible (i.e., weak contact stable) motions if it finds horizontal contact surfaces. If at least one of the feet is located on a tilted plane, the WBC may deviate from the planned trajectory in order to ensure the no-slip condition. Theoretically, it is possible to generalize the model to a more general terrain structure by enforcing the last property in Propositions 1 and 2 through hard constraints. However, it may be very difficult to find such footholds.

Second, TAMOLS does not take into account the full kinematics. The simplified kinematic constraints formulated in task space greatly help to avoid leg over-extension but knee joint collisions still remain a problem.

As a final limitation, we mention the strong dependence of the footholds and base pose on the quality of the elevation map. State estimator drift shifts the map relative to odometry. This leads to a mismatch between realized and planned foot locations, which, in extreme cases, can destabilize the system.

VIII. CONCLUSION

First, we have derived a differentiable and contact-force free dynamic stability metric. It was shown to be at most as complex as ZMP constraints while attaining a large fraction of validity of SRBD. In the second part of this work, we have embedded our model into a terrain-aware motion optimizer. Experiments have suggested that underlying assumptions generalize well to rough, human-engineered environments. By deploying graduated optimization, we have shown that it is possible to jointly optimize footholds and base pose over rough terrain in real-time. Compared to our previous work, we have shown that the inclusion of footholds into the optimization problem successfully removes the appearance of leg-over-extension.

APPENDIX A

A. Geometric Derivation and Interpretation of the GIAC

To start with, let us consider a two-legged 2-D robot that establishes point-contacts with the environment at p_1 and p_2 ,

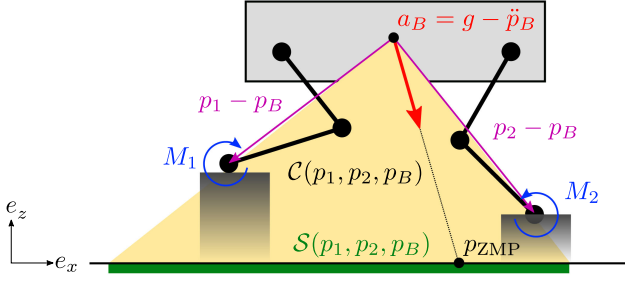


Fig. 21. The GIA vector $\mathbf{a}_B = \mathbf{g} - \dot{\mathbf{p}}_B$ induces the moment $\mathbf{M}_{1,2}$ about the contact positions $\mathbf{p}_{1,2}$. The cone $\mathcal{C}$, spanned by the base $\mathbf{p}_B$ and the two footholds, defines the set of all admissible $\mathbf{a}_B$. The ZMP equals the intersection of the ray $\mathbf{p}_B + \gamma \mathbf{a}_B$ with the ground. The associated support polygon $\mathcal{S}$ is obtained by projecting the footholds onto the ground.

as illustrated in Fig. 21. For this particular example, we assume the following.

Assumption 7 (Inverted pendulum mode):

- 1) The limbs have zero mass.
- 2) The base inertia is negligible.
- 3) Contact forces can only push on the ground.
- 4) Friction is infinitely large.
- 5) Contacts are established on horizontal planes.

The gravito-inertia acceleration vector $\mathbf{a}_B$ induces a moment $\mathbf{M}_i = m(\mathbf{p}_B - \mathbf{p}_i) \times \mathbf{a}_B$ around the two feet. The rays defined by the vectors $\mathbf{p}_i - \mathbf{p}_B$ play an important role as they define the support cone:

Definition 1 (GIAC): The convex cone of rays, connecting the base position and the footholds, is called gravito-inertia acceleration cone (GIAC) $\mathcal{C}$.

If the GIA vector $\mathbf{a}_B$ stays on the left side of the vector $\mathbf{p}_B - \mathbf{p}_2$, the robot may be stable. However, if it were located on the opposite side, the induced moment would be such that the robot might tip over the right foot. Similar considerations also apply to the opposite foot, and it can be concluded that

Proposition 3: Under Assumption 7, a robot does not tip over any supporting foot if $\mathbf{a}_B \in \mathcal{C}$.

We note that the opposite is not always true, i.e., $\mathbf{a}_B \notin \mathcal{C}$ does not imply that the robot will tip over a supporting foot. For instance, taking the example of Fig. 21, the contact force $\mathbf{f}_1$ of the left foot can create a counter moment $(\mathbf{p}_1 - \mathbf{p}_B) \times \mathbf{f}_1$ to balance a GIA vector that lies on the right side of the GIAC.

First and foremost, notice that Proposition 3 is not restricted to a certain dimensionality. In two dimensions, the stability criterion $\mathbf{a}_B \in \mathcal{C}$ can be easily converted to inequality constraints by checking the sign of the induced moment about each foot. A similar conversion can be found in three dimensions, where we consider the moment induced about the line connecting two neighboring footholds $\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i$, resulting in the condition $m\mathbf{p}_{ij}^T \cdot [(\mathbf{p}_B - \mathbf{p}_i) \times \mathbf{a}_B] \leq 0$, or, more compactly

$$\det(\mathbf{p}_{ij}, \mathbf{p}_B - \mathbf{p}_i, \mathbf{a}_B) \leq 0. \quad (43)$$

The formulation (43) is more conservative than Proposition 3 because it restricts the GIA vector to the largest inscribing convex cone $\tilde{\mathcal{C}}$ of $\mathcal{C}$. This cone satisfies $\mathcal{C} = \tilde{\mathcal{C}}$ (for $N \leq 3$) or $\mathcal{C} \subseteq \tilde{\mathcal{C}} \cup \emptyset$ (for $N > 3$). Fig. 8 visualizes three possible geometries of the inscribing cone for a full stance phase, i.e., $N = 4$.

By comparing (43) with (17b), we can now see that a nonzero angular momentum derivative can be observed as a change of shape and size of the GIAC.

B. Relation to ZMP

By definition, the ZMP is a point $\mathbf{p}_{\text{ZMP}}(\mathbf{O}, \mathbf{n})$ where the moment caused by the contact wrench aligns with the unit vector $\mathbf{n}$ [43]. If the moment is computed w.r.t. to a point $\mathbf{O}$, then $\mathbf{p}_{\text{ZMP}}(\mathbf{O}, \mathbf{n})$ is located on the plane $\Pi(\mathbf{O}, \mathbf{n})$ which contains $\mathbf{O}$ and is orthogonal to $\mathbf{n}$. Without loss of generality, we can assume $\mathbf{O} = \mathbf{O}_W$ and $\mathbf{n} = \mathbf{e}_z$. For the time being, we omit the rate of change of the angular momentum, then [47]

$$\mathbf{p}_{\text{ZMP}} := \frac{\mathbf{e}_z \times (\mathbf{p}_B \times \mathbf{a}_B)}{\mathbf{e}_z^T \mathbf{a}_B}. \quad (44)$$

A simple computation reveals that $\mathbf{p}_{\text{ZMP}}$ is the projection of the base $\mathbf{p}_B$ along the vector $\mathbf{a}_B$ onto the ground plane. We write this projection as $\mathcal{P}_{\mathbf{a}_B}(\mathbf{p}_B)$ with $\mathcal{P}_{\mathbf{a}_B} : \mathbb{R}^3 \rightarrow \Pi(\mathbf{O}_W, \mathbf{e}_z)$.

Proof: The point (44) can be written as

$$\mathbf{p}_{\text{ZMP}} = \frac{\mathbf{p}_B(\mathbf{e}_z^T \mathbf{a}_B) - \mathbf{a}_B(\mathbf{e}_z^T \mathbf{p}_B)}{\mathbf{e}_z^T \mathbf{a}_B} = \mathbf{p}_B - \mathbf{a}_B \frac{\mathbf{e}_z^T \mathbf{p}_B}{\mathbf{e}_z^T \mathbf{a}_B}. \quad (45)$$

The intersection of the ray, extending $\mathbf{p}_B$ along $\mathbf{a}_B$, with the plane $\Pi(\mathbf{O}_W, \mathbf{e}_z)$ satisfies the equation $\mathbf{e}_z^T(\mathbf{p}_B + \mathbf{a}_B\gamma) = 0$. Solving for γ , the intersection can be found as

$$\mathcal{P}_{\mathbf{a}_B}(\mathbf{p}_B) = \mathbf{p}_B + \mathbf{a}_B\gamma = \mathbf{p}_B - \mathbf{a}_B \frac{\mathbf{e}_z^T \mathbf{p}_B}{\mathbf{e}_z^T \mathbf{a}_B}. \quad (46)$$

□

This relation amounts to the following proposition:

Proposition 4: On flat ground, Proposition 3 equals the ZMP stability criterion, i.e., a motion of a robot is stable if $\mathbf{p}_{\text{ZMP}} \in \mathcal{S}$ with $\mathcal{S}$ being the support polygon.

Proof: Proposition 3 can be written in span representation as

$$\mathbf{p}_B + \mathbf{a}_B = \sum_i \lambda_i (\mathbf{p}_i - \mathbf{p}_B), \quad \lambda_i \geq 0. \quad (47)$$

The projection of the left-hand side of (47) on the ground was shown to yield $\mathbf{p}_{\text{ZMP}}$. Because each point along the ray $\mathbf{p}_B + \mathbf{a}_B\gamma$ is contained in the GIAC, and $\mathbf{p}_{\text{ZMP}}$ is part of that ray, we can write

$$\mathbf{p}_{\text{ZMP}} = \sum_i \tilde{\lambda}_i \mathcal{P}_{\mathbf{p}_i - \mathbf{p}_B}(\mathbf{p}_B), \quad \lambda_i \neq \tilde{\lambda}_i \geq 0. \quad (48)$$

On flat ground we have that $\mathcal{P}_{\mathbf{p}_i - \mathbf{p}_B}(\mathbf{p}_B) = \mathbf{p}_i$, and the set spanned by the right-hand side of (48) simplifies to the convex hull of contacts, conventionally termed *support polygon*. □

The concept of ZMP can be extended to also account for the angular momentum. The resulting model is the equivalent of an inverted pendulum with an attached flywheel and its ZMP can be computed as

$$\begin{aligned} \tilde{\mathbf{p}}_{\text{ZMP}} &= \frac{\mathbf{e}_z \times (\mathbf{p}_B \times m\mathbf{a}_B - \dot{\mathbf{L}}_B)}{m\mathbf{e}_z^T \mathbf{a}_B} \\ &= \frac{\mathbf{e}_z \times (\mathbf{p}_B \times \mathbf{a}_B)}{\mathbf{e}_z^T \mathbf{a}_B} - \frac{\mathbf{e}_z \times \dot{\mathbf{L}}_B}{m\mathbf{e}_z^T \mathbf{a}_B}. \end{aligned} \quad (49)$$

Hence, the additional angular momentum derivative leads to a shift of the ZMP while leaving the support polygon unchanged.

On flat ground, the ZMP is typically computed w.r.t. the ground plane, while rough terrains require the definition of some *virtual plane* [5], [43], [62]. The notion of ZMP can, thus, be easily extended to any terrain geometry; however, the generalization of the support polygon is not straightforward. Sugihara *et al.* [62] introduced virtual contact locations as the intersection of the virtual plane with lines, connecting the COM and the footholds. Harada *et al.* [63] derived a projection from a convex 3-D support volume to the virtual plane. Caron *et al.* [43] introduced a method for projecting the entire CWC, thereby also accounting for pull contacts and frictional constraints. In contrast to those methods, the GIAC model does not depend on rigorous, geometric construction of the support area.

A different line of research has tried to simplify the projection by choosing a virtual plane that best matches the contact locations. Sato *et al.* [64] constructed the support area by heuristically predicting the COM trajectory, and Bellicoso *et al.* [5] projected the footholds vertically on the virtual plane. These methods produce a support area independent of the COM but introduce a projection error that grows with prediction uncertainty or terrain nonlinearity, respectively. On the other hand, the GIAC model does not depend on the definition of a virtual plane and does not produce a projection error.

C. Relation to Wrench Models

If each contact force satisfies the no-slip condition (6b), then the CW w^c lies in a 6-D CWC, encompassing projections of all friction cones via the mapping defined by w^c in (1). By linearizing the friction cones, the CWC becomes a convex polyhedral cone, which can be written as [43]

$$\mathcal{C}^{\text{CWC}} = \left\{ \sum_{i,j} \lambda_{ij} \begin{bmatrix} e_{ij} \\ p_i \times e_{ij} \end{bmatrix} \mid \lambda_{ij} \geq 0 \right\}. \quad (50)$$

This form is called the *span* or *ray* representation of the cone. The unit vector e_{ij} denotes the j th facet of the i th friction pyramid. According to polyhedral cone theory [48], it is possible to eliminate the slack variables λ_{ij} and write (50) in the so called *face* or *half-space* representation, e.g., by leveraging numerical tools such as the DDM

$$\mathcal{C}^{\text{CWC}} = \underbrace{\{Vz \mid z \geq 0\}}_{\text{span}(V)} \xLeftrightarrow{\text{DDM}} \mathcal{C}^{\text{CWC}} = \underbrace{\{x \mid Ux \leq 0\}}_{\text{face}(U)}. \quad (51)$$

The face representation is preferred in TO methods as it constitutes a minimal representation of the cone.

Due to the inherent relation $w^{\text{gi}} + w^c = 0$ (1), both gravito-inertia and contact wrench belong to the same cone, i.e.,

$$w^c \in \mathcal{C}^{\text{CWC}} \iff w^{\text{gi}} \in \mathcal{C}^{\text{CWC}}. \quad (52)$$

Iff

$$w^{\text{gi}} \in \mathcal{C}^{\text{CWC}} \quad (53)$$

then the resulting motion is *weak-contact stable* [46].

Definition 2 (Weak-contact stability [65]): A motion is said to be weak-contact stable iff there exists a set of contact forces $\{f_i\}_{1,\dots,N}$ s.t. the EOM (1) are satisfied and the contact forces are bounded by its Coulomb friction cones (2).

Weak contact stability has led to a new branch of research in the TO-community—namely, the *wrench models*. Related methods follow a five-staged MMO structure as follows:

- 1) foothold selection;
- 2) geometric construction of the CWC;
- 3) conversion of the geometric cone to inequality constraints;
- 4) motion optimization subject to the stability criterion (53);
- 5) computation of joint torques that realize stable contact forces.

Step 4) can be seen as the bottleneck because the conversion is of pure algorithmic nature, and footholds need to be provided as known parameters.

A similar (perhaps less general) conclusion holds for the GIAC model. From Propositions 1 and 2, we can deduce that:

Proposition 5 (Generalization of GIAC stability) If:

- 1) the GIAC constraints (17a) to (17f) are satisfied;
- 2) the rate of change of the angular momentum has a negligible effect on the contact forces;
- 3) the torso is overhanging;
- 4) the ground is flat (and tilted), or can be segmented into planes perpendicular to gravity,

then the motion of the robot is weak-contact stable.

We note that Proposition 5 guarantees weak-contact stability of our model under Assumption III-B.

Apart from being described by SRBD, wrench models assume only polyhedral Coulomb friction. Even if Assumption III-B is satisfied, CWC constraints may appear less conservative than GIAC constraints. This can be seen by Proposition 3 or 1 and 2, which hold with “if” but not with “iff.” The additional assumption of flat ground leads to a special case, for which the propositions hold with “iff.”

APPENDIX B

We consider a constrained NLP (28) and impose the limitation that the objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ can be factorized with some vector-valued functions $g_i : \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$f_i(x) = 0.5g_i(x)^T W_i g_i(x), \quad W_i = W_i^T \succ 0. \quad (54)$$

By approximating g_i by its first-order Taylor expansion about x_0 , i.e., $g_i(x) \approx g_i(x_0) + \nabla g_i^T(x_0)(x - x_0)$, we can compute the Jacobian and approximate Hessian of the objective as

$$J_i(x_0) = \nabla g_i(x_0)^T W_i g_i(x_0) \quad (55)$$

$$H_i(x_0) = \nabla g_i(x_0)^T W_i \nabla g_i(x_0). \quad (56)$$

It is easy to verify that the Hessian matrix (56) is always positive semidefinite. This technique is generally known as the *Gaussian–Newton* approximation.

REFERENCES

- [1] D. Bellicoso *et al.*, “Advances in real-world applications for legged robots,” *J. Field Robot.*, vol. 35, no. 8, pp. 1311–1326, 2018.
- [2] M. Tranzatto *et al.*, “CERBERUS in the DARPA Subterranean Challenge,” *Sci. Robot.*, vol. 7, no. 66, 2022, Art. no. eabp9742, doi: [10.1126/scirobotics.abp9742](https://doi.org/10.1126/scirobotics.abp9742).
- [3] M. Hutter *et al.*, “Anymal - a highly mobile and dynamic quadrupedal robot,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 38–44.
- [4] H. Mobahi and J. W. Fisher, “On the link between gaussian homotopy continuation and convex envelopes,” in *Energy Minimization Methods in*

- Computer Vision and Pattern Recognition*, X.-C. Tai, E. Bae, T. F. Chan, and M. Lysaker, Eds., Cham, Switzerland:Springer, 2015, pp. 43–56.
- [5] C. D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, “Dynamic locomotion and whole-body control for quadrupedal robots,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 3359–3365.
 - [6] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, “Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2261–2268, Jul. 2018.
 - [7] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “Mit cheetah 3: Design and control of a robust, dynamic quadruped robot,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2245–2252.
 - [8] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, “Dynamic locomotion in the mit cheetah 3 through convex model-predictive control,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.
 - [9] R. Orsolino, M. Focchi, C. Mastalli, H. Dai, D. G. Caldwell, and C. Semini, “Application of wrench-based feasibility analysis to the online trajectory optimization of legged robots,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3363–3370, Oct. 2018.
 - [10] P. Fernbach, S. Tonneau, O. Stasse, J. Carpentier, and M. Taïx, “C-CROC: Continuous and convex resolution of centroidal dynamic trajectories for legged robots in multicontact scenarios,” *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 676–691, Jun. 2020.
 - [11] P. Fernbach, S. Tonneau, and M. Taïx, “CROC: Convex resolution of centroidal dynamics trajectories to provide a feasibility criterion for the multi contact planning problem,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.
 - [12] P. Fankhauser, M. Bjelonic, C. Dario Bellicoso, T. Miki, and M. Hutter, “Robust rough-terrain locomotion with a quadrupedal robot,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 5761–5768.
 - [13] O. Magaña *et al.*, “Fast and continuous foothold adaptation for dynamic locomotion through CNNs,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2140–2147, Apr. 2019.
 - [14] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter, “Perceptive locomotion in rough terrain – online foothold optimization,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5370–5376, Oct. 2020.
 - [15] D. Kim *et al.*, “Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 2464–2470.
 - [16] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, “A control architecture for quadruped locomotion over rough terrain,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 811–818.
 - [17] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, “Fast, robust quadruped locomotion over challenging terrain,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2665–2670.
 - [18] A. Winkler *et al.*, “Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6476–6482.
 - [19] C. Mastalli, I. Havoutis, A. W. Winkler, D. G. Caldwell, and C. Semini, “On-line and on-board planning and perception for quadrupedal locomotion,” in *Proc. IEEE Int. Conf. Technol. Practical Robot Appl.*, 2015, pp. 1–7.
 - [20] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, “Footstep planning for autonomous walking over rough terrain,” in *Proc. IEEE-RAS 19th Int. Conf. Humanoid Robots*, 2019, pp. 9–16.
 - [21] W.-L. Ma and A. D. Ames, “From bipedal walking to quadrupedal locomotion: Full-body dynamics decomposition for rapid gait generation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 4491–4497.
 - [22] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, “3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1447–1454.
 - [23] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, “Trajectory optimization through contacts and automatic gait discovery for quadrupeds,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1502–1509, Jul. 2017.
 - [24] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, “Gait and trajectory optimization for legged systems through phase-based end-effector parameterization,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1560–1567, Jul. 2018.
 - [25] M. Neunert, F. Farshidian, and J. Buchli, “Efficient whole-body trajectory optimization using contact constraint relaxation,” in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 43–48.
 - [26] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, “Feedback MPC for torque-controlled legged robots,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4730–4737.
 - [27] G. Bledt and S. Kim, “Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6316–6323.
 - [28] O. Melon, R. Orsolino, D. Surovik, M. Geisert, I. Havoutis, and M. Fallon, “Receding-horizon perceptive trajectory optimization for dynamic legged locomotion with learned initialization,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 9805–9811, doi: [10.1109/ICRA48506.2021.9560794](https://doi.org/10.1109/ICRA48506.2021.9560794).
 - [29] M. Neunert *et al.*, “Whole-body nonlinear model predictive control through contacts for quadrupeds,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1458–1465, Jul. 2018.
 - [30] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, “Trajectory optimization for legged robots with slipping motions,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 3013–3020, Jul. 2019.
 - [31] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, “RLOC: Terrain-aware legged locomotion using reinforcement learning and optimal control,” *IEEE Trans. Robot.*, pp. 1–20, 2022, doi: [10.1109/TRO.2022.3172469](https://doi.org/10.1109/TRO.2022.3172469).
 - [32] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, “Motion planning for quadrupedal locomotion: Coupled planning, terrain mapping, and whole-body control,” *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1635–1648, Dec. 2020.
 - [33] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, “Multi-layered safety for legged robots via control barrier functions and model predictive control,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 8352–8358, doi: [10.1109/ICRA48506.2021.9561510](https://doi.org/10.1109/ICRA48506.2021.9561510).
 - [34] O. Melon, M. Geisert, D. Surovik, I. Havoutis, and M. Fallon, “Reliable trajectories for dynamic quadrupeds using analytical costs and learned initializations,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 1410–1416.
 - [35] D. E. Orin and A. Goswami, “Centroidal momentum matrix of a humanoid robot: Structure and properties,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 653–659.
 - [36] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 295–302.
 - [37] R. Budhiraja, J. Carpentier, and N. Mansard, “Dynamics consensus between centroidal and whole-body models for locomotion of legged robots,” in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 6727–6733.
 - [38] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, “An efficient optimal planning and control framework for quadrupedal locomotion,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 93–100.
 - [39] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, “A unified MPC framework for whole-body dynamic locomotion and manipulation,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4688–4695, Jul. 2021.
 - [40] Y. Ding, A. Pandala, and H. Park, “Real-time model predictive control for versatile dynamic motions in quadrupedal robots,” in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 8484–8490.
 - [41] G. Bledt, P. M. Wensing, and S. Kim, “Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 4102–4109.
 - [42] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, “Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control,” *CoRR*, vol. abs/1909.06586, 2019, [Online]. Available: <http://arxiv.org/abs/1909.06586>
 - [43] S. Caron, Q.-C. Pham, and Y. Nakamura, “ZMP support areas for multi-contact mobility under frictional constraints,” *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 67–80, Feb. 2017.
 - [44] J. Carpentier and N. Mansard, “Multicontact locomotion of legged robots,” *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1441–1460, Dec. 2018.
 - [45] H. Dai and R. Tedrake, “Planning robust walking motion on uneven terrain via convex optimization,” in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 579–586.
 - [46] S. Caron, Q.-C. Pham, and Y. Nakamura, “Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics,” in *Proc. Robot.: Sci. System*, 2015, pp. 1–9.
 - [47] P. Sardain and G. Bessonnet, “Forces acting on a biped robot. center of pressure-zero moment point,” *IEEE Trans. Syst., Man, Cybern. - Part A: Syst. Humans*, vol. 34, no. 5, pp. 630–637, Sep. 2004.
 - [48] K. Fukuda and A. Prodon, “Double description method revisited,” in *Combinatorics and Computer M. Science*, R. Deza Euler, and I. Manoussakis, Eds., Berlin, Germany:Springer, 1996, pp. 91–111.

- [49] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *Proc. Int. Conf. Climbing Walking Robots*, 2014, pp. 433–440.
- [50] E. Arias-Castro and D. Donoho, "Does median filtering truly preserve edges better than linear filtering?," *Ann. Statist.*, vol. 37, no. 3, pp. 1172–1206, 2006.
- [51] G. Bradski, "The opencv library," *Dr. Dobbs's J. Softw. Tools*, 2000, Art. no. 2236121.
- [52] M. Gaertner, M. Bjelonic, F. Farshidian, and M. Hutter, "Collision-free MPC for legged robots in static and dynamic scenes," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 8266–8272.
- [53] C. Gehring *et al.*, "Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot," *IEEE Robot. Autom. Mag.*, vol. 23, no. 1, pp. 34–43, Mar. 2016.
- [54] Luca Di Gaspero, "QuadProg," 1998. [Online]. Available: <http://quadprog.sourceforge.net/>
- [55] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.
- [56] B. Bell, "Ccppadcodegen: Extends the cppad library in order to perform hybrid automatic differentiation by using operator-overloading and produces source-code," 2021. [Online]. Available: <https://github.com/joaoleal/CppADCodeGen/>
- [57] N. Mansard, A. DelPrete, M. Geisert, S. Tonneau, and O. Stasse, "Using a memory of motion to efficiently warm-start a nonlinear predictive controller," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2986–2993.
- [58] C. Dario Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 558–564.
- [59] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, Dec. 2017.
- [60] M. Focchi *et al.*, "Heuristic planning for rough terrain locomotion in presence of external disturbances and variable perception quality," in *Proc. Adv. Robot. Res.: Lab Market*, 2020, pp. 165–209.
- [61] G. Blede, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 4399–4406.
- [62] T. Sugihara, Y. Nakamura, and H. Inoue, "Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 2, pp. 1404–1409.
- [63] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "Dynamics and balance of a humanoid robot during manipulation tasks," *IEEE Trans. Robot.*, vol. 22, no. 3, pp. 568–575, Jun. 2006.
- [64] T. Sato, S. Sakaino, E. Ohashi, and K. Ohnishi, "Walking trajectory planning on stairs using virtual slope for biped robots," *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1385–1396, Apr. 2011.
- [65] J.-S. Prof and J. J. Trinkle, "Stability characterizations of rigid body contact problems with coulomb friction," *ZAMM – J. Appl. Math. Mechanics/Zeitschrift für Angewandte Mathematik Mechanik*, vol. 80, pp. 643–663, 2000.



Fabian Jenelten received the B.Sc. and M.Sc. degrees in mechanical engineering in 2015 and 2018, respectively, from ETH Zurich, Zurich, Switzerland, where he is currently working toward the Ph.D. degree with the robotic systems laboratory, under the supervision of Prof. M. Hutter.

His research interests include model- and learning-based control approaches for legged robot locomotion.



Ruben Grandia received the B.Sc. degree in aerospace engineering from TU Delft, Delft, the Netherlands, in 2014, and the M.Sc. degree in robotics, systems, and control in 2017 from ETH Zurich, Zurich, Switzerland, where he is currently working toward the Ph.D. degree with the robotic systems laboratory, under the supervision of Prof. M. Hutter.

His research interests include nonlinear optimal control and its application to dynamic mobile robots.



Farbod Farshidian received the M.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2012 and the Ph.D. degree from ETH Zurich, Zurich, Switzerland, in 2017.

He is currently a Senior Scientist with Robotic System Laboratory, ETH Zurich. He is a part of the National Centre of Competence in Research (NCCR) Robotics and NCCR Digital Fabrication, Zurich, Switzerland. His research interests include the motion planning and control of mobile robots, intending to develop algorithms and techniques to endow these

robotic platforms to operate autonomously in real-world applications.



Marco Hutter received the M.Sc. and Ph.D. degrees from ETH Zurich, Zurich, Switzerland, in 2009 and 2013.

He is currently an Associate Professor for Robotic Systems with ETH Zurich. He is a part of the National Centre of Competence in Research (NCCR) Robotics and NCCR Digital Fabrication, Zurich, Switzerland, and PI in various international projects (e.g., EU NI) and challenges. His research interests include the development of novel machines and actuation concepts together with the underlying control, planning, and machine learning algorithms for locomotion and manipulation.