TAMOLS

Terrain-Aware Motion Optimization for Legged Robots

MICRO-507 - Legged Robots

Frederik Kaiser
Grzegorz Pawlak
Tomasz Niedziałkowski

Executive Summary — Key challenges & contributions

Three major challenges

1. Computation vs. generality
2. Nonconvexity & local optima.
3. Perception uncertainty & occlusion

Contributions

1. Contact-force-free, differentiable stability metric (GIAC).
2. Fast TO over footholds + base pose.
3. Map extrapolation & denoising pipeline
4. Robust tracking loop.

Implementation & Validation

- Validated on ANYmal (stairs, stepping stones, gaps).
- Fixed gait pattern used to reduce problem size.

Core model: Centroidal / SRBD & GIA

From CD to SRBD and GIA

- **Centroidal Dynamics (CD)**
  - ↳ Project full robot dynamics to base position and centroidal momentum

$$\underbrace{m\begin{bmatrix} \ddot{\boldsymbol{p}}_B - \boldsymbol{g} \\ m\boldsymbol{p}_B \times (\ddot{\boldsymbol{p}} - \boldsymbol{g}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \dot{\boldsymbol{L}} \end{bmatrix}}_{\text{gravito-inertia wrench (GIW)}} = \underbrace{\sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{p}_i \times \boldsymbol{f}_i \end{bmatrix}}_{\text{contact wrench (CW)}}$$

- **Single-Rigid-Body approximation (SRBD)**
  - ↳ Neglect limb inertia and model the robot as one virtual rigid body plus contact wrenches.

- **Gravitio-inertia acceleration (GIA)**
  - ↳ $a_B = g - \ddot{p}_B$. captures the direction in which the base accelerates relative to gravity

Gravitio-inertia acceleration-Cone (GIAC)

Definition:

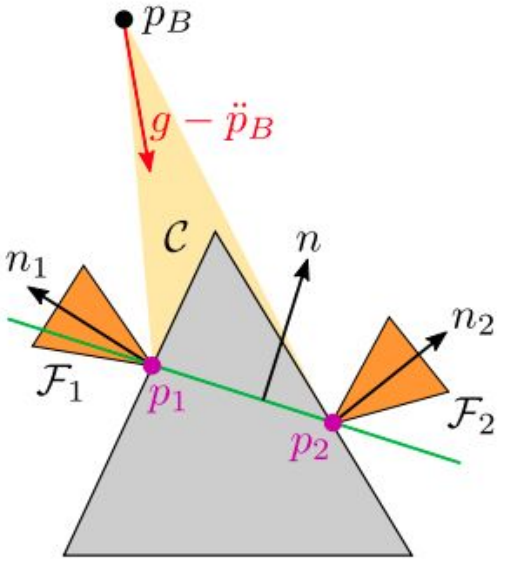
- GIAC $\mathcal{C}$ is the convex cone spanned by rays from the base $\boldsymbol{p}_B$ to the footholds $\boldsymbol{p}_i$

Intuition:

- If the GIA vector lies inside this cone, no net tipping moment occurs at any support edge under our model assumptions.

Why GIAC?

- Geometric balance check on GIA instead of optimising contact forces
    ↳ cheap and differentiable



Model Assumptions:

1. Single Body .
2. Small angular-momentum influence.
3. Push-only contacts
4. Coulomb friction Model
5. Horizontal contact planes
6. Overhanging torso

Motion parameterization & NLP

Base pose parameterization

How the body (base) motion is represented over time?

The **time horizon** is split into **contact phases** (set of legs on the ground is constant)

The **base position (pos + orientation)** is described by 6-D, 5th-order **polynomials (quintic),** one per phase.

Footholds are the optimized 3D landing positions for the swing legs, chosen continuously on the **terrain map.**

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

for $t \in (0, \tau_k)$. Stack spline coefficients + footholds p_i + slacks into the decision vector x.

$$\min_{\boldsymbol{x}} \sum_i f_i(\boldsymbol{x}) \qquad \text{(NLP)}$$

$$\text{s.t.} \quad \boldsymbol{c}_{\text{eq}}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{c}_{\text{ineq}}(\boldsymbol{x}) \leq \boldsymbol{0}$$

Nonlinear Program (NLP)

What is optimized and under what rules?

Variables

- Base trajectory parameters
- Foothold landing points
- Slack variables

Objective terms

- Track desired velocity / angular
- Prefer footholds on flat, safe terrain (avoid edges, use hs_1 + hs_2 maps)
- Keep motion smooth (low jerk, low momentum rate)
- Maintain nominal leg posture

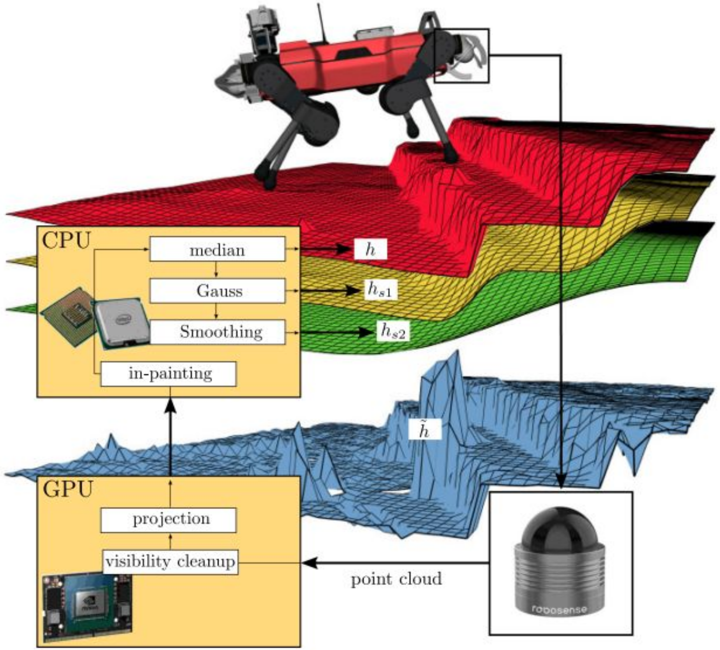
Constraints:

- GIAC stability
- Friction & no-slip
- Kinematic reach limits
- Convex footprint & leg-collision avoidance

Solver:
Fast SQP with automatic differentiation that converges in 1–2 iterations (~6 ms)

Perception: elevation mapping and layers

Two LiDARs provide a raw elevation map at ~20 Hz. The **GPU** performs projection and visibility cleanup, and the **CPU** applies median filtering, Gaussian smoothing, and in-painting. The result is a set of **three terrain layers** used by the optimizer.



Three elevation layers:

- **h** – the cleaned height map, preserving real terrain geometry.
- **hs$_1$** – slightly smoothed version used to compute gradients for edge avoidance and foothold centering.
- **hs$_2$** – the heavily smoothed *virtual floor*, generated through dilation and larger-scale filtering, used for robust initialization.

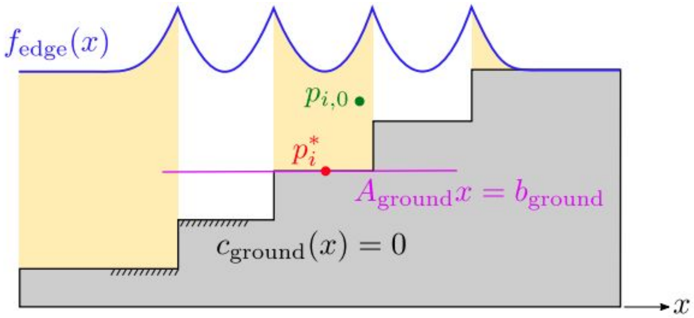
These layers provide both accurate terrain information and simplified surfaces that make **optimization more stable**.

Initialization: batch search & graduated optimization

Why initialization matters?

Terrain **geometry** and foothold **objectives** make the NLP **highly nonconvex.**

Naïve guesses often fall into poor **local minima**, like stepping on stair edges and failing to climb.



Graduated optimization

Three stages:

1. The optimizer solves a **coarse** problem on the **virtual floor (hs_2)**, which removes most terrain discontinuities and gives a near-convex initialization for the **base trajectory**.

2. Footholds are refined with a batch search on the **real height map (h)**, penalizing occlusions, steep gradients from hs_1, and unsafe edges.

3. This refined trajectory is used to warm-start the full NLP on the **true terrain**, enabling fast convergence and reliable solutions on rough environments.

Whole Body Control - Tracking

- WBC converts trajectories (body, legs) into joint commands solving hierarchical optimization problem
- Tasks are categorized into 3 blocks:
  a. physical constraints
  b. stable contact and tracking the reference trajectories in task space
  c. low-priority tasks to completely resolve the null-space
- Developed in another ETH project

TASK PRIORITIES USED FOR WBC

task	type	priority
equations of motion	$=$	0
joint torque limits	$\leq$	0
kinematic limits	$\leq$	1
friction pyramid (stance legs)	$\leq$	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

Whole Body Control - Disturbance Rejection

Generalized Momentum (GM) observer – momentum integrating structure that estimates the external forces using only the measurements $\{q, u, \tau\}$

$$F_{ext} = [f_B \, \tau_B \, f_1 \dots f_N] \in \mathbb{R}^{6+3N}$$

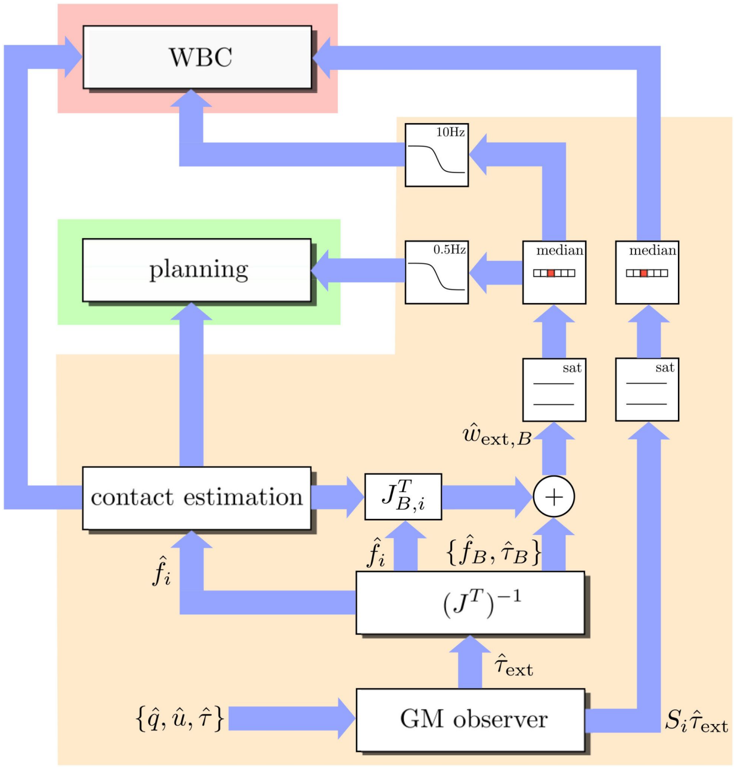
$$\hat{\tau}_{ext} = GM\left(q, u, \tau\right) = J(q)^T \hat{F}_{ext} \in \mathbb{R}^{n+6}$$

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

$\hat{\tau}_{ext}$ – disturbances in **joint** space (n joints + 6 base DOF)
$\hat{w}_{ext,B}$ – disturbances in **task** space (6 base DOF)

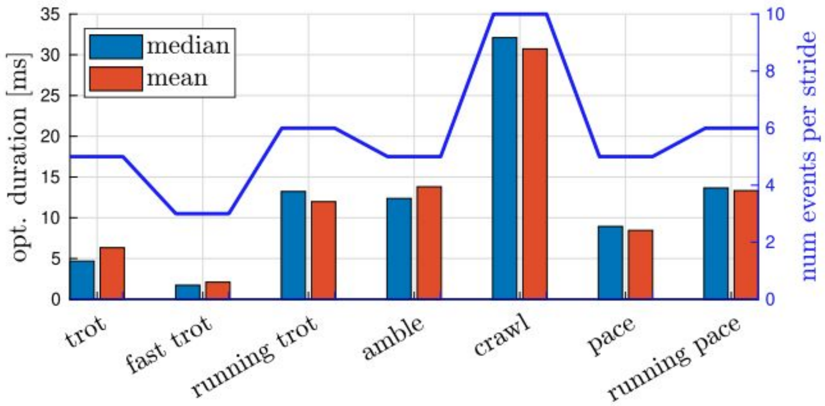
$$S_B \tilde{h}\left(q, u\right) = S_B h\left(q, u\right) - \hat{w}_{ext,B}$$

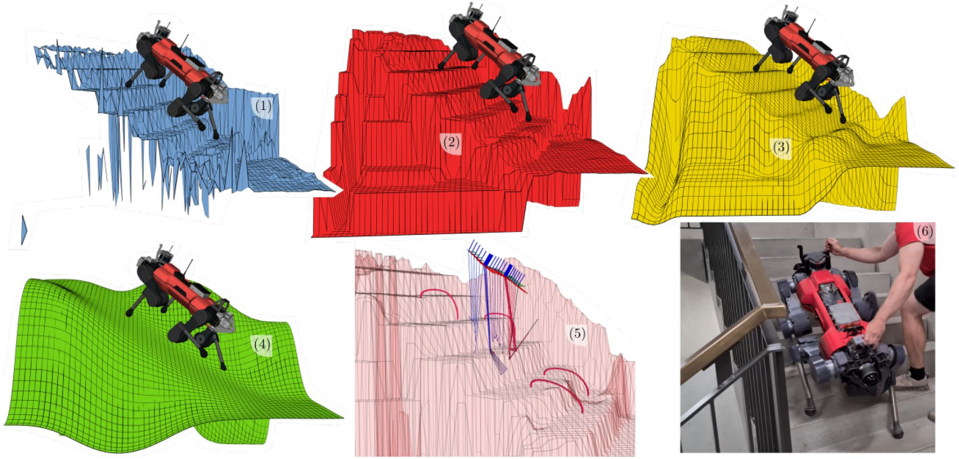
$$S_i \tilde{h}\left(q, u\right) = S_i h\left(q, u\right) - S_i \tilde{\tau}_{ext} \; \forall \, \text{swinging} \, i$$



Results

- Tested both in simulation and real world
- Reliable locomotion on:
  a. stairs
  b. gaps
  c. stepping stones
- Trot gait average **optimization duration ≈ 6.3 ms** (one stride horizon). **48 times faster than SOTA** CMO pipeline, 2.9 times slower than MMO





Pros

Real-time performance

Hierarchical optimization
Using a smooth heightmap first helps avoid local minima

Integrated disturbance rejection
Provided by GM observer

GIAC
Contact-force free differentiable stability criterion

Cons

Modeling approximation
Accepting only horizontal footholds

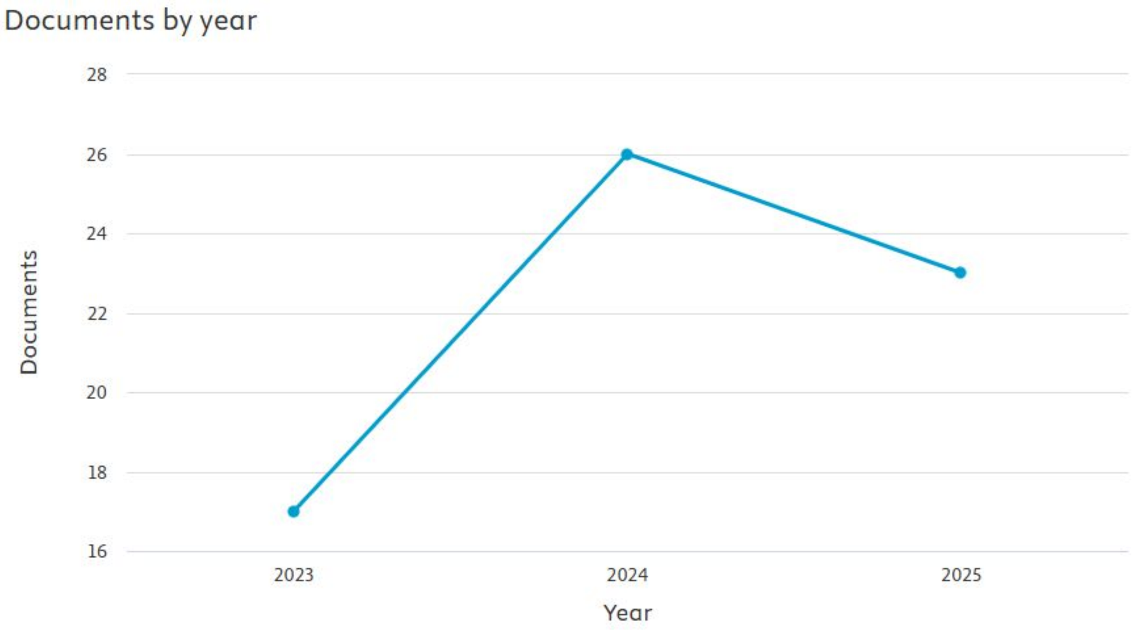
Simplified kinematic constraints
Knee joint collisions still remain a problem

Quality of the elevation maps
State estimator drift shifts the map relative to odometry → mismatch between realized and planned foot locations

EPFL

Citations

Total citations: 66



Citations

Other groups typically use TAMOLS in one of three ways:

Baseline	Reference trajectory generator	Conceptual inspiration

Important follow-up work

Jenelten, F., He, J., Farshidian, F., & Hutter, M. (2024).
DTC: Deep Tracking Control.
Science Robotics, 9(86).
https://doi.org/10.1126/scirobotics.adh5401

Thank you for your attention!

Any questions?