Zero-order optimization with contact priors for locomotion

Victor Dhédin[1], and Majid Khadiv[1]

***Abstract*— In this work, we introduce a contact-explicit trajectory optimization framework adapted for zero-order optimization methods. Our approach optimizes state–control trajectories for predefined full-horizon contact sequences (including both contact status and contact surface). We demonstrate the effectiveness of our method on multiple quadruped locomotion tasks. Even though zero-order methods do not require any contact information, we show that explicitly providing it significantly improves performance in challenging scenario, such as clearing a long gap with two lateral walls support. This approach is inherently parallelizable, which opens the door to large-scale parallel data collection for contact-rich tasks, an important direction given recent advances in imitation learning.**

I. Introduction

Recent progress in fast parallel simulations has changed the legged locomotion field in the past few years. Deep Reinforcement Learning (DRL) is currently the most popular approach, achieving state-of-the-art results in dynamic locomotion, even in highly constrained environments [1]. However, such results often come at the expense of tedious reward shaping, carefully designed curriculum, and very long training time.

More recently, sampling-based controllers (SBC) [2], [3], [4] have gained interest in the community, mainly because of their simplicity. Nonetheless, they are intrinsically limited when applied to embodiments with numerous degrees of freedom or for long-horizon tasks, due to the large dimensionality of the sampling space [5]. Still, [6] recently showed real-time whole-body control on quadrupedal locomotion tasks and promising results on a humanoid robot.

Contact interaction is at the heart of any locomotion problem. Yet, both DRL and SBC treat contact implicitly, as contact dynamics are handled by a black-box simulator. Hence, it is common practice to specify hand-defined reward/cost terms that help the algorithms make the right contacts. Approaches considering contacts in a more explicit manner have other advantages. They can potentially generalize better across tasks (as many tasks can be represented by different contact sequences), which is still a challenge for traditional DRL policies. This also allows them to capture multi-modal behaviors, since multiple contact sequences may solve the same task. Such flexibility is crucial for adapting to the diverse situations encountered in the real world.

As important as it may be, leveraging contact information with DRL or SBC can be done in many different ways and is still an open research question. [6], [7] considered a reward term to encourage foot clearance. [7] additionally specified a binary contact reward to realize a given gait pattern. [8], [9] added reward terms to reach some desired feet positions.

In this paper, we introduce a new contact-explicit framework based on a sampling-based trajectory optimization (also referred to as zero-order optimization later). In the proposed formulation, we treat contacts as discrete variables describing the contact status and contact surface (patch) of each end-effector at each time step. We believe that this provides enough information on how to solve the desired task while not constraining too much the optimization of the trajectories.

By using modern simulators as black boxes, our method enables whole-body trajectory planning with realistic dynamics and full collision models. We focus specifically on solving the trajectory optimization (TO) problem for a given candidate contact sequence. Planning for both the trajectories and contact variables has been addressed in many other works [10], [11], [12], [13]. More specifically, in our previous work [14], we showed that Monte Carlo Tree Search (MCTS) can plan all contact variables given a TO solver optimizing the trajectories. Since the proposed zero-order solver could easily be integrated in this framework, we considered the full contact planning problem less compelling for this work and focus only on the TO part.

The main contributions of this work are summarized below:

- We propose a contact-explicit TO formulation tailored for zero-order optimization techniques. The optimization process results in a state-control trajectory realizing a predefined contact sequence.
- We demonstrate that our framework can execute a wide variety of contact sequences on a quadruped robot, including a highly dynamic locomotion task.
- We show for a locomotion task in sparse environment that, although zero-order methods do not require contact information, providing it is crucial for successfully solving the task.

The rest of the paper is structured as follows. In Section II, we present the fundamentals about zero-order optimization. In Section III, we present our method. In Section IV, we

This work was partially supported by the Huawei-TUM joint laboratory- Individual Project Agreements TC20231218038-2025-2 and TC20231218038-2025-3

[1]Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich (TUM), Germany. firstname.lastname@tum.de

show the results and discuss them. Finally, in Section V, we conclude our findings and outline the potential future research directions

II. Zero-order optimization

Zero-order optimization algorithms address the following generic optimization problem $\min_{x\in\mathbb{R}^n} f(x)$ by only pointwise evaluating a known function f to be minimized. The gradient of f is not required which makes it popular for non-smooth non-convex optimization.

For trajectory optimization, zero-order optimization is usually used in a single shooting scheme, where the objective is to find a control sequence $\{\mathbf{u}_0, \dots, \mathbf{u}_{T-1}\}$ minimizing a cost function J while satisfying the system's dynamics $\mathbf{x}_{t+1} = f_{dyn}(\mathbf{x}_t, \mathbf{u}_t)$, as formulated below:

$$\begin{aligned} \min_{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{T-1}} \quad & J(\mathbf{x}_{0:T}, \mathbf{u}_{0:T-1}) \\ \text{s.t.} \quad & \mathbf{x}_0 = \mathbf{x}, \;\; \mathbf{x}_{t+1} = f_{\text{dyn}}(\mathbf{x}_t, \mathbf{u}_t). \end{aligned} \tag{1}$$

$\mathbf{x}_t = (\mathbf{q}_t, \mathbf{v}_t)$ and $\mathbf{u}_t$ denotes respectively the state (joint positions and velocities) and control of the system at time t. To satisfy the dynamics constraint, control sequences $\mathbf{u}_{0:T-1}$ are rolled-out from the initial state $\mathbf{x}_0$ using a black-box simulator, which ultimately outputs state trajectories $\mathbf{x}_{0:T}$ needed to evaluate the cost J.

The most commonly used algorithms in robotics are Cross-Entropy Method (CEM) [15] [16] and Model Predictive Path Integral (MPPI) [17]. Both algorithms rely on similar mechanisms as outlined in algorithm 1. In the literature, those algorithms are mostly used in a receding horizon fashion, warm-starting from the solutions of the previous time-step.

Algorithm 1 Zero-order Optimization in Robotics, Skeleton

Inputs $\boldsymbol{\mu} \in \mathbb{R}^{T\cdot n_u}, \boldsymbol{\Sigma} \in \mathbb{R}^{T\cdot n_u \times T\cdot n_u}$, N samples, K iterations
for iteration $k = 1, 2, \dots, K$ **do**
  Sample N control sequences $\{\mathbf{u}^{(i)}_{0:T-1}\}_{i=1}^N \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
  for each sample i **do**
    Roll out dynamics: $\mathbf{x}^{(i)}_{t+1} = f_{\text{dyn}}(\mathbf{x}^{(i)}_t, \mathbf{u}^{(i)}_t)$
    Evaluate cost: $J^{(i)} = J(\mathbf{x}^i_{0:T}, \mathbf{u}^i_{0:T-1})$
  end for
  Update parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ using $\{J^{(i)}\}_{i=1}^N$
  e.g. CEM: average over the elite-set, MPPI: exponentially weighted average
end for
Return best control sequence $\mathbf{u}^*_{0:T-1}$

Recent works [18] [6] have shown theoretical links between an iteration of MPPI and a gradient descent step on a smoothed version of the cost function J, which could explain part of its effectiveness. Additionally MPPI can be seen as a special case of Consensus Based Optimization [19], which have theoretical guaranties to converge at the global optimum under mild assumptions. This motivates the use of sampling-based techniques as an optimizer in a TO setting.

III. Method

In this section, we present our framework, which uses zero-order optimization to plan trajectories to realize a given contact sequence. For clarity, we first present our contact planning formalism in Section III-A. We then explain how the contact constraints are handled by the zero-order optimization process in Section III-B. Finally we present our zero-order optimizer in Section III-C.

A. Contact planning formalism

We note in the following a contact plan $\mathcal{C}$ as a sequence of contact modes $\mathsf{M}_0 \rightarrow \dots \mathsf{M}_t \cdots \rightarrow \mathsf{M}_T$. At each time step t, the contact mode specifies which end-effectors are in contact and, for those that are, their corresponding contact patches. Each end-effector $i \in [\![1, n_{\text{ee}}]\!]$ has a binary contact variable $\mathsf{c}_{t,i}$, equal to 1 if the end-effector is in contact and 0 otherwise. The set of end-effectors that are in contact at time t is denoted by $\mathcal{C}_t$. For every $i \in \mathcal{C}_t$, we assign a contact patch $\mathsf{p}_{t,i} \in [\![1, n_{\mathsf{p}}]\!]$. The contact mode is then written as

$$\mathsf{M}_t \;=\; ((\mathsf{c}_{t,i})_{i\in[\![1,n_{\text{ee}}]\!]},\; (\mathsf{p}_{t,i})_{i\in\mathcal{C}_t}). \tag{2}$$

B. TO formulation

Evaluating if the end-effectors made contact with the correct patches at the correct time steps from the trajectory roll-outs can be done in practice using the state trajectory $\mathbf{x}_{0:T}$ and forward kinematics. Since simulators have to detect contacts to simulate the robot, one could also directly access the roll-outs simulation data (if available). We define an indicator function $I_{\mathcal{C}}(t, i, \mathbf{x}_t)$ for this purpose. $I_{\mathcal{C}}(t, i, \mathbf{x}_t)$ is equal to 0 if the end-effector i is in contact with the correct patch $\mathsf{p}_{t,i}$ imposed by the mode M_t at time step t, and to 1 otherwise. Contact constraints are then satisfied when the following condition is verified:

$$\iota_{\mathcal{C}}(\mathbf{x}_{0:T}) := \sum_{t=0}^{T} \sum_{i=1}^{n_{\text{ee}}} I_{\mathcal{C}}(t, i, \mathbf{x}_t) = 0 \tag{3}$$

One way to handle the contact constraints could be to reject samples not realizing the desired contact sequence. However, this would make the process very inefficient in high-dimensional setting. To circumvent this, we relax the problem by considering the contact constraint term 3 as an additional cost term, as defined in the following optimization problem:

$$\begin{aligned} \min_{\mathbf{u}_0, \dots, \mathbf{u}_{T-1}} \quad & J(\mathbf{x}_{0:T}, \mathbf{u}_{0:T-1}) + w_{\mathcal{C}}\iota_{\mathcal{C}}(\mathbf{x}_{0:T}) \\ \text{s.t.} \quad & \mathbf{x}_0 = x, \;\; \mathbf{x}_{t+1} = f_{\text{dyn}}(\mathbf{x}_t, \mathbf{u}_t) \end{aligned} \tag{4}$$

where $w_{\mathcal{C}} \in \mathbb{R}$ scales the importance of the contact constraint in the optimization. Note that with this formulation, the 3D contact location within a patch is unconstrained and is free to be optimized by the algorithm.

C. Zero-order optimizer

We solve 4 using CEM as an optimizer. Controls $\mathbf{u}_{0:T-1}$ correspond to a joint position target for a low level PD policy. Following [16], we use momentum to update μ and Σ (noted β_μ and β_Σ) and scale controls to lie inside the permitted action interval. Moreover, we used cubic spline interpolation to reduce the sampling space dimension. For each joint, the full control trajectory is computed from $s \leq T$ equally time spread control knots, reducing the sampling space dimension to $s \cdot n_u$.

IV. Results

We evaluate our method on a variety of locomotion tasks with a quadruped robot. We present our experimental setup in Section IV-A. In Section IV-B, we demonstrate the realization of different gait patterns. We then show in Section IV-C that our framework can also handle more challenging scenarios, such as clearing a gap with lateral support walls.

A. Experimental setup

We used MuJoCo XLA (MJX) [20] to simulate roll-outs in parallel on a NVIDIA GeForce RTX 4060 GPU. One can access at each time step all the geometry pairs in contact with MuJoCo, we relied on this to compute $I_{\mathcal{C}}(t, i, \mathbf{x}_t)$. We used a Go2 quadruped platform for our experiments. Torques are computed from a low-level PD policy with $K_p = 30$ and $K_d = 1$. We considered $T = 200$ discretization nodes in the optimization problem. Roll-outs are performed with a simulation timestep of 0.01 second, making each trajectories 2 seconds long. Splines are computed from $s = 20$ knots. For all experiments, we used the same CEM parameters, as detailed in Table I.

TABLE I

CEM Hyperparameters

Parameters	Value
Number of samples N	1024
Elite set proportion ρ	0.05
Mean momentum β_μ	0.99
Covariance momentum β_Σ	0.3

B. Gaited locomotion

We first evaluate the algorithm on a simple task: locomotion with a predefined gait. Three gaits are considered: trot, pronk, and bound. For all of them, we used the same cost function J. The cost J is defined as the sum of a running cost for all $t < T$ and a terminal cost at $t = T$, where each term is a weighted sum of ℓ_2 norms between state or control variables and their reference values (see Table II). We set the desired velocity to $v_{\text{des}} = [0.5, 0, 0]$ and run the algorithm for $K = 200$ iterations.

Results on Figure 1 show that the realized contact sequence is very similar to the one planed for all the gaits. The mean contact error, defined as $\frac{1}{T \cdot n_{ee}} \iota_{\mathcal{C}(\mathbf{x}_{0:T})}$, equals 0.02, 0.02 and 0.06 for the trot, pronk and bound gait respectively.

C. Clearing a gap with lateral wall support

This task consist of clearing a gap of length $l_{gap} = 0.75$m by using two lateral walls of $\alpha = 45°$ inclination. The system can make contact with 4 patches: the plane on which the robot starts, the two walls and the goal plane to reach across the gap.

TABLE II

Cost terms for trajectory optimization

Quantity	Reference	Weight value
Running costs		
Base height z	h_{nom}	5
Base orientation q	q_{upright}	10
Joint positions q_j	$q_{j,\text{nom}}$	5
Base lin. vel. v	v_{des}	$[5, 1, 1]$
Base ang. vel. ω	$w_{des} = 0$	1
Terminal costs		
Base position p_T	$p_0 + v_{\text{des}} Tdt$	200
Base orientation q_T	q_{upright}	20
Joint positions $q_{j,T}$	$q_{j,\text{nom}}$	10
Regularization		
Joint velocities $\dot{q}_j$	0	10^{-2}
Controls u	0	10^{-2}
Contacts		
Contact constraint $w_{\mathcal{C}}$	$\mathcal{C}$	7.5

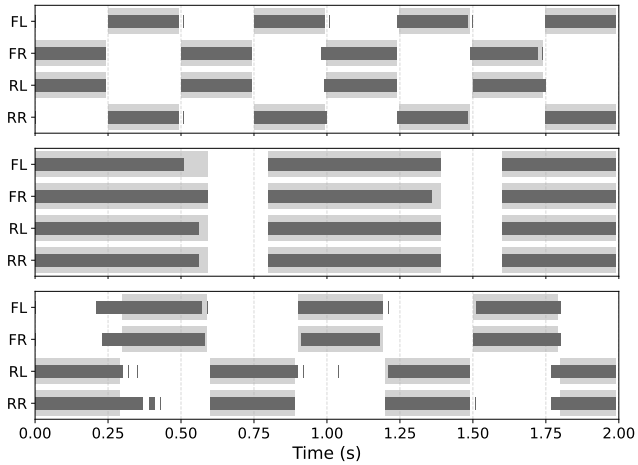

Fig. 1. Comparison of the planned and realized contact sequence for the different gaits (top: trot, middle: pronk, bottom: bound). The planned sequence is colored in light gray, the realized one in dark gray.

We prespecified a contact sequence for this task and let the algorithm run for $K = 1000$ iterations. We used a similar cost function as in Section IV-B. We also set a forward desired velocity that would lead to the goal at the end of the trajectory ($v_{des} = [0.5, 0, 0]$). Snapshots of the final trajectory can be seen on Figure 2. The mean contact error is quite low at 0.1 as the planned contact sequence is tracked well, as shown on Figure 3.

To highlight the importance of using contact-explicit methods for this kind of complex tasks, we tried our method in a contact-implicit setting ($w_{\mathcal{C}} = 0$), where only invalid contacts with the floor are penalized (with weight $w_{\mathcal{C}}^{\text{floor}} = 7.5$), while keeping all other parameters the same. In this case, our framework fails to find a solution that crosses the gap without relying on the floor, as shown in Figure 4. This result underlines the necessity of explicitly modeling contacts when solving challenging locomotion problems.

V. Conclusion and future work

We proposed a zero-order optimization approach for realizing given contact sequences with legged robots. Penalizing invalid contacts from the roll-outs in the TO cost enabled

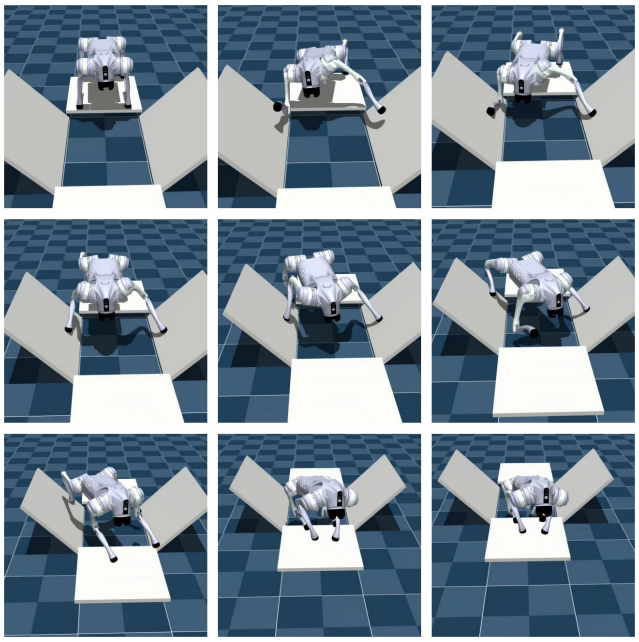

Fig. 2. Snapshots of the optimized trajectory for the clearing a gap task.

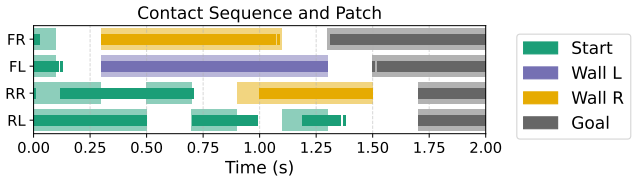

Fig. 3. Comparison of the planned and realized contact sequence for the cross gap task. Each color represents a different patch. Thicker bars in background correspond to the planned sequence, the thinner ones to the realized sequence. Note that no contact is made with the floor.

a quadruped robot to execute multiple gait patterns. On a more complex locomotion task in sparse environment, our method outperformed a traditional sampling-based contact-implicit approach. Our framework naturally supports parallel optimization of multiple contact plans, making it well-suited for large-scale data collection in contact-rich scenarios.

Even though this method has inherent limitations in very sparse environments and for long-horizon tasks, it has potential on medium-horizon problems with multiple contact switches. Building on this, future work will extend the approach to try planning basic manipulation skills on a humanoid robot, such as grasping, lifting, and placing objects. The long-term objective is to build an efficient data generation pipeline for those kinds of skills.

References

[1] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion: State of the art and future perspectives," *The International Journal of Robotics Research*, p. 02783649241312698, 2024.

[2] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," *arXiv preprint arXiv:2212.00541*, 2022.

[3] J. Alvarez-Padilla, J. Z. Zhang, S. Kwok, J. M. Dolan, and Z. Manchester, "Real-time whole-body control of legged robots with model-predictive path integral control," *arXiv preprint arXiv:2409.10469*, 2024.

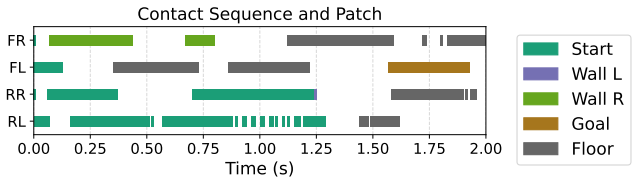

Fig. 4. Realized contact sequence for the cross gap task without specifying a contact plan. Only invalid contacts with the floor are penalized. The goal patch is not reached and multiple contacts are still made with the floor.

[4] H. Keshavarz, A. Ramirez-Serrano, and M. Khadiv, "Control of legged robots using model predictive optimized path integral," *arXiv preprint arXiv:2508.11917*, 2025.

[5] V. Kurtz and J. W. Burdick, "Generative predictive control: Flow matching policies for dynamic and difficult-to-demonstrate tasks," *arXiv preprint arXiv:2502.13406*, 2025.

[6] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing," 2024. [Online]. Available: https://arxiv.org/abs/2409.15610

[7] G. Kim, Y.-H. Lee, and H.-W. Park, "A learning framework for diverse legged robot locomotion using barrier-based style rewards," 2025. [Online]. Available: https://arxiv.org/abs/2409.15780

[8] H. Kim, H. Oh, J. Park, Y. Kim, D. Youm, M. Jung, M. Lee, and J. Hwangbo, "High-speed control and navigation for quadrupedal robots on complex and discrete terrain," 2025. [Online]. Available: https://arxiv.org/abs/2506.02835

[9] B. van Marum, A. Shrestha, H. Duan, P. Dugar, J. Dao, and A. Fern, "Revisiting reward design and evaluation for robust humanoid standing and walking," 2024. [Online]. Available: https://arxiv.org/abs/2404.19173

[10] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2017.

[11] S. Tonneau, D. Song, P. Fernbach, N. Mansard, M. Taix, and A. Del Prete, "Sl1m: Sparse l1-norm minimization for contact planning on uneven terrain," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6604–6610.

[12] H. Zhu, A. Meduri, and L. Righetti, "Efficient object manipulation planning with monte carlo tree search," in *2023 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2023.

[13] V. Dhédin, A. K. C. Ravi, A. Jordana, H. Zhu, A. Meduri, L. Righetti, B. Schölkopf, and M. Khadiv, "Diffusion-based learning of contact plans for agile locomotion," in *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2024, pp. 637–644.

[14] V. Dhédin, H. Zhao, and M. Khadiv, "Simultaneous contact sequence and patch planning for dynamic locomotion," 2025. [Online]. Available: https://arxiv.org/abs/2508.12928

[15] P.-T. Boer, D. Kroese, S. Mannor, and R. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, pp. 19–67, 02 2005.

[16] C. Pinneri, S. Sawant, S. Blaes, J. Achterhold, J. Stueckler, M. Rolinek, and G. Martius, "Sample-efficient cross-entropy method for real-time planning," 2020. [Online]. Available: https://arxiv.org/abs/2008.06389

[17] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017. [Online]. Available: https://doi.org/10.2514/1.G001921

[18] A. Jordana, J. Zhang, J. Amigo, and L. Righetti, "An introduction to zero-order optimization techniques for robotics," 2025. [Online]. Available: https://arxiv.org/abs/2506.22087

[19] M. Fornasier, T. Klock, and K. Riedl, "Consensus-based optimization methods converge globally," *SIAM Journal on Optimization*, vol. 34, no. 3, p. 2973–3004, Sep. 2024. [Online]. Available: http://dx.doi.org/10.1137/22M1527805

[20] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs *et al.*, "Mujoco playground," *arXiv preprint arXiv:2502.08844*, 2025.