This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Contact-Implicit Optimal Planning and Iterative Learning Control for Quadrupedal Robots

Pietro Gori, *Graduate Student Member, IEEE*, Vincenzo Degiacomo, Michele Pierallini, *Member, IEEE*, Franco Angelini, *Member, IEEE*, and Manolo Garabini, *Member, IEEE*

***Abstract*—In this article, we propose a novel approach for motion planning and control in quadruped robots that simultaneously optimizes the base trajectory and contact sequence along the longitudinal direction. The method relies on a simplified 2-D single rigid body model to compute an optimal trajectory, which is then mapped at the joint level. To compensate for model approximations and environmental uncertainties, we introduce an iterative learning control (ILC) scheme that progressively improves tracking performance across repetitions. Compared to existing approaches, our formulation unifies contact planning and trajectory optimization in a learning-while-doing framework, enhancing robustness to disturbances. We validate the method on the quadruped robot Solo-12, demonstrating adaptive gait generation and stable locomotion across different terrains and perturbations.**

***Index Terms*—Legged robots, nonlinear optimal control, numerical optimization, quadruped robots.**

I. Introduction

RESEARCH in mobile robotics has led to the development of complex and high-performance systems, among which quadrupedal robots stand out [1]. They consist of a floating base and four articulated legs, making their base pose independent of the surrounding environment [2]. This capability allows them to move over challenging terrains where wheeled robots fail [3], [4], enabling applications such as search and rescue [5], environmental monitoring [6], and space exploration [7]. These advantages, however, come at the price of highly nonlinear, impulsive, and hybrid dynamics [8], [9].

Received 2 September 2025; revised 25 November 2025; accepted 8 December 2025. This work was supported in part by the Italian Ministry of University and Research (MUR) - Fondo Italiano per la Scienza Applicata (FISA), through the OCCAM project under Grant FISA 2023-00324 CUP I53C25000700001 and the “FoReLab” (Future-oriented Research Lab) Project (Departments of Excellence), and in part by the European Union HORIZON-MSCA-2023-SE-01-01 - MSCA Staff Exchanges 2023 Program under Grant 101182891 (NEUTRAWEED). *(Pietro Gori and Vincenzo Degiacomo contributed equally to this work.)*
(Corresponding author: Pietro Gori.)

Pietro Gori, Franco Angelini, and Manolo Garabini are with the Centro di Ricerca “Enrico Piaggio,” Università di Pisa, 56126 Pisa, Italy, and also with the Dipartimento di Ingegneria dell’Informazione, Università di Pisa, 56126 Pisa, Italy (e-mail: pietro.gori@phd.unipi.it).

Vincenzo Degiacomo and Michele Pierallini are with the Centro di Ricerca “Enrico Piaggio,” Università di Pisa, 56126 Pisa, Italy.
This article has supplementary downloadable material available at https://doi.org/10.1109/TIE.2025.3645471, provided by the authors.

Digital Object Identifier 10.1109/TIE.2025.3645471

Legged robots can already execute dynamic motions [8], [10], but achieving energy-efficient, fast, and adaptive locomotion in real-world environments remains an open challenge [11]. Existing planning and control strategies [12], [13] often rely on accurate models or terrain assumptions, which limit their robustness. For instance, optimization-based planners can generate efficient motions on flat terrain but may fail on irregular or deformable surfaces due to unmodeled ground interactions. Learning-based approaches [14] can improve adaptability, yet they often require large amounts of training data and may generalize poorly when the robot encounters unexpected disturbances such as sudden pushes or foot slippage. In field conditions, these limitations translate into failure scenarios such as stumbling over hidden roots, losing stability when transitioning from grass to gravel, or being unable to maintain speed under external perturbations. This gap highlights the need for methods that remain tractable while being robust to model inaccuracies and environmental uncertainties. This article addresses this challenge by proposing a simplified yet effective motion planning and control framework that combines model-based trajectory generation with data-driven disturbance rejection, Fig. 1.

The main contributions of this article are as follows.

1) A tractable motion planning framework that simultaneously optimizes the base trajectory and contact sequence using a simplified 2-D single rigid body model and leverages the sigmoid function to continuously unify the contact phases.
2) The integration of iterative learning control (ILC) to progressively reduce the model–reality gap, improving tracking performance under disturbances in a *“learning-while-doing”* approach.
3) Experimental validation on the quadruped Solo-12, demonstrating adaptive gait generation and robust locomotion across diverse terrains and perturbations.

II. Related Works

Successful legged locomotion can be achieved by fixing the gait sequence or not, i.e., contact-implicit methods. Among the first class, in [15] the zero moment point approach [16] is used at the cost of limiting the potential base motions. Conversely,

© 2026 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
QUALITATIVE COMPARISON OF RELATED METHODS AND OUR APPROACH

Method	Model Fidelity	Contact Seq. Planning	Online Gait Adaptation	Terrain/Model Knowledge	Pre training	Real-World Learning
[12], [15], [17]	Simplified/full	Fixed	No	Moderate	No	No
[21], [22], [23]	Simplified/full	Implicit	Limited	High	No	Limited
[26]	Full	Optimized (discrete)	Limited	High	No	No
[30], [32], [31]	Full	Implicit/online	Yes	High	No	Limited
[33], [34], [35], [37]	Simplified/full	Learned/implicit	Yes	Low	Yes (extensive)	Yes (but costly)
[39], [40]	Simplified	Fixed	Limited	Moderate	Sometimes	Few trials
Ours	Simplified	Implicit	Yes	Low/robust	No	Few trials (ILC) (ILC) (ILC)

in [17] a simplified model is used, and the constraints on the foothold sequence are relaxed by fixing only the time instants at which each foot is in contact. Additionally, in [12], an optimal control problem solves the locomotion problem for a switching system. In [18], the full system dynamics is considered in a differential dynamic programming (DDP) framework.

On the other hand, contact-implicit methods allow more flexibility at the cost of the challenges introduced by contact planning [19], [20]. In [21], the problem is tackled by reshaping a relaxed version of the linear complementarity constraint (LCP) and then stiffening it progressively. In [22], [23], LCP is avoided using a soft contact model, which approximately solves the impulse theorem. In [24], [25], the duration of the contact phases is included as an optimization variable. The dynamic evolution of each phase is then known and always satisfies the LCP. Conversely, [26] optimizes the gait treating the contacts as an integer optimization variable. Next, the dynamics is optimized by formulating a mixed integer problem (MIP). These problems are numerically more complex than those with only continuous variables [27] and usually require a double-level optimization, i.e., gait and dynamics [28].

Literature also proposes contact-implicit methods that simultaneously plan and control quadruped robots' motion. Specifically, model predictive control (MPC) methods exactly predict the system state and contact forces. The efficiency of such methods relies entirely on the accuracy of the model and the speed with which it is simulated. Since the goal of the method is to calculate exactly the impulsive forces that generate the motion of the floating base, its effectiveness depends strongly on how it handles the contacts between feet and ground, and from an accurate knowledge of the ground shape, stiffness, and friction. Direct methods [29], [30] and DDP/iLQR methods [31], [32] have been applied to this topic. In [30], a contact-implicit MPC is proposed, which can readjust the assigned gait as needed, using a custom interior point method and a computationally efficient formulation of LCP. Kim et al. [32] presents a DDP method with a hard contact model, where the MPC framework is able to generate motions using an arbitrary ground contact sequence generated online. A customized iLQR method for hybrid systems (hybrid iLQR) is described in [31], where the HiLQR-MPC adapts the gait online to handle external disturbances, such as temporary motor deactivation. Despite the success of these methods, they assume the exact knowledge of the robot model and the terrain model, which is not always feasible in practice.

Data-driven methods, e.g., supervised, unsupervised, and reinforcement learning, can solve the planning modeling gaps. In [33], CoM and foot trajectories are planned for a biped with a spring loaded inverted pendulum (SLIP) model and fixed gait. This is then translated into the joint space and tracked with low-level PD. The loop is further closed on a neural network (NN) that takes as input the planned Cartesian trajectories and estimated trajectories and provides additional position references for the PDs. The training of the NN takes place offline with a reinforcement learning algorithm in MuJoCo [22] and requires several hours of training. Similarly, in [34], [35], [36] the policy is trained using trajectories derived from a model-based planner, but then, only the trained policies are used in the control loop. Specifically, [34] and [36] generate the trajectories based on simplified models, i.e., single rigid body (SRB) and SLIP, while [35] mimics the behavior of an MPC as references. In [37], instead, Xie et al. directly train an end-to-end policy on a simplified SRB model with good results in the sim-to-real transition. However, these methods may lack safety and/or stability guarantees [38] and require a large amount of data, and several hours of training time. Conversely, ILC methods can quickly fill the reality gap with only a few trials, at the cost of a task-specific result. In [39], a framework based on trajectory optimization and functional ILC [40] is proposed. However, the method is based on an SLIP model with a fixed pronking gait.

In this article, we propose a framework for planning and controlling legged systems. The contact-implicit planner uses a simplified model to efficiently compute feasible contact sequences that enable different gaits, such as pronking or bounding. The reality gap is handled by a PD+ILC controller that combines the stability of feedback control with the adaptability of a learned feedforward term. This improves tracking performance and environmental interaction without sacrificing robustness. Similar to [33], the ILC works in a *"learning-while-doing"* manner, learning directly from real data in just a few iterations and requiring no pretraining.

As summarized in Table I, our approach stands out from existing methods that depend on accurate terrain models or heavy offline training. It combines an efficient contact-implicit planner, able to produce diverse gaits, with an online ILC loop that bridges the reality gap in only a few trials, making the framework practical for real-world deployment.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

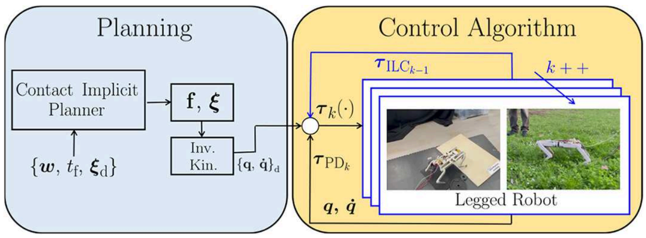


Fig. 1. Illustration of the proposed pipeline for legged robots. We plan the cartesian desired motion. We compute the joints' trajectory, which an iterative learning controller tracks. The framework exploits a *"learning-while-doing"* approach.

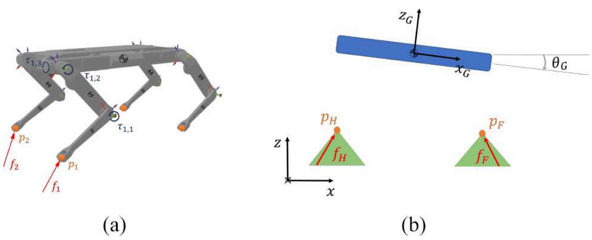


Fig. 2. Simplification of the rigid body model (RB) in the planar single rigid body model (2-D-SRB). (a) RB. (b) 2D-SRB.

III. Problem Definition

In this section, we start modeling a legged robot, and then, we describe the planning and control problems without a pre-determined contact sequence.

Any legged robot in Fig. 2(a) can be described through the generalized coordinates $\mathbf{q} = [\mathbf{q}_{\mathrm{b}}^{\top} \ \ \mathbf{q}_{\mathrm{l}}^{\top}]^{\top} \in \mathcal{Q}^{n+6} \subset \mathbb{R}^{n+6}$, where $\mathbf{q}_{\mathrm{b}} \in \mathcal{Q}^6$ includes the base's pose and $\mathbf{q}_{\mathrm{l}} \in \mathcal{Q}^n$ the leg joints. Recalling [9], the dynamic motion can be rewritten as follows:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{S}^{\top}\boldsymbol{\tau} + \mathbf{J}(\mathbf{q})^{\top}\mathbf{f} \tag{1}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{(6+n)\times(6+n)}$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^{(6+n)\times(6+n)}$ the Coriolis matrix, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{(6+n)}$ the gravity terms, $\mathbf{S} = \mathtt{diag}\{[\mathbf{0}_{n\times 6}, \mathbf{I}_{n\times n}]\}$ a selection matrix that only applies the torque $\boldsymbol{\tau} \in \mathcal{T} \subset \mathbb{R}^n$ to the n joint rows and $\mathbf{J}(\mathbf{q})$ the Jacobian matrix mapping forces $\mathbf{f} \in \mathcal{F} \subset \mathbb{R}^{3n_k}$ at the n_k legs end-effectors to $6+n$ dimensional generalized forces. The time variable is omitted and the paper takes the problem of a finite time horizon, i.e., $t \in [0, t_{\mathrm{f}}]$.

Assuming that the joint velocities produce negligible momentum and the inertia of the whole body is nevertheless comparable to that of the nominal joint positions [24], (1) can be simplified as in Fig. 2. Defining $\mathbf{p}_i \in \mathcal{P} \subset \mathbb{R}^3$ the coordinates of the ith $i \in [1, n_k]$ contact point, $\mathbf{I} \in \mathbb{R}^{3\times 3}$ the principal moments of inertia, $\boldsymbol{\omega} \in \mathbb{R}^3$ the system's angular velocity, and $\mathbf{r} \in \mathcal{R} \subset \mathbb{R}^3$ the center of mass (CoM), the Newton-Euler equations [41] of the single rigid body (SRB) model result in

$$m\ddot{\mathbf{r}} = m\mathbf{g} + \sum_{i=1}^{n_k} \mathbf{f}_i \tag{2a}$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} = \sum_{i=1}^{n_k} \mathbf{f}_i \times (\mathbf{r} - \mathbf{p}_i). \tag{2b}$$

Note that (2) neglects the joints' kinematics, which will be calculated later through inverse kinematics.

From (2), it is clear that the base motion is generated by the contact forces $\mathbf{f}_i$, which are subject to the *Complementarity Constraint* [42], i.e., $f_i^z p_i^z = 0, \ \forall i \in [1, n_k]$, where f_i^z and p_i^z are the vertical component and the height of the ith contact point, respectively. This means that $f_i^z \geq 0$ if and only if $p_i^z = 0$.

The models described in (1) and (2) can be related by the differential kinematics [41]

$$\dot{\boldsymbol{\xi}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{3}$$

where $\boldsymbol{\xi} = [\mathbf{q}_{\mathrm{b}}^{\top}, \mathbf{p_1}^{\top}, \ldots, \mathbf{p}_{n_k}{}^{\top}]^{\top}$ contains the base's pose and feet position.

We assume what follows for the system in (1), (2), and (3).

Assumption 1: The planning and control problem is said to be feasible if $\forall \boldsymbol{\xi}_{\mathrm{d}} \in \mathbb{R}^{6+3n_k}$, $\exists \boldsymbol{\tau}_{\mathrm{d}}(\cdot) : [0, \ t_{\mathrm{f}}] \to \mathcal{T}$ and $\exists \mathbf{f}_{\mathrm{d}}(\cdot) : [0, \ t_{\mathrm{f}}] \to \mathcal{F}$ such as $||\boldsymbol{\tau}_{\mathrm{d}}(t)|| \leq b_{\boldsymbol{\tau}_{\mathrm{d}}} < +\infty$ and $||\mathbf{f}_{\mathrm{d}}(t)|| \leq b_{f_{\mathrm{d}}} < +\infty$, and $\exists \mathbf{q}_{\mathrm{d}}(\cdot), \dot{\mathbf{q}}_{\mathrm{d}}(\cdot) : [0, \ t_{\mathrm{f}}] \to \mathcal{Q}^{n+6}$ such that (1), (2), and (3) are satisfied.

Assumption 1 ensures the existence of a realizable control action that can guide the system to a precise state in a finite time. Since the feasibility constraints will be modeled in the optimal control problem, this assumption is not strict.

The main goal is to create a pipeline for quadrupedal robot motion without a fixed contact sequence. To this end, the pipeline is composed of two phases: planning and control.

Goal 1: Given the initial pose $\boldsymbol{\xi}_0$, determine a sequence of $\mathbf{q}_{\mathrm{d}}(\cdot) \triangleq [\mathbf{q}_{\mathrm{bd}}^{\top} \ \mathbf{q}_{\mathrm{ld}}^{\top}]^{\top}$, $\dot{\mathbf{q}}_{\mathrm{d}}(\cdot) : [0, \ t_{\mathrm{f}}] \to \mathcal{Q}^{n+6}$, and $\mathbf{f}_{\mathrm{d}}(\cdot) : [0, \ t_{\mathrm{f}}] \to \mathcal{F}$ that bring the robot into the desired pose $\boldsymbol{\xi}_{\mathrm{f}}$.

Goal 2: Find the finite control action $\boldsymbol{\tau}(\cdot) : [0, \ t_{\mathrm{f}}] \to \mathcal{T}$ able to track the planned trajectory, i.e., $\mathbf{q}_{\mathrm{l}}(t) \to \mathbf{q}_{\mathrm{ld}}(t), \ \forall t \in [0, \ t_{\mathrm{f}}]$ while compensating for unmodeled disturbances and artifacts.

The first goal, i.e., G1, is related to the planning, while G2 focuses on the control problem.

IV. Proposed Framework

In this section, we propose a framework for achieving the goals defined in Section III. First, we derive a simplified robot model, focusing on the kinematics of the feet and contact handling. Next, we plan a feasible trajectory in Cartesian space using *direct multiple shooting* (DMS) [10], [43], [44], [45], which is mapped into joint space. Finally, we implement an ILC [46] algorithm to track the joint references while compensating for uncertainties and disturbances not included in the planning.

A. Planar Model

The goal of this section is to derive a simple yet not trivial model to plan the robot's motion. Considering a longitudinal motion and a symmetric base, a 2-D-SRB model can be adopted. This choice inevitably neglects lateral dynamics, roll and yaw rotations, and possible asymmetries between left and right legs. Contact forces are also restricted to the longitudinal plane, so terrain irregularities that cause out-of-plane effects are not explicitly captured. Nevertheless, for the gaits considered in this article (bounding and pronking), the dominant motion is longitudinal and symmetric, meaning that forward progression, vertical dynamics, and pitch regulation are the most relevant aspects. Hence, the 2-D approximation preserves the essential dynamics while keeping the model tractable for motion planning. Additionally, we assume that both front legs and hind legs move identically. This leads to having only four motion phases, i.e., *Full Flight, Front Contact, Hind Contact, Full Contact*, each characterized by different dynamics. It is worth mentioning that the planner can output different gaits even under this assumption, as it is shown in Section V.

Referring to Fig. 2(b), we define $\overline{\mathbf{r}}_G = [x_\mathrm{G},\ 0,\ \ z_\mathrm{G}]^\top$, the coordinates of the floating base, and $\overline{\theta}_G = [0,\ \theta_\mathrm{G},\ 0]$ its Euler angles, where only the pitch one is nonzero. We also define $\overline{\mathbf{p}}_i = [p_i^x,\ 0,\ p_i^z]^\top$, $i \in \{F,\ H\}$ the coordinates of each pair of feet (Front or Hind). The overall continuous dynamics is as follows:

$$\ddot{x}_\mathrm{G} = \frac{1}{m} \sum_{i=H,F} \sigma(-p_i^z + h_t(p_i^x)) f_i^x \tag{4a}$$

$$\ddot{z}_\mathrm{G} = \frac{1}{m} \sum_{i=H,F} \sigma(-p_i^z + h_t(p_i^x)) f_i^z - g \tag{4b}$$

$$\ddot{\boldsymbol{\theta}}_\mathrm{G} = \frac{1}{I_z} \sum_{i=H,F} [\sigma(-p_i^z + h_t(p_i^x))\overline{\mathbf{f}}_\mathbf{i}] \times (\overline{\mathbf{p}}_i - \overline{\mathbf{r}}_\mathrm{G}) \tag{4c}$$

$$\dot{p_i}^x = \sigma(p_i^z - h_t(p_i^x))(v_i^x + \dot{x}_\mathrm{G} + \dot{\theta}_\mathrm{G}(p_i^z - z_\mathrm{G})) \tag{4d}$$

$$\dot{p_i}^z = \sigma(p_i^z - h_t(p_i^x))(v_i^z + \dot{z}_\mathrm{G} - \dot{\theta}_\mathrm{G}(p_i^x - x_\mathrm{G})) \tag{4e}$$

where $\overline{\mathbf{f}}_\mathbf{i} = [f_i^x, 0,\ f_i^z]^\top$ and $[v_i^x, 0,\ v_i^z]$, $\forall i \in \{F,\ H\}$ are the contact forces and the relative velocities of the feet related to the base, respectively. They serve as inputs to the system. The $h_t(x_i)$ contains the ground height. The key aspect of this model is the sigmoid function $\sigma(z) = (1 + e^{-z\sigma_\mathrm{sl}})^{-1}$, where σ_sl is the slope. This function enables to unify the four distinct dynamics, e.g., flight or in contact, into a single continuous one. More details are reported in the Section IV-B. The no-slip constraint is a key condition to impose, i.e.,

$$-\mu f_i^z \le f_i^x \le \mu f_i^z, \quad f_i^z \ge 0 \tag{5}$$

where $\mu \in [0, 1]$ is the static friction coefficient of the terrain, f_i^x and f_i^z are respectively the tangential and normal components of the ith contact force. Note that (5) describes a friction planar constraint [10], Fig. 2(b).

B. Feet Kinematics

Each foot is considered massless, it is modeled kinematically, and it is constrained to remain in the operative space of the corresponding leg. Considering the ith pair of feet, during the contact phase, due to the no-slipping constraint, it holds that $\dot{p}_i^x = 0$, and $\dot{p}_i^z = 0$ with $i \in \{F,\ H\}$. Conversely, during the flight phase, any feet move as follows:

$$\begin{cases} \dot{p}_i^x = v_i^x + \dot{x}_\mathrm{G} + \dot{\theta}_\mathrm{G}(p_i^z - z_\mathrm{G}) \\ \dot{p}_i^z = v_i^z + \dot{z}_\mathrm{G} - \dot{\theta}_\mathrm{G}(p_i^x - x_\mathrm{G}) \end{cases} \tag{6}$$

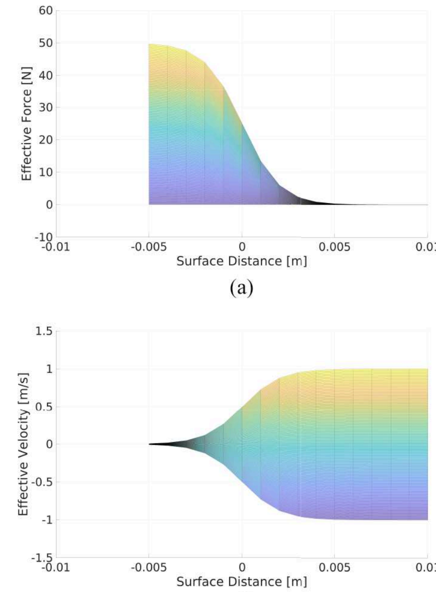
for $i \in \{F,\ H\}$. Note that, (6) allows effects of the rotational dynamics of the base on the robot's feet.

To ensure the foot remains inside the leg's workspace, we introduce geometric constraints by constructing a boundary box. We define $l_\mathrm{box} = x_\mathrm{ub} - x_\mathrm{lb}$, $h_\mathrm{box} = z_\mathrm{ub} - z_\mathrm{lb}$, and $d_\mathrm{min} = a_i^z - z_\mathrm{ub}$ as free parameters of the optimization problem, while $a_i = [a_i^x,\ a_i^z]^\top$ and $p_i = [p_i^x,\ p_i^z]^\top$ represent the hip and the foot position of the ith leg. Therefore, we set a linear constraint on each foot coordinate, i.e.,

$$\begin{bmatrix} x_\mathrm{lb} \\ z_\mathrm{lb} \end{bmatrix} \le \begin{bmatrix} p_i^x - a_i^x \\ p_i^z - a_i^z \end{bmatrix} \le \begin{bmatrix} x_\mathrm{ub} \\ z_\mathrm{ub} \end{bmatrix}. \tag{7}$$

The feet bounding box (7) is convex, thus it can be easily handled by numerical solvers, and it ensures that the leg does not fully extend, preventing singularities.

In the proposed planner, only the presence or absence of the contact forces depends on the system's state. At the same time, the solver chooses the value of the contact forces f_i^z freely. Recalling (4) and assuming $h_t(p_i^x) = 0$, it holds $f_{\mathrm{effective},i}^z = \sigma(-p_i^z) f_i^z$, see Fig. 3(a). To maintain a pure kinematic feet model and limit the maximum ground penetration, feet velocity is subject to the sigmoid effect following (4d) (4e), Fig. 3(b). The sigmoid function activates when moving away from the contact surface. Additionally, the choice of the control weights w steers the areas selected by the solver, avoiding those where both inputs are active, and preferring those where the controls have full effect on the system. This means areas where $\sigma(\cdot) \to 1$, i.e., maximum ground compression for contact force or in the air for foot velocities.



(a)

(b)

Fig. 3. Effect of the proposed contact model. The filled surface represents the solution space available to the solver. Acting on the cost function causes the solver to act in regions where forces and velocities are maximized, i.e., $\sigma(\cdot) \to 1$. (a) Sigmoid effects w.r.t. the contact forces. (b) Sigmoid effects w.r.t. the foot velocities.

C. Numerical Optimization

DMS reformulates the finite time optimal control problem into a finite dimension nonlinear programming (NLP) one. Given a finite time horizon t_f, we divide it in a time grid of N equally spaced temporal instants $t_i = i\Delta t\ : \Delta t = t_\mathrm{f}/N, i \in$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

$[0, N]$. Let us define the control input $\mathbf{u}(\cdot) \in \mathbb{R}^{m_u}$ such as $\mathbf{u}(t) = \mathbf{u}_i : t \in [t_i, t_{i+1}], \forall i \in [0,\ N-1]$. Moreover, we define the system state $\mathbf{x} \in \mathbb{R}^{m_x}$ and the state transition function $f(\cdot,\cdot) : \mathbb{R}^{m_x} \times \mathbb{R}^{m_u} \to \mathbb{R}^{m_x}$ [45]. Using such controls, the dynamics is simultaneously solved N times, one for each interval $[t_i, t_{i+1}]$, starting from an artificial state $\mathbf{s}_i \in \mathbb{R}^{m_x}$, i.e.,

$$\begin{aligned} \dot{\mathbf{x}}_i &= f(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}_i \ : t \in [t_i,\ t_{i+1}],\ \forall i \in [0,\ N] \\ \mathbf{x}(t_i) &= \mathbf{s}_i \end{aligned} \tag{8}$$

where the continuity between each interval is imposed with an equality constraint, i.e., $\mathbf{s}_{i+1} = \mathbf{x}(t_{i+1}, \mathbf{s}_i, \mathbf{u}_i),\ \forall i \in [0, N]$.

DMS solves the following optimal control problem leading to a feasible piecewise trajectory, i.e.,

$$\underset{\mathbf{s},\mathbf{u}}{\text{minimize}} \sum_{i=0}^{N-1} L(\mathbf{s}_i, \mathbf{u}_i) + E(\mathbf{x}_N) \tag{9a}$$

$$\text{subject to} \mathbf{x}_0 - \mathbf{s}_0 = 0 \tag{9b}$$

$$\mathbf{s}_{i+1} - \mathbf{x}(t_{i+1}, \mathbf{s}_i, \mathbf{u}_i) = 0\ \forall i \in [0, N] \tag{9c}$$

$$\mathbf{g}(\mathbf{x}_i, \mathbf{u}_i) \geq 0\ \forall i \in [0, N] \tag{9d}$$

$$\mathbf{h}(\mathbf{s}_N) = 0. \tag{9e}$$

We apply the DMS approach to our planning problem in Section III. Defining the state vector $\mathbf{x} = [\xi^\top,\ \dot{x}_{\text{G}},\ \dot{z}_{\text{G}},\ \dot{\theta}_{\text{G}}]^\top$ and the control vector $\mathbf{u} = [f_F^x, f_F^z, f_H^x, f_H^z, v_F^x, v_F^z, v_H^x, v_H^z]^\top$, the model (4) can be used in (8) as a state update function. Equations (5) and (7) are then used as inequality constraint (9d), while the terminal constraint (9e) is the L_2-norm of the distance between the final robot state $\mathbf{x}(t_{\text{f}})$ and the final desired state $\mathbf{x}_{\text{d}}$. The cost function (9a) penalizes the L_2-norm of the input $\mathbf{u}$ and the state $\mathbf{x}$. Hence, the optimization result will be the trajectory of the robot CoM, feet, and the contact force sequence. Note that the weights of (9a) can be selected to enable different gaits. Different from other works [24], [47], we generate the contact sequence as part of the optimization process, without relying on a predefined one. The initial guess defines the initial state, but the optimization solver can deviate from it to find a more general solution. Unlike other approaches that constrain state variables to the initial guess to reduce the problem size, we allow for a larger optimization problem to achieve feasibility and explore solutions beyond the initial guess.

D. Inverse Kinematics

To perform the optimal motion planned in Section IV-C, we transform the CoM and feet trajectories into joint ones. In the planar case, the ith leg is a 2-DoF robot. Recalling (7), the hip position a_i and foot position p_i are used to compute the joint angles via an Inverse Kinematics algorithm, see, e.g., [48], $\mathbf{q}_{\text{ld}_i} = \text{IK}_i(\mathbf{d}_{\text{ap}}^{\text{A}})$, where $\text{IK}_i(\cdot) : \mathbb{R}^2 \to \mathbb{R}^2$ is the kinematic inversion map of the ith leg, and $\mathbf{d}_{ap}^A \in \mathbb{R}^2$ is the hip–foot distance in the hip frame. The velocities $\dot{\mathbf{q}}_{\text{d}}$ can then be derived either numerically or analytically while satisfying G1. Thus, for each joint, we obtain a position and velocity reference trajectory.

Recalling (1) and (4), the optimal contact forces $\overline{\mathbf{f}}(\cdot)$ are also used to limit the torques $\boldsymbol{\tau}$ required by the controller defined in Section IV-E. During contact phases, the leg dynamics can be expressed as follows:

$$\boldsymbol{\tau} \simeq \mathbf{J}_{\text{l}}^\top(\mathbf{q})\mathbf{f} \tag{10}$$

where $\mathbf{J}_{\text{l}}(\mathbf{q})$ is the leg Jacobian. Since the contact forces are explicitly bounded in the optimization problem, i.e.,

$$\mathbf{f}_{\text{lb}} \leq \mathbf{f} \leq \mathbf{f}_{\text{ub}} \tag{11}$$

and the elements of $\mathbf{J}_{\text{l}}(\mathbf{q})$ remain bounded by the leg geometry (with $|J_{\text{l}}^{ij}| < l_1 + l_2$ for link lengths $l_1, l_2 \in \mathbb{R}$), it follows that the resulting joint torques are also bounded. By appropriately selecting the force limits $\mathbf{f}_{\text{lb}}$, $\mathbf{f}_{\text{ub}}$ and the kinematic parameters l_{box}, h_{box}, and d_{min}, the feasibility of the planned trajectories in terms of torque requirements is guaranteed.

E. Iterative Learning Controller

The joint reference trajectory is tracked by a low-level controller. This is composed of two terms. One is a PD controller, i.e.,

$$\boldsymbol{\tau}_{\text{PD}} = \mathbf{K}_{\text{P}}(\mathbf{q}_{\text{ld}} - \mathbf{q}_{\text{l}}) + \mathbf{K}_{\text{d}}(\dot{\mathbf{q}}_{\text{ld}} - \dot{\mathbf{q}}_{\text{l}}) \tag{12}$$

where $\mathbf{K}_{\text{P}} = K_{\text{p}}\mathbf{I}_{n\times n}$ and $\mathbf{K}_{\text{d}} = K_{\text{d}}\mathbf{I}_{n\times n} \in \mathbb{R}^{n\times n}$ are the proportional and derivative gains, respectively.

The second term is included to increase robustness. Indeed, when using simple models and controllers such as (4) and (12), respectively; discrepancies between the real robot motion and the planned one are likely to occur. This depends on neglecting leg dynamics, actuator artifacts, different terrains, poor contact models, external disturbances, etc. ILC strategy tackles this challenge [46]. The idea behind this method is to repeat the same task several times and learn from the error made in the previous iteration. Iteration after iteration, this knowledge is translated into a feedforward control term to improve trajectory tracking performance. In practice, the planned task is repeated identically N_k-times, and at the kth iteration the new feedforward term can be computed as follows:

$$\boldsymbol{\tau}_{\text{ff}_k} = \boldsymbol{\tau}_{\text{ff}_{k-1}} + \overline{\mathbf{K}}_{\text{p}}(\mathbf{q}_{\text{ld}} - \mathbf{q}_{\text{l},k-1}) + \overline{\mathbf{K}}_{\text{d}}(\dot{\mathbf{q}}_{\text{ld}} - \dot{\mathbf{q}}_{\text{l},k-1}) \tag{13}$$

where $\boldsymbol{\tau}_{k-1}^{\text{ff}} \in \mathbb{R}^n$ is the old feedforward term and $\overline{\mathbf{K}}_{\text{p}} = \overline{K}_{\text{p}}\mathbf{I}_{n\times n} \in \mathbb{R}^{n\times n}$ and $\overline{\mathbf{K}}_{\text{d}} = \overline{K}_{\text{d}}\mathbf{I}_{n\times n} \in \mathbb{R}^{n\times n}$ are the gains of the ILC updating law, weighing the error at the previous iteration. The overall control law at the kth iteration is as follows:

$$\boldsymbol{\tau}_k = \boldsymbol{\tau}_{\text{PD}_k} + \boldsymbol{\tau}_{\text{ff}_k}. \tag{14}$$

To theoretically analyze the convergence of the controller (14), we recall the desired state $\mathbf{q}_{\text{ld}}, \dot{\mathbf{q}}_{\text{ld}} \in \mathbb{R}^n$. Under this control law, the input remains bounded with respect to time and the iteration domain, and it converges to the desired control input while compensating for model uncertainties, i.e.,

$$||\mathbf{d}(\mathbf{q}_{\text{l}}, \dot{\mathbf{q}}_{\text{l}}, t)|| \leq b_{\text{d}}\left(||\mathbf{q}_{\text{l}}|| + ||\dot{\mathbf{q}}_{\text{l}}||\right) \tag{15}$$

where $\mathbf{d}(\cdot) \in \mathbb{R}^n$ denotes state-repetitive disturbances [49] and $b_{\text{d}} \in \mathbb{R}$ is finite.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

The boundedness of (14) can be shown by induction. At $k = 0$, $\boldsymbol{\tau}_0 = \boldsymbol{\tau}_{\mathrm{PD}_0}$ is bounded by construction, i.e., $|\boldsymbol{\tau}_0| < b_{\tau_0}$ finite. At $k = 1$, $\boldsymbol{\tau}_1$ is also bounded since both the PD component and the feedforward term depend on bounded signals. Assuming $|\boldsymbol{\tau}_k| < b_{\tau_k}$, one has for $k+1$ that both the feedback and feedforward components remain bounded, implying $|\boldsymbol{\tau}_{k+1}| < b_{\tau_{k+1}}$.

Convergence is analyzed by recalling the system state $\mathbf{x} \triangleq \left[\mathbf{q}_{\mathrm{l}}^\top, \dot{\mathbf{q}}_{\mathrm{l}}^\top\right]^\top \in \mathbb{R}^{2n}$ and its affine state-space form (8). Assuming Lipschitz continuity of (1) and defining $\delta(\cdot)_k \triangleq (\cdot)_{k+1} - (\cdot)_k$, one obtains

$$||\delta\mathbf{x}_k|| \leq \int_0^t (b_{\mathrm{f}} + b_{\mathrm{d}})\, ||\delta\mathbf{x}_k(z)|| + b_{\mathrm{g}}\, ||\delta\boldsymbol{\tau}_k(z)||\, \mathrm{d}z \tag{16}$$

where b_{f} and b_{g} are the Lipschitz constants of $\mathbf{f}(\cdot)$ and $g(\cdot)$, while b_{d} accounts for disturbances.

Applying Gronwall's Lemma to (16) yields

$$||\delta\mathbf{x}_k|| \leq \int_0^t \left(b_{\mathrm{g}}\, ||\delta\boldsymbol{\tau}_k(z)||\right) \mathrm{e}^{(b_{\mathrm{f}}+b_{\mathrm{d}})(t-z)}\mathrm{d}z. \tag{17}$$

Considering two consecutive iterations, the following inequality holds:

$$||\delta\boldsymbol{\tau}_{k+1}|| = \left|\left|\delta\boldsymbol{\tau}_k - \overline{K}\left(\delta\mathbf{q}_{\mathrm{l},k} + \delta\dot{\mathbf{q}}_{\mathrm{l},k}\right)\right|\right| \geq \left|\,|\delta\boldsymbol{\tau}_k| - |\overline{K}\delta\mathbf{x}_k|\,\right| \tag{18}$$

with $\overline{K} = \max|\overline{\mathbf{K}}_{\mathrm{p}}|, |\overline{\mathbf{K}}_{\mathrm{d}}|$. Substituting (17) into (18) and evaluating the λ-norm gives

$$|\delta\boldsymbol{\tau}_{k+1}|\lambda \leq (\pm 1 \mp \epsilon)\, |\delta\boldsymbol{\tau}_k|\lambda \tag{19}$$

where $\epsilon > 0$ for sufficiently large λ. This shows a contraction property, i.e., $(|\delta\boldsymbol{\tau}_{k+1}|\lambda / |\delta\boldsymbol{\tau}_k|\lambda) \in (-1, 0]$, proving the convergence of the ILC law.

V. Planner Validation

In this section, we validate the performance of the proposed planner by demonstrating its ability to generate a trajectory that moves the robot's floating base by 0.2 m in the X direction, in the world frame. This trajectory can be executed periodically, enhancing the robot's performance by leveraging the ILC capabilities. By adjusting the cost function weights, the planner can produce distinct behaviors for the floating base and feet trajectories, effectively enabling different gaits. A direct outcome of Sections IV-A and IV-B is that the framework can generate two primary gaits: *Pronking* and *Bounding*.

The implementation is carried out in `Python` using `CasADi` [50] to formulate the NLP problem described in Section IV-C. The optimization problem is solved using `Ipopt` [51] on a laptop equipped with an Intel Core i7-13650HX processor and 32 GB of RAM.

The platform chosen to validate the method is the 12 degrees of freedom (DoF) Solo-12 a lightweight quadrupedal robot. It weights about 2.5 kg. It has a body module and four identical 3 DoF legs. The robot's base is 45×30 cm and its maximum height is 34 cm. Recalling Section IV-D, every joint has a PD controller that operates at 10 kHz. The joint torque range is $[-2.7, 2.7]$ Nm and the robot is only equipped with proprioceptive sensors, such as joint encoders. The NLP problem is formulated and solved with $t_{\mathrm{f}} = 1.0$ s, $N = 100$ and a single *Forward Euler* integration method.

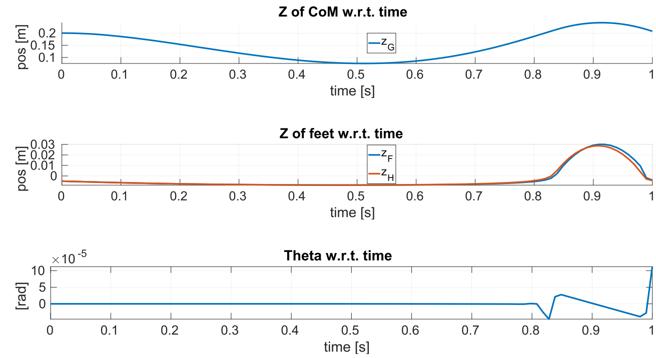


Fig. 4. Pronking: CoM, feet, and θ_G cartesian trajectories.

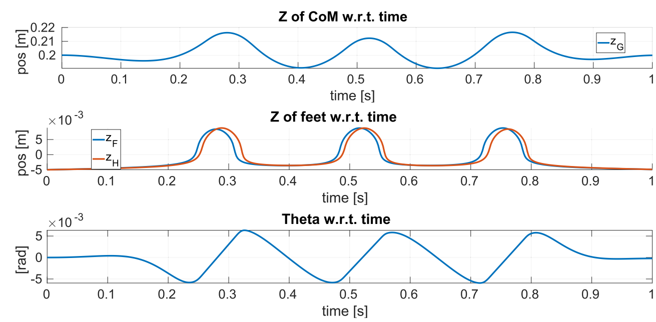


Fig. 5. Bounding: CoM, feet, and θ_G cartesian trajectories.

A. Pronking

We set an elevated attitude cost term $w_\theta = 1e6$ in the optimal planning problem (9). This leads the system to perform the displacement with a single jump, i.e., Pronking, as shown in Fig. 4. Moreover, the contact is relaxed with $\sigma_{\mathrm{sl}} = 500$, and the NLP problem is solved in 2.3 s.

B. Bounding

In this case, the system is required to move without changing the height of the base. This is done by relaxing the cost term on the attitude, $w_\theta = 1e2$, and increasing the term of the base vertical distance $w_{z_0} = 1e6$, from the reference height, i.e., $z_0 = 0.2$ m. The problem is solved in 8.4 s with $\sigma_{\mathrm{sl}} = 1e3$. This results in a shift in the front and hind feet motion, i.e., a Bounding gait as shown in Fig. 5.

VI. Controller Validation

In this section, we validate the effectiveness and robustness of the ILC with eight experiments both in structured and unstructured environments with varying terrain, disturbances, obstacles, etc. Moreover, we compare the performance of our

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

controller and a state-of-the-art (SoA) method, namely, OCS2 [12], [13]. Each of the planned trajectories in Section V is concatenated L-times and the learning process is performed online by exploiting their periodicity, i.e., *learning-while-doing*. In some experiments, an external disturbance, such as a small step or weight, is introduced during the learning process. The different types of disturbances in the learning process enable us to demonstrate how much a simple iterative learning strategy handles disturbances with few data. To evaluate the accuracy of joint position tracking during the kth iteration, we report the root mean square error (RMSE) of the joint position tracking across all joints, namely, (RMSE_k). This metric provides a concise measure of the overall tracking performance and allows for direct comparison between different experimental conditions. The method's first iteration always corresponds to the one performed with the PD controller only. Then the ILC approach updates the control law with a feed-forward term (12). Every experiment uses the same controller gains, which were selected through empirical testing and tuning procedures [52]: $K_\mathrm{p} = 30.0$, $K_\mathrm{d} = 0.05$, $\overline{K}_\mathrm{p} = 0.5$, and $\overline{K}_\mathrm{d} = 0.075$. The ILC gains selection proportionally affects the convergence velocity. Fig. 6 shows representative snapshots from the experiments. Additional details are provided in the attached video extension.

A. SoA Comparison

Here, we compare the performance of the proposed method with the SoA methods in the literature. We focus on the comparison with the methods that are based on the OCS2 toolbox. The OCS2 [12], [13] is a C++ toolbox tailored for optimal control for switched systems. Our method is based on a contact-implicit continuous model that exploits sigmoid activation functions to activate and deactivate feet contact, Sections IV-A and IV-B. The OCS2 toolbox, on the other hand, is based on switched systems, where the contact activation and deactivation is done by switching between different dynamics.

To evaluate the effectiveness of our approach, we compare it against the OCS2 planning method on a flat terrain scenario. Specifically, we analyze the trajectories generated by our method (Section V-B) alongside the one from the OCS2 toolbox, under the same settings. The planned tasks last for $1.0\,\mathrm{s}$, with the robot moving forward $0.2\,\mathrm{m}$. For a fair comparison, the trajectories produced by both planners are executed using the proposed controller, described in Section IV-E.

To compare the performance of our method with OCS2, we performed two simulations with 15 ILC iterations each. Fig. 7 shows the comparison outcome in terms of the joint positions RMSE. As observed, the RMSE of the joint positions for our method is slightly lower than that of the OCS2 planning. However, the difference is not significant, indicating that both methods yield comparable performance in terms of joint position accuracy.

These results demonstrate that our proposed method generates optimal trajectories that are competitive with state-of-the-art approaches. Furthermore, the comparison highlights the effectiveness and robustness of the ILC-based controller, which successfully executes trajectories planned by both our method and the OCS2 toolbox. This indicates that the controller is capable of handling diverse planning strategies while maintaining high performance.

B. Structured Environment

Both planned motions were tested on flat terrain similar to that used in the planning phase. Each trial consisted of 20 iterations with identical initial conditions and feedback gains. The first iteration used only the PD controller, while subsequent ones included ILC, which progressively compensated for modeling errors and improved trajectory tracking. The resulting RMSE values for both tasks are shown in Fig. 8.

C. Structured Environment With Disturbances

1) Extra Weight Ongoing: Starting from the *Bounding* task (Section V-B), once satisfactory performance was reached around the 10th iteration, an additional payload equal to 25% of the robot mass (0.5 kg) was placed on its floating base. The effect is visible in the RMSE in Fig. 8. The experiment was stopped at the 17th iteration to prevent overheating and was conducted only for the *Bounding* gait, as *Pronking* showed excessive base oscillations.

2) Multiterrains: To assess controller robustness, the robot was tested on three surfaces with different hardness and friction: hard rubber ($\mu = 0.6$), medium-hard sponge ($\mu = 0.75$), and laminated wood ($\mu = 0.5$), with small steps (7.5–15 mm) between them. Both gaits were tested, and RMSE trends are shown in Fig. 8.

D. Unstructured Environment

1) Grassland: Both planned gaits were tested on uneven terrain with vegetation and roots, none of which were included in the planning phase. As shown in Fig. 8, learning began from scratch, allowing ILC to compensate for modeling errors and terrain irregularities.

2) Grassland With Obstacle: The robot performed the *Pronking* task on grass while overcoming a 30 mm-high steel manhole cover. The corresponding RMSE and experimental results are reported in Figs. 6 and 8, respectively.

VII. Discussions and Limitations

1) Discussions: Section V demonstrates that the proposed planning method effectively generates periodic trajectories optimized for integration with the ILC framework. By adjusting the cost function, the planner produces distinct gaits and phase timings for the same displacement. Differences in foot height between Figs. 4 and 5 arise from the chosen σ_sl value, which only constrains joint torques and does not affect control performance. Simulation results show that $\sigma_\mathrm{sl} > 500$ ensures feasible solutions, while larger values mainly increase NLP computation time. The maximum achievable forward distance in 1 s is 0.45 m, beyond which high landing velocities cause unwanted bounces.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

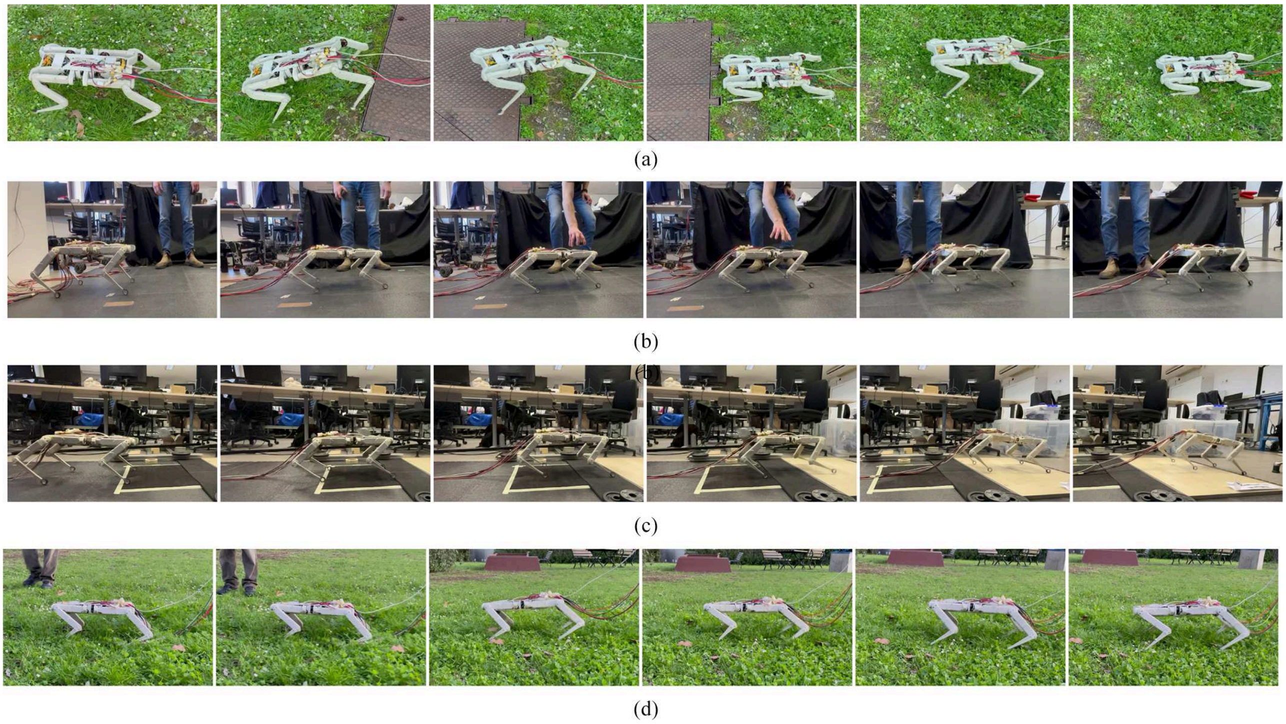


Fig. 6. Photo-sequence of four out of the eight performed experiments. Each test presents a different gait, a different terrain, or a different unmodeled external disturbance. Each picture represents a different ILC iteration in increasing order from left to right. (a) Pronking on grass with obstacle. (b) Bounding with added weight on iteration 10. (c) Bounding on terrain composed of multiple materials with different physical properties. (d) Bounding on grass.

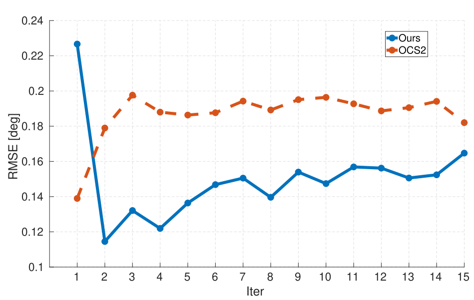


Fig. 7. RMSE of joint positions over 15 ILC iterations for trajectories planned by the OCS2 toolbox and our proposed method. Both planners are evaluated under identical conditions using the same controller.

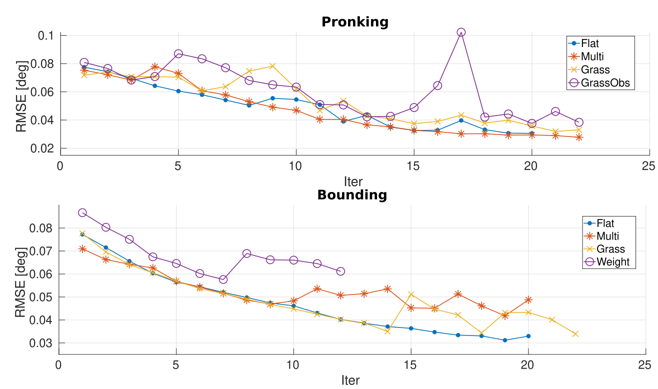


Fig. 8. ILC performance: RMSE over iterations for pronking and bounding gaits. Each plot shows the joint position tracking error across successive learning iterations, highlighting the improvement in tracking accuracy and the robustness of the proposed controller on different terrains.

The experiments described in Section VI prove the ILC's suitability for periodic trajectories in quadrupedal robots. We conducted independent experiments under different environmental conditions (flat terrain, multisurface soils, and grass), each evaluated separately for the Bounding and Pronking tasks. Moreover, we performed an additional experiment on flat terrain with an added weight of 0.5 kg to the front of the robot's base to assess the method's robustness to external disturbances. As shown in Fig. 8 (*Flat*), in the case of flat terrain, the robot learns progressively w.r.t. the iteration domain, making the tracking error at the joints decrease exponentially. This is more evident in the *Bounding* task since, initially, the robot fails to perform the displacement, succeeding from the $k = 4$ onward. On multisurface soils (Fig. 8, *Multi*), the *Pronking* task suffers only from crossing over the sponge, $k = 4, 5$, still recovering in a couple of cycles and overcoming the other soils without problems. The *Bounding* task, on the other hand, is more sensitive to changes in friction and hardness as it transitions from one surface to another, $k = 11,\ 14,\ 17,\ 20$. The method manages to compensate for such disturbances as early as the first iteration by resuming the initial learning trend. The same behavior occurs when a weight of 0.5 kg is added to the front of the base at $k = 10$ (Fig. 8 top, *Weight*). In this case, the task was terminated at $k = 17$ due to overheating of the actuators, which are capable of delivering a maximum torque of 2.5 N·m. In the case of an unstructured environment (Fig. 8, *Grass*), the robot starts moving only at $k = 15$ of the *Bounding* task. In subsequent iterations, the controller learns

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

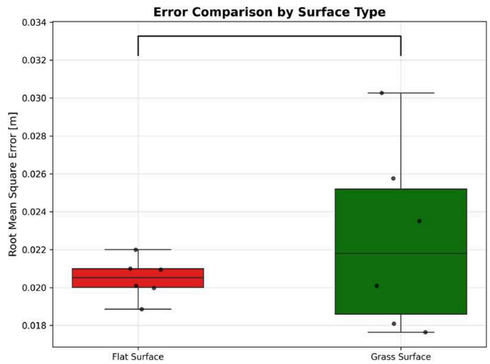


Fig. 9. Comparison between planned and executed base z-positions for flat terrain and grass experiments using the pronking gait. Each dot represents an iteration of the planned trajectory. The plot shows that while median tracking errors are similar between flat and grass surfaces, the variability of error is much greater on grass, reflecting the challenges posed by unstructured terrain.

the necessary torques to reject environmental disturbances such as roots sticking in the robot's feet. The same happens for the other planning, where the robot starts moving as early as the first iteration and improves tracking at joints as the execution numbers increase. The controller was also able to manage the transition from the lawn to a raised manhole (Fig. 8 top, *GrassObs*), $k = 4, 5$, and again to the meadow, $k = 16, 17$. In the latter, in particular, the transition from a surface with high hardness and friction to one with opposite characteristics generated a significant tracking error that was handled by the controller in two iterations, resuming the learning trend. It is noteworthy that, according to (14), in the first iteration of each experiment, the ILC contribution is zero, thus only the PD feedback control is active. To enhance the evaluation of the controller's performance, we include additional metrics comparing the planned and executed base trajectories. Specifically, we analyze the vertical (z) position of the base, which is less influenced by unmodeled external disturbances than the other components. Fig. 9 illustrates the error between the planned (Fig. 4) and executed base z-positions for both flat terrain and grass experiments using the pronking gait. The results indicate that, even though only the leg joints are controlled, the base trajectory closely follows the planned path, demonstrating the controller's effectiveness in maintaining desired body posture.

2) Limitations and Future Works: While the proposed ILC scheme effectively compensates for model inaccuracies through real-world adaptation, it is inherently task-specific and does not generalize across different motion patterns or environments. This represents a current limitation of our approach, as the learned corrections are tailored to the specific planned trajectory. To further enhance the ILC generalization capabilities, future work can leverage time-varying or control-oriented formulations [53], [54], or integrate ILC with deep reinforcement learning techniques [55] to enable adaptation across tasks and terrains. Extending the proposed framework to more complex gaits mainly requires updating the robot model. The current 2-D single rigid body (SRB) formulation, limited to the longitudinal plane, ensures clarity and efficiency but only supports symmetric gaits. To generate asymmetric patterns such as walking or trotting, a 3-D SRB model with full spatial dynamics and 3-D foot kinematics is needed, expanding dynamic and contact constraints to include lateral motion and independent leg behavior. Although this increases problem dimensionality, the overall framework remains unchanged: once the center of mass and foot trajectories are planned, the kinematic inversion and PD+ILC controller operate identically. Relaxing symmetry constraints within the same contact-implicit formulation would then allow the synthesis of general, nonsymmetric gaits.

VIII. Conclusion

In this article, we proposed a new pipeline to plan and control a quadruped robot. The planning phase simultaneously derives the base trajectory and contact sequence, which are then mapped at the joint level. Then, ILC is used to close the gap between the approximated model and the real system. Finally, we tested the effectiveness of the proposed method on the quadruped robot Solo-12 for forward movement with varying gaits, disturbances, and terrains. The proposed method is not intended for real-time applications, but rather as a tool to generate high-quality trajectories that can be executed in real-time.

References

[1] J. Zhang, D. Zhang, and Z. Guo, "Research on quadruped robots: Review and prospect," in *Proc. 5th Int. Conf. Inf. Technol. Computer Commun.*, 2023, pp. 38–45.
[2] H. Chai et al., "A survey of the development of quadruped robots: Joint configuration, dynamic locomotion control method and mobile manipulation approach," *Biomimetic Intell. Robot.*, vol. 2, no. 1, 2022, Art. no. 100029.
[3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, 2020, Art. no. eabc5986
[4] G. Valsecchi, R. Grandia, and M. Hutter, "Quadrupedal locomotion on uneven terrain with sensorized feet," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1548–1555, Feb. 2020.
[5] C. Cruz Ulloa, G. Prieto Sánchez, A. Barrientos, and J. Del Cerro, "Autonomous thermal vision robotic system for victims recognition in search and rescue missions," *Sensors*, vol. 21, no. 21, 2021.
[6] F. Angelini et al., "Robotic monitoring of habitats: The natural intelligence approach," *IEEE Access*, vol. 11, pp. 72575–72591, 2023.
[7] H. Kolvenbach, M. Breitenstein, C. Gehring, and M. Hutter, "Scalability analysis of legged robots for space exploration," in *Proc. 68th Int. Astronautical Congr. (IAC)*, Adelaide, Australia, Sep. 25–29, 2017.
[8] M. Hutter et al., "Anymal-a highly mobile and dynamic quadrupedal robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS).* Piscataway, NJ, USA: IEEE Press, 2016, pp. 38–44.
[9] C. D. Remy, "Optimal exploitation of natural dynamics in legged locomotion," Ph.D. dissertation, Switzerland, ETH Zurich, 2011.
[10] C. Mastalli et al., "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA).* Piscataway, NJ, USA: IEEE Press, 2020, pp. 2536–2542.
[11] J. Hwangbo et al., "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaau5872.
[12] 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. (ICRA).* Piscataway, NJ, USA: IEEE Press, 2017, pp. 93–100.
[13] F. Farshidian, M. Kamgarpour, D. Pardo, and J. Buchli, "Sequential linear quadratic optimal control for nonlinear switched systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1463–1469, 2017.
[14] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Sci. Robot.*, vol. 9, no. 88, 2024, Art. no. eadi7566.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[15] 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. (ICRA)*, 2014, pp. 6476–6482.
[16] M. Vukobratović and B. Borovac, "Zero-moment point—thirty five years of its life," *Int. J. Humanoid Rob.*, vol. 1, no. 01, pp. 157–173, 2004.
[17] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Adv. Rob.*, vol. 24, no. 5-6, pp. 719–737, 2010.
[18] C. Mastalli, S. P. Chhatoi, T. Corbéres, S. Tonneau, and S. Vijayakumar, "Inverse-dynamics mpc via nullspace resolution," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 3222–3241, Apr. 2023.
[19] Z. Manchester, N. Doshi, R. J. Wood, and S. Kuindersma, "Contact-implicit trajectory optimization using variational integrators," *Int. J. Robot. Res.*, vol. 38, no. 12–13, pp. 1463–1476, 2019.
[20] A. Aydinoglu and M. Posa, "Real-time multi-contact model predictive control via admm," in *Proc. Int. Conf. Robot. Autom. (ICRA).* Piscataway, NJ, USA: IEEE Press, 2022, pp. 3414–3421.
[21] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 69–81, 2014.
[22] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.
[23] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* Piscataway, NJ, USA: IEEE Press, 2012, pp. 4906–4913.
[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. Automat. Lett.*, vol. 3, no. 3, pp. 1560–1567, Mar. 2018.
[25] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler, "Contact-implicit trajectory optimization using orthogonal collocation," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 2242–2249, Feb. 2019.
[26] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 279–286.
[27] S. Wei, K. Uthaichana, M. Žefran, R. A. DeCarlo, and S. Bengea, "Applications of numerical optimal control to nonlinear hybrid systems," *Nonlinear Anal. Hybrid Syst.*, vol. 1, no. 2, pp. 264–279, 2007.
[28] J. Wang et al., "Automatic gait pattern selection for legged robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS).* Piscataway, NJ, USA: IEEE Press, 2020, pp. 3990–3997.
[29] M. Neunert et al., "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1458–1465, Mar. 2018.
[30] S. L. Cleac'h et al., "Fast contact-implicit model predictive control," *IEEE Trans. Robot.*, vol. 40, pp. 1617–1629, 2024.
[31] N. J. Kong, C. Li, G. Council, and A. M. Johnson, "Hybrid ilqr model predictive control for contact implicit stabilization on legged robots," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4712–4727, Dec. 2023.
[32] G. Kim, D. Kang, J.-H. Kim, S. Hong, and H.-W. Park, "Contact-implicit model predictive control: Controlling diverse quadruped motions without pre-planned contact modes or trajectories," *Int. J. Robot. Res.*, vol. 44, no. 3, pp. 486–510, 2025.
[33] K. Green, Y. Godse, J. Dao, R. L. Hatton, A. Fern, and J. Hurst, "Learning spring mass locomotion: Guiding policies with a reduced-order model," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3926–3932, Feb. 2021.
[34] R. Batke et al., "Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning," in *Proc. IEEE-RAS 21st Int. Conf. Humanoid Robots (Humanoids).* Piscataway, NJ, USA: IEEE Press, 2022, pp. 714–721.
[35] D. Youm, H. Jung, H. Kim, J. Hwangbo, H.-W. Park, and S. Ha, "Imitating and finetuning model predictive control for robust and symmetric quadrupedal locomotion," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7799–7806, Nov. 2023.
[36] Y. Fuchioka, Z. Xie, and M. Van de Panne, "Opt-mimic: Imitation of optimized trajectories for dynamic quadruped behaviors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA).* Piscataway, NJ, USA: IEEE Press, 2023, pp. 5092–5098.
[37] Z. Xie, X. Da, B. Babich, A. Garg, and M. V. de Panne, "Glide: Generalizable quadrupedal locomotion in diverse environments with a centroidal model," in *Proc. Int. Workshop Algorithmic Foundations Robot.* New York: Springer, 2022, pp. 523–539.
[38] P. Arena, A. Li Noce, and L. Patanè, "Stability and safety learning methods for legged robots," *Robotics*, vol. 13, no. 1, 2024, Art. no. 17.
[39] J. Ding, M. A. v Löben Sels, F. Angelini, J. Kober, and C. Della Santina, "Robust jumping with an articulated soft quadruped via trajectory optimization and iterative learning," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 255–262, Feb. 2024.
[40] C. Della Santina and F. Angelini, "Iterative learning in functional space for non-square linear systems," in *Proc. 60th IEEE Conf. Decis. Control (CDC).* Piscataway, NJ, USA: IEEE Press, 2021, pp. 5858–5863.
[41] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, 2nd ed. Springer Handbooks. Cham: Springer Int. Publishing AG, 2016.
[42] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph. (ToG)*, vol. 31, no. 4, pp. 1–8, 2012.
[43] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA).* Piscataway, NJ, USA: IEEE Press, 2014, pp. 1168–1175.
[44] S. P. Chhatoi, M. Pierallini, F. Angelini, C. Mastalli, and M. Garabini, "Optimal control for articulated soft robots," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3671–3685, May 2023.
[45] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast Motions in Biomechanics Robotics: Optim. Feedback Control*, M. Diehl and K. Mombaur, Eds., vol 340. Berlin, Heidelberg: Springer, 2006, pp. 65–93.
[46] M. Pierallini, F. Angelini, R. Mengacci, A. Palleschi, A. Bicchi, and M. Garabini, "Iterative learning control for compliant underactuated arms," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 6, pp. 3810–3822, Jun. 2023.
[47] 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. (ICRA)*, 2017, pp. 93–100.
[48] M. A. Sen, V. Bakircioglu, and M. Kalyoncu, "Inverse kinematic analysis of a quadruped robot," *Int. J. Scientific Technol. Res.*, vol. 6, no. 9, pp. 285–289, 2017.
[49] M. Pierallini, F. Angelini, R. Mengacci, A. Palleschi, A. Bicchi, and M. Garabini, "A robust iterative learning control for continuous-time nonlinear systems with disturbances," *IEEE Access*, vol. 9, pp. 147471–147480, 2021.
[50] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: A software framework for nonlinear optimization and optimal control," *Math. Program. Computat.*, vol. 11, pp. 1–36, 2019.
[51] A. Wächter, "Short tutorial: Getting started with ipopt in 90 minutes," in *Proc. Dagstuhl Seminar Proc. Schloss Dagstuhl-Leibniz-Zentrum Fuer Informatik*, 2009.
[52] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Trans. Syst., Man, Cybern., Part C Appl. Rev.*, vol. 37, no. 6, pp. 1099–1121, Jun. 2007.
[53] F. Angelini et al., "Time generalization of trajectories learned on articulated soft robots," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3493–3500, Feb. 2020.
[54] F. Angelini, C. Della Santina, M. Garabini, M. Bianchi, and A. Bicchi, "Control architecture for human-like motion with applications to articulated soft robots," *Front. Robot. AI*, vol. 7, 2020, Art. no. 117.
[55] A. Pandala, R. T. Fawcett, U. Rosolia, A. D. Ames, and K. A. Hamed, "Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced-and full-order models," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6622–6629, Mar. 2022.