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

Soft Bilinear Inverted Pendulum: A Model to Enable Locomotion With Soft Contacts

Davide De Benedittis, Franco Angelini, *Member, IEEE*, and Manolo Garabini, *Member, IEEE*

***Abstract*—The robotics research community has developed several effective techniques for quadrupedal locomotion. Most of these methods ease the modeling and control problem by assuming a rigid contact between the feet and the terrain. However, in the case of compliant terrain or robots equipped with soft feet, this assumption no longer holds, as the contact point moves and the reaction forces experience a delay. This article presents a novel approach for quadrupedal locomotion in the presence of soft contacts. The control architecture consists of two blocks: 1) upstream, the motion planner (MP) computes a feasible trajectory using model predictive control (MPC) and 2) downstream, the tracking controller (TC) employs hierarchical optimization (HO) to achieve motion tracking. This choice allows the control architecture to employ a large time horizon without heavily compromising the model's accuracy. For the first time, both blocks consider the contact compliance: in the MP, the classic linear inverted pendulum model is extended by proposing the soft bilinear inverted pendulum (SBIP) model; conversely, the TC is a whole-body controller (WBC) that considers the full dynamics model, including the soft contacts. Simulations with multiple quadrupedal robots demonstrate that the proposed approach enables traversing soft terrains with improved stability and efficiency. Furthermore, the performance benefits of including the compliance in the MP and TC are evaluated. Finally, experiments on the SOLO12 robot walking on soft terrain validate the proposed approach's effectiveness.**

***Index Terms*—Contacts, legged locomotion, optimal control, predictive control, quadratic programming.**

I. Introduction

BIOLOGICAL systems leverage body compliance to achieve highly robust and energy-efficient locomotion. The knowledge about the advantages provided by softness, which have been recognized and widely studied in scientific literature [1], [2], [3], [4], [5], [6], is pushing toward the adoption of principles to exploit softness into robotic quadrupedal locomotion.

Indeed, modern quadrupedal robots are benefiting from the integration of compliant elements, which greatly enhance their performance and enable dynamic gaits and maneuvers. Examples of this are HyQ's linear elastic ankle joint [7], [8], Cheetah's flexible spine and passive rotary ankle joints [9], [10], or ANYmal's ANYdrive series elastic actuators [11]. The increased agility of legged systems has granted them the ability to traverse rough and complex environments.

Following a similar idea, researchers started to include compliance in the foot design, developing more sophisticated robotic feet inspired by the animal kingdom [12], [13], [14], [15], [16]. Although the results are promising, using more complex feet geometries introduces new control challenges that are currently unaddressed.

One of the main issues when designing a locomotion controller for quadrupeds equipped with bioinspired feet is considering the contact softness. These feet exploit compliance [16] and adapt to the terrain profile [13], violating the rigid contact model's hypotheses. Similarly, legged robots traversing compliant terrains do not accurately fit the rigid contact hypothesis. This occurs, for instance, when locomoting on sand, mud, snow, or dense vegetation [17], which are typical use cases for legged robots. In these conditions, legged robots may experience reduced stability and worse performance [18].

Currently, most of the literature does not consider soft contacts due to the convenience of the rigid contact model. For instance, with model-based approaches, employing a rigid contact enables projecting the system dynamics in the null space of the contact constraints, reducing the optimization vector size [19]. On the other hand, model-free methods typically require a simulation training phase, which is computationally heavier and more challenging when considering soft contacts. Even when compliance is indeed considered, it is typically done only in a few selected parts of the control pipeline. For instance, in the case of frameworks composed of a motion planner (MP) and a tracking controller (TC), contact softness is usually included only in one of the two blocks; e.g., in [20] only the TC is explicitly adapted for dealing with softness.

Moreover, no work has systematically studied the performance of legged robots on soft grounds. This gap in the literature suggests that the problem introduced by significant compliance, albeit relevant, has not been fully addressed. Nonetheless, lifting the rigid feet–terrain contact

Received 7 September 2024; accepted 17 November 2024. This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme (Natural Intelligence) under Grant 101016970; in part by the European Union's HORIZON-MSCA-2023-SE-01-01 - MSCA Staff Exchanges 2023 Program under Grant 101182891 (NEUTRAWEED); in part by the European Union's Next Generation EU "Ecosistema dell'Innovazione" Tuscany Health Ecosystem (THE, PNRR, Spoke 4: Spoke 9: Robotics and Automation for Health) under Project ECS00000017; and in part by the Italian Ministry of Education and Research in the Framework of the "FoReLab" (Future-Oriented Research Laboratory) Project (Departments of Excellence). This article was recommended by Associate Editor C.-C. Tsai. *(Corresponding author: Davide De Benedittis.)*

The authors are with the Centro di Ricerca "E. Piaggio" and the Dipartimento di Ingegneria dell'Informazione, Università di Pisa, 56122 Pisa, Italy (e-mail: davide.debenedittis@phd.unipi.it).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TSMC.2024.3504342.

Digital Object Identifier 10.1109/TSMC.2024.3504342

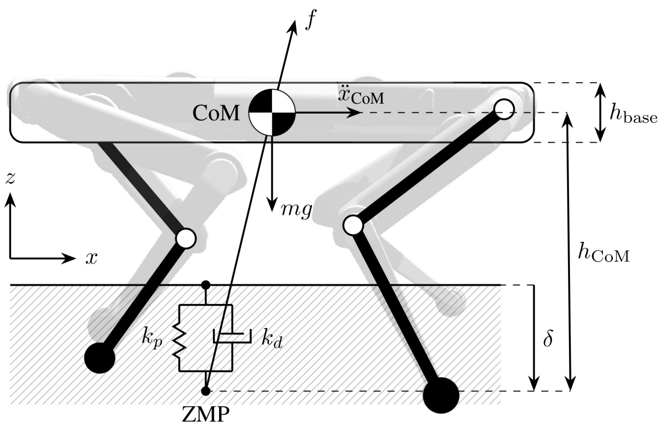


Fig. 1. SBIP model scheme.

hypothesis can increase robot stability and performance in real-world scenarios by covering the mismatch with the actual compliant interaction with the terrain [20]. Our results, both in simulation (Section VI) and experiments (Section VII), further suggest that considering compliance can reap significant improvements.

This article presents a novel control framework for quadrupedal locomotion that accounts for soft contacts both in the MP and the TC. The original contributions are as follows.

1) The integration of soft contacts in both the MP and the TC. Unlike prior approaches that focus on either the MP or the TC independently, the proposed framework considers and adapts the two-control-blocks approach in all its parts. Furthermore, the impact of this integration in the two blocks is quantitatively analyzed.
2) The introduction of the soft bilinear inverted pendulum (SBIP), a novel model that accounts for the foot–terrain interpenetration. This model (see Fig. 1) treats a legged robot as a point mass with a mass-less rigid body attached to it. The point mass represents the robot's base, while the pendulum body represents the robot's legs. The pendulum is in contact with the ground through a spring-damper system, which models the interaction between the pendulum and the terrain.
3) The usage of the proposed SBIP model in the MP. Additionally, the MP considers how the terrain deformations affect the contact plane. This approach enables the generation of robust and efficient trajectories of the robot's base and feet when traversing soft terrains.
4) The inclusion of a soft contact model in the whole-body controller (WBC). The WBC employs hierarchical quadratic programming (HQP) to optimize the control with multiple concurrent and conflicting tasks.
5) The quantitative analysis of the performance improvements achieved by considering compliance in each block of the two-control-blocks approach.

It is shown through exhaustive simulations and experiments that the proposed approach aids locomotion when traversing compliant terrains.

The remainder of this article is organized as follows. Section II provides a literature overview of the main existing approaches for locomotion control that account for soft contacts. Section III introduces the necessary theoretical concepts used to formally describe the problem that is tackled. Section IV-C introduces the SBIP model and describes the MP. Subsequently, Section V presents the proposed hierarchical optimization (HO)-based soft WBC. In Section VI, the proposed controller is validated through simulations on two quadrupeds, comparing the proposed framework with a "standard" control approach. Section VII presents the experimental validation of the proposed framework on the SOLO12 robot. Finally, Section VIII concludes this article.

II. State of the Art

Literature shows that compliance is fundamental to achieving efficient quadrupedal locomotion [1], [2], [3], [4], [5], [6]. However, it is usually inserted and modeled only in the robot structure, while feet–terrain contacts are typically assumed to be rigid. For instance, the planning of the robot base and swing leg trajectories, commonly computed through the seminal linearized inverted pendulum (LIP) model [21], has been extended to consider the joint flexibility in the so-called flexible linear inverted pendulum (FLIP) [22], [23]. Alternatively, center of mass (CoM) oscillations have been captured through the spring-loaded inverted pendulum (SLIP) [24]. However, none of these articles consider soft contacts. For this reason, this work proposes the SBIP model (Fig. 1). This model enables simultaneously planning the robot base and swing feet trajectories while accounting for terrain penetration or foot deformation.

Soft contacts have been discussed in a limited number of state-of-the-art articles. Kim et al. [25] proposed a WBC, i.e., a TC, in which the contact constraint is relaxed. By dropping the rigid contact assumption, the contact point is allowed to move, while its acceleration is minimized. Although this approach does not include a physically meaningful contact model, nor does it explicitly address the problem of soft contacts, the control commands are optimized to reduce the jerk during transitions of the contact feet.

An extended multipart (x-MP) controller that uses a nonlinear contact model is proposed for monopod hopping robots in [26] and for quadrupedal robots in [27]. While this method yields promising results using an unconventional controller and contact model, it is limited by the unique robot design, making it unsuitable for most traditional quadrupeds.

In [28], vertical jumping on granular media is studied. The deformable terrain dynamics is learned with a kernel machine, while the motion planning is solved with nonlinear programming (NLP). This approach attempts to lay the foundations for concurrently learning and exploiting the terrain dynamics, but it is not applied to realistic scenarios.

A momentum-based balancing controller for legged robots on compliant terrains is developed in [29]. Another admittance optimization-based controller for balance control of bipedal robots under linear visco-elastic contacts is proposed in [30]. These papers use a seminal nonlinear contact model and formulate an optimization controller; however, the nonlinearity of the optimization problem limits its real-time performance and the only task considered is the balance control.

TABLE I
LITERATURE OVERVIEW

Name	HO	MP+TC	SP	SC
Bellicoso et al. [34]	✓	✓	×	×
Vasilopoulos et al. [27]	×	×	✓	NA
Neunert et al. [31]	×	✓	×	✓
Fahmi et al. [20]	×	✓	×	✓
Proposed approach	✓	✓	✓	✓

The nonlinear model predictive controller (NMPC) described in [31] includes a nonlinear contact model in its problem formulation. However, the contact model is not used to address compliant contacts but only to adhere to the constraints imposed by the nonlinear optimization technique and reduce the computational effort.

More recently, the QP-based WBC framework proposed in [32] was extended to ensure consistency with terrain compliance in [20]. This controller is tested in experiments on different terrains and shows improved performance with the presence of softness. However, the MP does not account for the terrain compliance. Moreover, the WBC is formulated as a single QP problem with constraint softening. Thus, it does not ensure strict task prioritization but only a soft hierarchy.

Finally, Choi et al. [33] used reinforcement learning (RL) with a granular media model to develop a controller for compliant and granular terrains. This approach demonstrated impressive results on sand, which is typically a very challenging environment, subject to the usual pros and cons of RL-based approaches.

Table I reports a summary of the main model-based controllers for quadrupedal locomotion. The column HO specifies if the method can deal with task prioritization. Column MP + TC is true when the control is divided into an MP and a TC. SP stands for soft motion planner, and SC for soft tracking controller, and they are true when the MP or TC explicitly consider soft contacts, respectively.

In this work, the soft contact model is included both in the MP and in the TC. The MP exploits the proposed SBIP model, while the TC is a WBC augmented with the soft contact model introduced in Section III-B3.

III. PROBLEM FORMULATION

This section formally introduces the problem addressed in this work. That is, given a mathematical description of the robot (Section III-A) and foot–terrain contact model (Section III-B), find a suitable robot trajectory that allows it to locomote and the joint torques required to track this trajectory.

A. Model of the Robot

Let the robot generalized coordinates vector $\boldsymbol{q}$ and the generalized velocity vector $\boldsymbol{\upsilon}$ be

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{p}_b \\ \boldsymbol{q}_b \\ \boldsymbol{q}_j \end{bmatrix} \in SE(3) \times \mathbb{R}^{n_j}, \quad \boldsymbol{\upsilon} = \begin{bmatrix} \boldsymbol{\upsilon}_b \\ {}_B\boldsymbol{\omega}_b \\ \dot{\boldsymbol{q}}_j \end{bmatrix} \in \mathbb{R}^{n_v} \tag{1}$$

where $n_v = n_j + 6$, with n_j the number of joints. The vector $\boldsymbol{p}_b$ is the position of the robot base, $\boldsymbol{q}_b$ is the base orientation quaternion, and $\boldsymbol{q}_j$ is the leg joint position vector. The linear velocity of the base is denoted by $\boldsymbol{\upsilon}_b$, the angular velocity of the base expressed in the base frame B is ${}_B\boldsymbol{\omega}_b$, and $\dot{\boldsymbol{q}}_j$ is the legs joint velocity vector.

The equations of motion (EoM) of a legged robot are

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\upsilon}} + \boldsymbol{h}(\boldsymbol{q}, \boldsymbol{\upsilon}) = \boldsymbol{S}^{\mathsf{T}}\boldsymbol{\tau} + \boldsymbol{J}_c^{\mathsf{T}}(\boldsymbol{q})\boldsymbol{f}_c. \tag{2}$$

Here, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n_v \times n_v}$ is the generalized mass matrix, and $\boldsymbol{h}(\boldsymbol{q}, \boldsymbol{\upsilon}) \in \mathbb{R}^{n_v}$ is the vector of Coriolis, centrifugal and gravitational terms. The matrix $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{0}_{n_j \times (n_v - n_\tau)} & \boldsymbol{I}_{n_j \times n_j} \end{bmatrix}$ is the so-called selection matrix, $\boldsymbol{\tau} \in \mathbb{R}^j$ is the joint torques vector, $\boldsymbol{J}_c(\boldsymbol{q}) \in \mathbb{R}^{n_v \times 3n_c}$ is the contact Jacobian matrix, $\boldsymbol{f}_c \in \mathbb{R}^{3n_c}$ is the vector of contact forces, and n_c is the number of feet in contact with the ground.

B. Contact Model

Due to its convenience, the rigid contact model has become the de-facto standard for modeling the robot-terrain interaction during locomotion. Only more recently the development of control algorithms that consider the softness of the contacts has gained some attraction.

1) Rigid Contact Model: This model consists in a monolateral kinematic constraint that imposes zero velocity at the contact: $\boldsymbol{p}_c = \boldsymbol{0}$. Differentiating it twice w.r.t. time t yields a constraint on the generalized joint accelerations $\dot{\boldsymbol{\upsilon}}$, i.e.,

$$\ddot{\boldsymbol{p}}_c = \boldsymbol{J}_c\dot{\boldsymbol{\upsilon}} + \dot{\boldsymbol{J}}_c\boldsymbol{\upsilon} = \boldsymbol{0} \tag{3}$$

where $\ddot{\boldsymbol{p}}_c$ is the contact point acceleration and $\dot{\boldsymbol{J}}_c$ is the time derivative of the Jacobian matrix $\boldsymbol{J}_c$.

However, the assumption of zero velocity at the contact is often not reflected in the reality of foot–terrain interactions. This justifies the need for lifting the rigid constraint.

2) Kelvin–Voight Model: The Kelvin–Voight (KV) model [35], also known as the spring–damper model, is a soft contact model that handles the contact between two bodies by introducing linear springs and dampers in parallel in the three Cartesian directions. This model defines the interaction force as a function of the foot–terrain interpenetration $\boldsymbol{\delta}$ and of its time-derivative $\dot{\boldsymbol{\delta}}$, i.e.,

$$\boldsymbol{f}_c = \boldsymbol{K}_p\boldsymbol{\delta} + \boldsymbol{K}_d\dot{\boldsymbol{\delta}} \tag{4}$$

where $\boldsymbol{K}_p, \boldsymbol{K}_d \in \mathbb{R}^{3\times3}$ are diagonal matrices representing the stiffness and the damping coefficients of the contact.

The KV model, widely used for approximating soft contact, is simple and linear in its parameters, making it an ideal candidate for use in quadratic optimizations.

3) Monodirectional Spring-Damper Contact Model: This model describes a soft interaction only in the direction perpendicular to the terrain plane, modeling it with a linear spring and damper in parallel. Conversely, the contact is modeled as rigid in the horizontal direction. Considering $\boldsymbol{h}$, $\boldsymbol{l}$, and $\boldsymbol{n}$ as the heading, lateral, and terrain normal direction unit vectors, respectively, the interaction with the terrain is governed by

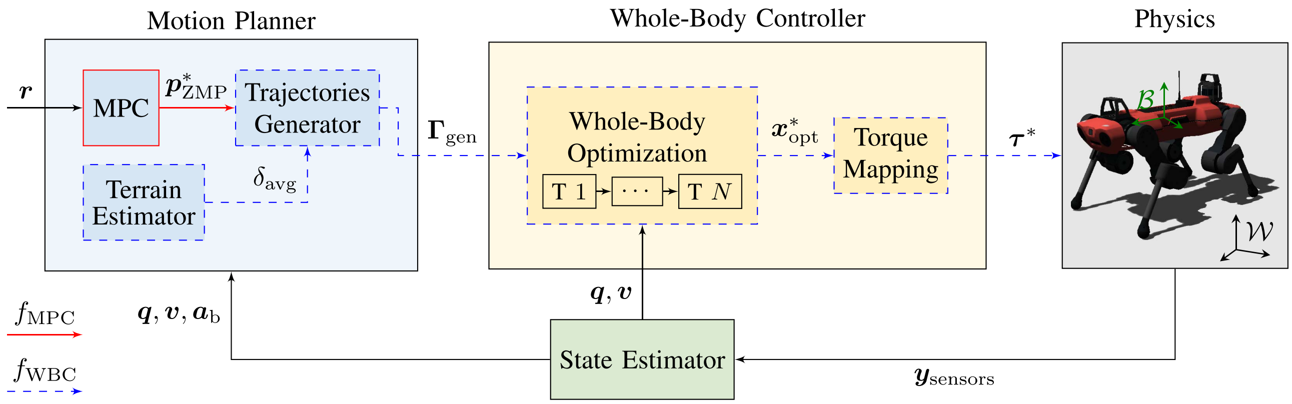


Fig. 2. Overview of the control architecture. Given the reference signal $\boldsymbol{r}$, the MP generates the generalized trajectory $\boldsymbol{\Gamma}_{\mathrm{gen}}$ consisting of the base and swing feet motions. Then, the WBC computes the torques $\boldsymbol{\tau}^*$ that track the given trajectory $\boldsymbol{\Gamma}_{\mathrm{gen}}$. f_{MPC} and f_{WBC} are the frequencies of the MP MPC and of the other control components, respectively.

$$\begin{aligned} \boldsymbol{f}_c \cdot \boldsymbol{n} &= k_p\delta_n + k_d\dot{\delta}_n \\ \ddot{\boldsymbol{p}}_c \cdot \boldsymbol{h} &= \left(\boldsymbol{J}_c\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_c\boldsymbol{v}\right)\cdot \boldsymbol{h} = 0 \\ \ddot{\boldsymbol{p}}_c \cdot \boldsymbol{l} &= \left(\boldsymbol{J}_c\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_c\boldsymbol{v}\right)\cdot \boldsymbol{l} = 0 \end{aligned} \tag{5}$$

where δ_n and $\dot{\delta}_n$ are the interpenetration and its time-derivative in the normal direction, respectively. Additionally, k_p and k_d are the stiffness and damping coefficients in the terrain's normal direction. Finally, given two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, $\boldsymbol{a}\cdot\boldsymbol{b}$ represents the dot product between them.

This contact model is often used in simulators when simulating both rigid—i.e., large values of k_p and k_d—and soft contacts. Moreover, it is a good approximation of the behavior of a terrain whose softness is mainly in the normal direction.

C. Solution Approach

This work tackles the control problem of computing the joint torques $\boldsymbol{\tau}^*$ that achieve stable locomotion while satisfying physical constraints, given the floating-base dynamic model of the robot (2) and a soft contact model (5).

This problem can be addressed with the usage of two separate controllers, allowing for considering both long-term dynamics and short-term constraints. The resulting control architecture is represented in Fig. 2, which also highlights the frequencies of the various control blocks (f_{MPC} and f_{WBC}). This problem is not fully solved in traditional model-based approaches. To the best of the authors' knowledge, the majority of current control frameworks fail to address the presence of compliant contacts, neglecting it. Recently, some recent researchers have tackled this problem, but most of them do not adopt a two-control-blocks-based approach, hindering their performance or real-time feasibility. In [20], the control problem is split into two parts, but the softness is considered only in the controller downstream. Upstream, the MP employs the proposed SBIP, that expands the LIP (see Section IV-A) with a soft contact model. This allows to consider movements of the contact point and noninstantaneously varying forces. The MP employs model predictive control (MPC) to plan the base trajectory (Section IV-B) and heuristics to generate the feet trajectories (Section IV-C) while taking into account the terrain deformation (Section IV-D). Downstream, the TC uses quadratic programming (Section V-A) and HO (Section V-B) to solve all the tasks (Section V-D). In particular, one task enforces a soft contact model to consider the delay of the reaction force and the contact point movement. The proposed framework works using either the KV model or the monodirectional spring-damper model. However, without loss of generality, the rest of this article will focus on the monodirectional spring-damper model.

IV. Motion Planner

This section describes the MP block in Fig. 2. Its goal is to generate a proper base and swing feet trajectory ($\boldsymbol{\Gamma}_{\mathrm{gen}}$) that follows the input reference $\boldsymbol{r} = \begin{bmatrix} \upsilon_h & \upsilon_l & \dot{\varphi} \end{bmatrix}^\intercal$, where υ_h and υ_l are the commanded longitudinal velocity and lateral velocity, and $\dot{\varphi}$ is the reference yaw rate. The following focuses on a trotting gait, but the proposed framework could be easily extended to other gait types.

In the following MPC formulation, the robot is represented as a SBIP, a novel model hereby developed to address the issues connected with soft contacts. It builds on the LIP model [21], which assumes that the entire mass of the robot is located in the robot base, neglecting the limb's inertial effects. Moreover, it disregards the rotational inertia of the base and restricts the motion to maintain a constant base height. By solving the MPC problem, the desired zero moment point (ZMP) position at the next footstep is obtained, denoted as $\boldsymbol{p}^*_{\mathrm{ZMP},1}$. The leg's desired trajectories are computed from $\boldsymbol{p}^*_{\mathrm{ZMP},1}$, while the base desired trajectory is obtained from the SBIP model. The trajectory of the base and feet constitute the so-called generalized trajectory $\boldsymbol{\Gamma}_{\mathrm{gen}}$.

A. Soft Bilinear Inverted Pendulum Model

The SBIP consists in a LIP that lies on a soft surface, modeled as a linear spring-damper system in the normal direction (Fig. 1). The dynamic equations are

$$\ddot{x}_{\mathrm{CoM}} = \frac{g + \ddot{z}_{\mathrm{CoM}}}{h_{\mathrm{CoM}}}\left(x_{\mathrm{CoM}} - x_{\mathrm{ZMP}}\right) \tag{6}$$

$$\ddot{y}_{\mathrm{CoM}} = \frac{g + \ddot{z}_{\mathrm{CoM}}}{h_{\mathrm{CoM}}}\left(y_{\mathrm{CoM}} - y_{\mathrm{ZMP}}\right) \tag{7}$$

$$\ddot{z}_{\mathrm{CoM}} = -\ddot{\delta} = -g + \frac{1}{m}\left(k_p\delta + k_d\dot{\delta}\right) \tag{8}$$

where x_{CoM}, y_{CoM}, and z_{CoM} are the coordinates of the CoM of the robot base, x_{ZMP} and y_{ZMP} are the coordinates of the ZMP, g is the module of the gravity acceleration, m is the total mass of the robot, and h_{CoM} is the height of the CoM relative to the terrain. Finally, the terrain deformation in the SBIP contact point is δ, while k_p and k_d are the terrain stiffness and damping, respectively.

Despite being a relatively simple model, the SBIP has no general closed-form solution. However, it can be noted that the dynamic along the z-axis is independent of the dynamics in the horizontal directions. It follows that the dynamics in the x and y directions are decoupled from each other. Additionally, the behavior in the vertical direction is that of a mass-spring-damper model and can be easily analytically solved. Afterward, the solution can be plugged into the horizontal dynamics, which can be solved numerically.

An alternative approach consists in neglecting the effect of the vertical displacement on the horizontal dynamics, resulting in a system with a linear closed-form solution. Defining $\boldsymbol{x}_{\mathrm{CoM}}(t) = \begin{bmatrix} x_{\mathrm{CoM}}(t) & \dot{x}_{\mathrm{CoM}}(t) \end{bmatrix}^{\mathsf{T}}$, the horizontal dynamics in the x direction can be written as

$$\boldsymbol{x}_{\mathrm{CoM}}(t) = \boldsymbol{A}(t)\boldsymbol{x}_{\mathrm{CoM}}(0) + \boldsymbol{B}(t)x_{\mathrm{ZMP}} \tag{9}$$

where

$$\boldsymbol{A}(t) = \begin{bmatrix} \cosh(\omega t) & \omega^{-1}\sinh(\omega t) \\ \omega \sinh(\omega t) & \cosh(\omega t) \end{bmatrix}, \ \boldsymbol{B}(t) = \begin{bmatrix} 1-\cosh(\omega t) \\ -\omega\sinh(\omega t) \end{bmatrix} \tag{10}$$

with $\omega = \sqrt{g/h_{\mathrm{CoM}}}$. Following similar steps it is possible to write the dynamics in the y direction.

B. Model Predictive Controller

A trotting gait with a fixed swing duration T_s and instant contact-swing shifting is assigned to the robot. The controller is used to compute the desired future ZMP positions that "catch" the falling CoM while tracking the velocity command. The SBIP dynamics are used to compute the base trajectory, while the desired foothold positions are calculated from the desired ZMP position. Afterward, a polynomial interpolation is used to generate the swing trajectory (Section IV-C).

The MPC considers a time horizon of N footsteps, and each optimization step refers to one footstep. Since in the SBIP model the x and y coordinates are decoupled, it is possible to divide the problem into two identical subproblems in the two directions and solve them separately. In the following, only the MPC for the x coordinate will be written, but analogous considerations for the y coordinate are valid. The multiple shooting approach is used, thus the optimization variable is comprised of the input $x_{\mathrm{ZMP},i}$ and the state $\boldsymbol{x}^{\mathsf{T}}_{\mathrm{CoM},i}$ at the sample points $i = 1, \dots, N$, i.e.,

$$\boldsymbol{x}_{\mathrm{opt}} = \begin{bmatrix} x_{\mathrm{ZMP},1} & \boldsymbol{x}^{\mathsf{T}}_{\mathrm{CoM},1} & \cdots & x_{\mathrm{ZMP},N} & \boldsymbol{x}^{\mathsf{T}}_{\mathrm{CoM},N} \end{bmatrix}^{\mathsf{T}}. \tag{11}$$

The MPC optimization problem is formulated as a QP problem. The tasks are the system dynamics, the kinematic limits of the feet displacement, the base's linear velocity reference tracking, and the step length minimization.

1) System Dynamics: This task guarantees that the solution is consistent with the SBIP model. $\boldsymbol{x}_{\mathrm{CoM},i}$ is the CoM position at the end of the ith footstep, where the current footstep is the 0th, i.e.,

$$\begin{aligned} \boldsymbol{x}_{\mathrm{CoM},i} &= \boldsymbol{A}(T_s)\boldsymbol{x}_{\mathrm{CoM},i-1} + \boldsymbol{B}(T_s)x_{\mathrm{ZMP},i-1}, \ \ i = 1,\dots,N \\ \boldsymbol{x}_{\mathrm{CoM},0} &= \boldsymbol{A}(t_{\mathrm{rem}})\boldsymbol{x}_{\mathrm{CoM}}(0) + \boldsymbol{B}(t_{\mathrm{rem}})x_{\mathrm{ZMP},0} \end{aligned} \tag{12}$$

where t_{rem} is the remaining time until the end of the current step and $\boldsymbol{x}_{\mathrm{CoM},0} = \boldsymbol{x}_{\mathrm{CoM}}(0 + t_{\mathrm{rem}})$ is the x-coordinate of the CoM at the end of the current step. $\boldsymbol{x}_{\mathrm{CoM}}(0)$ is the current CoM state, $x_{\mathrm{ZMP},0}$ is the ZMP coordinate obtained from (6), $x_{\mathrm{ZMP},i}$ is the x-coordinate of the ZMP after i steps, and $i = 1, \dots, N$.

2) Kinematic Limits: This task guarantees that the robot does not take a step longer than its maximum step length d

$$x_{\mathrm{ZMP},i} - x_{\mathrm{ZMP},i-1} \leq d, \qquad i = 1,\dots,N. \tag{13}$$

3) Velocity Reference Tracking: It imposes that the robot follows the given linear velocity reference:

$$\dot{\boldsymbol{x}}_{\mathrm{CoM},i} = \dot{\boldsymbol{x}}^{\mathrm{des}}, \qquad i = 1,\dots,N. \tag{14}$$

4) Step Length Minimization: It imposes that the robot minimizes the step length to avoid unnecessary movements. This task is in conflict with the other tasks and is weighed with a lower weight to prevent affecting the locomotion negatively

$$x_{\mathrm{ZMP},i} - x_{\mathrm{ZMP},i-1} = 0, \qquad i = 1,\dots,N. \tag{15}$$

The overall QP problem is written as

$$\begin{aligned} \min_{\boldsymbol{x}_{\mathrm{opt}}} \ & \frac{1}{2}\sum_{i=1}^{N}\Big(\|x_{\mathrm{ZMP},i} - x_{\mathrm{ZMP},i-1}\|^2_{\boldsymbol{Q}_{\min}} + \|\dot{\boldsymbol{x}}_{\mathrm{CoM},i} - \dot{\boldsymbol{x}}^{\mathrm{des}}\|^2_{\boldsymbol{Q}_{\mathrm{des}}} \\ & + \|\boldsymbol{x}_{\mathrm{CoM},i} - \boldsymbol{A}(T_s)\boldsymbol{x}_{\mathrm{CoM},i-1} - \boldsymbol{B}(T_s)x_{\mathrm{ZMP},i}\|^2_{\boldsymbol{Q}_{\mathrm{dyn}}}\Big) \\ \text{s.t.} \ \ & x_{\mathrm{ZMP},i} - x_{\mathrm{ZMP},i-1} \leq d, \qquad i = 1,\dots,N \end{aligned} \tag{16}$$

where the $\|\cdot\|^2_{\boldsymbol{Q}}$ indicates the argument's two-norm weighted by the matrix $\boldsymbol{Q}$. Additionally, $\boldsymbol{Q}_{\mathrm{dyn}}$, $\boldsymbol{Q}_{\mathrm{des}}$, and $\boldsymbol{Q}_{\min}$ are the weighting matrices for the dynamics, the velocity reference tracking, and the step length minimization, respectively, and they are ordered with descending weight.

C. Trajectories Generation

The base horizontal trajectory is computed using the current measurements and the SBIP model. In the x direction it is

$$\boldsymbol{x}^{\mathrm{des}}_{\mathrm{CoM}} = \boldsymbol{A}(\Delta t)\ \boldsymbol{x}_{\mathrm{CoM}}(0)\ + \boldsymbol{B}(\Delta t)\ x_{\mathrm{ZMP},0} \tag{17}$$

where Δt is the reciprocal of the update frequency f_{WBC}, i.e., $\Delta t = 1/f_{\mathrm{WBC}}$.

The base vertical position is not constant while moving in the presence of soft contacts, e.g., on compliant terrains. Two constraints with decreasing priority should be satisfied

$$h_{\mathrm{CoM}} \leq h_{\mathrm{CoM,max}} \tag{18}$$

$$h_{\mathrm{CoM}} - \delta - \frac{h_{\mathrm{base}}}{2} \geq h_{cl} \tag{19}$$

where h_{CoM} is the CoM height relative to the contact point and $h_{\mathrm{CoM,max}}$ is the maximum base height (taking into account the kinematic limits of the robot). The terrain penetration is δ,

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 base height is h_{base}, and h_{cl} is the minimum required clearance with the terrain. The first constraint imposes that the base height does not violate the kinematic limits of the robot, while the second one imposes that the base maintains a certain distance from the terrain.

The desired footfall positions are computed from the ZMP position so that the next ZMP lies in the middle of the future contact feet. The footfall positions are

$$\begin{aligned} \boldsymbol{p}_{\text{LF}} &= \boldsymbol{p}^*_{\text{ZMP},1} + r\left[+\cos(\varphi + \Delta\gamma) \; +\sin(\varphi + \Delta\gamma)\right]^{\mathsf{T}} \\ \boldsymbol{p}_{\text{RF}} &= \boldsymbol{p}^*_{\text{ZMP},1} + r\left[+\cos(\varphi - \Delta\gamma) \; -\sin(\varphi - \Delta\gamma)\right]^{\mathsf{T}} \\ \boldsymbol{p}_{\text{LH}} &= \boldsymbol{p}^*_{\text{ZMP},1} + r\left[-\cos(\varphi - \Delta\gamma) \; +\sin(\varphi - \Delta\gamma)\right]^{\mathsf{T}} \\ \boldsymbol{p}_{\text{RH}} &= \boldsymbol{p}^*_{\text{ZMP},1} + r\left[-\cos(\varphi + \Delta\gamma) \; -\sin(\varphi + \Delta\gamma)\right]^{\mathsf{T}} \end{aligned} \tag{20}$$

where φ is the commanded yaw angle, computed by integrating the yaw rate command, while r and $\Delta\gamma$ are two parameters that represent the distance from the CoM and the angle about the heading direction when the robot is standing still.

The feet position is then interpolated from the initial to the desired foothold position using cubic splines. The interpolation is performed separately in the horizontal and vertical directions. The splines guarantee that the velocity profile is continuous and that the horizontal velocity is zero on impact.

When locomoting on very soft terrains, particular attention must be given to the vertical trajectory of the feet. High softness can lead to substantial penetrations along the vertical axis, posing challenges during the lifting of the feet.

Not accounting for this may prove inadequate for disengaging from the terrain, inducing undesired dragging of the feet or entailing entrapment within the terrain, potentially resulting in a failure of the robot. On the other hand, the vertical trajectory must respect kinematic and dynamic constraints, so that the vertical displacement is compatible with the gait. As a result of this tradeoff, the desired maximum step height (relative to the ground) is computed as

$$h_{\text{step}} = \begin{cases} \overline{h}_{\text{step}}, & \text{for } \delta_{\text{avg}} \leq 0 \\ \overline{h}_{\text{step}} + \delta_{\text{avg}}, & \text{for } 0 < \delta_{\text{avg}} < \delta h_{\max} \\ \overline{h}_{\text{step}} + \delta h_{\max}, & \text{for } \delta_{\text{avg}} \geq \delta h_{\max} \end{cases} \tag{21}$$

where h_{step} is the commanded step height and $\overline{h}_{\text{step}}$ is a default step height (chosen considering the robot kinematics and dynamics, and the terrain shape). The average penetration of the feet into the terrain is δ_{avg}, which is computed by the terrain estimator module (Section IV-D), while $\delta h_{\max}$ is the maximum step increase compatible with the robot kinematics and dynamics.

D. Terrain Estimation

Knowledge regarding the terrain plane is necessary to command the robot base and feet heights. As seen in Section IV-C, by having insights into the terrain characteristics and potential penetrations, the avoidance of any inadvertent collision between the terrain and the robot's base or feet can be ensured.

When a robotic system traverses soft terrains, it encounters a distinctive challenge—the contact plane is not stationary but rather moves beneath the robot. Consequently, it is imperative to account for the terrain motion in order to optimally locomote on compliant surfaces.

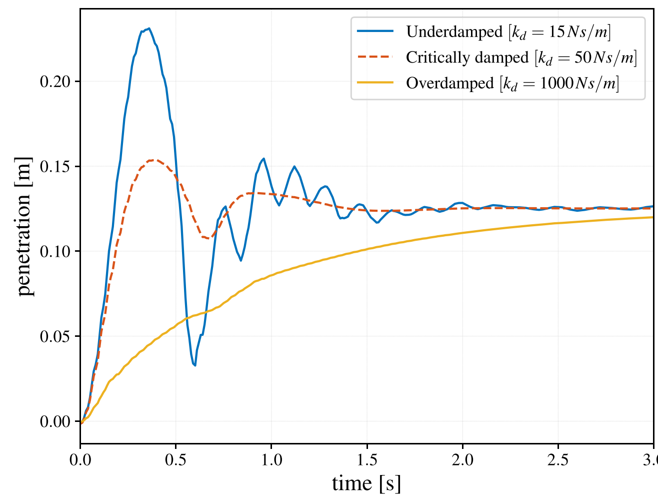


Fig. 3. Simulated terrain penetrations of the left-front foot with $k_p = 1000$ N/m and different dampings.

Three scenarios exist for the terrain-feet interaction. The contact dynamics can be either underdamped, critically damped, or overdamped (see Fig. 3). The end goal underlying the development of the terrain estimation algorithm is to ensure that the robot's commanded trajectory responds to terrain changes, all the while rejecting measurement noise and, more importantly, large oscillations in the underdamped case. Considering a contact plane that rapidly oscillates would cause the commanded base height to change just as rapidly, reducing stability and violating the SBIP model hypothesis.

The terrain plane estimator works under the assumption that the undeformed terrain is not "rapidly" changing. The terrain is obtained by finding the least squares plane given the four filtered contact points of the four feet. Considering a contact plane in the form of $ax + by + c = z$, and the filtered points (x_i, y_i, z_i) with $i = 1, \ldots, 4$, the contact plane coefficients are obtained by computing the least squares solution of

$$\begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_4 & y_4 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_4 \end{bmatrix}. \tag{22}$$

For the feet not currently in contact with the terrain, their last filtered contact position is used.

The feet position is filtered with a low-pass filter. The filter cut-off frequency is chosen considering the second-order system response time so that the oscillations in the underdamped case are eliminated. To this end, even a very simple low-pass filter, such as the first-order fading-memory filter, is sufficient, i.e.,

$$\hat{x}_k = \beta \hat{x}_{k-1} + (1 - \beta) x^*_k \tag{23}$$

where $\beta \in [0, 1]$ is the filter gain, $\hat{x}_k$ is the estimate at the timestep k, and x^*_k is the measurement at the timestep k.

V. Whole-Body Controller

The tracking problem is addressed by employing an approach that solves the tasks in a strictly hierarchical way, such that lower-priority tasks do not compromise higher-priority ones. Solving the locomotion problem using a hierarchical approach can yield several advantages. The main one is that it guarantees a solution to be found: with a weighted-optimization approach, it may be necessary to perform constraint relaxation to obtain a feasible solution, compromising the prioritization. On the contrary, with a hierarchical approach, higher-priority tasks are not affected by lower-priority ones.

In the following, Section V-A describes how a QP can be employed to solve a pair of equality and inequality tasks. Afterward, the sequential hierarchical QP formulation for multiple-priority tasks is introduced in Section V-B. Finally, the WBC with HO is presented by defining the optimization vector and the set of tasks, respectively, in Sections V-C and V-D.

A. Quadratic Programming for Single Task

A QP is an optimization problem with a quadratic cost function subject to linear constraints (both equality and inequality). A general QP problem has the form

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{p}^\mathsf{T}\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{C}\boldsymbol{x} \le \boldsymbol{d}. \end{aligned} \tag{24}$$

Let a task $\mathcal{T}$ be defined as a pair of equality and inequality constraints

$$\mathcal{T}: \begin{cases} \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{s}_e \\ \boldsymbol{C}\boldsymbol{x} - \boldsymbol{d} \le \boldsymbol{s}_i \end{cases} \tag{25}$$

where $\boldsymbol{s}_e$ and $\boldsymbol{s}_i$ are slack variables, of which the norm is to be minimized.

Defining $\boldsymbol{\xi} = \begin{bmatrix}\boldsymbol{x}^\mathsf{T} & \boldsymbol{s}_i^\mathsf{T}\end{bmatrix}^\mathsf{T}$, the task $\mathcal{T}$ can be reformulated as a single QP problem

$$\begin{aligned} \min_{\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{\xi}^\mathsf{T}\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{p}^\mathsf{T}\boldsymbol{\xi} \\ \text{s.t.} \quad & \widehat{\boldsymbol{C}}\boldsymbol{\xi} \le \widehat{\boldsymbol{d}}. \end{aligned} \tag{26}$$

Here, the matrices are

$$\begin{aligned} \boldsymbol{Q} &= \begin{bmatrix}\boldsymbol{A} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^\mathsf{T}\begin{bmatrix}\boldsymbol{A} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}, \quad \boldsymbol{p} = -\begin{bmatrix}\boldsymbol{A} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^\mathsf{T}\begin{bmatrix}-\boldsymbol{b}\\ \boldsymbol{0}\end{bmatrix} \\ \widehat{\boldsymbol{C}} &= \begin{bmatrix}\boldsymbol{C} & -\boldsymbol{I}\end{bmatrix}, \qquad \widehat{\boldsymbol{d}} = \boldsymbol{d}. \end{aligned} \tag{27}$$

B. Hierarchical Optimization

Let the set $\mathcal{T}_s = \{\mathcal{T}_1, \dots, \mathcal{T}_n\}$ be a list of n prioritized tasks of the form in (25) that are required to be solved. Defining the pth task pair as

$$\mathcal{T}_p: \begin{cases} \boldsymbol{A}_p\boldsymbol{x} - \boldsymbol{b}_p = \boldsymbol{s}_{e,p} \\ \boldsymbol{C}_p\boldsymbol{x} - \boldsymbol{d}_p \le \boldsymbol{s}_{i,p} \end{cases} \tag{28}$$

with $p = 1, \dots, n$, the HO approach is used, in which prioritization is ensured by projecting the initial solution $\boldsymbol{x}^*_{p+1}$ of the $(p+1)$th task into the null space of all the higher-priority equality constraints and adding the result to the solution $\overline{\boldsymbol{x}}^*_p$ of the pth task. This yields the actual solution $\overline{\boldsymbol{x}}^*_{p+1}$ of the $(p+1)$th task as

$$\overline{\boldsymbol{x}}^*_{p+1} = \overline{\boldsymbol{x}}^*_p + \overline{\boldsymbol{N}}_p\boldsymbol{x}^*_{p+1} \tag{29}$$

where $\overline{\boldsymbol{N}}_p = \text{Null}(\overline{\boldsymbol{A}}_p) = \boldsymbol{I} - \overline{\boldsymbol{A}}_p^{+}\overline{\boldsymbol{A}}_p$ is the nullspace projector of $\overline{\boldsymbol{A}}_p$ and $\overline{\boldsymbol{A}}$ is the vertical stack of the all the matrices $\boldsymbol{A}_p$ of the tasks with higher priority than $p+1$, i.e., $\overline{\boldsymbol{A}}_p = \begin{bmatrix}\boldsymbol{A}_1^\mathsf{T} & \cdots & \boldsymbol{A}_p^\mathsf{T}\end{bmatrix}^\mathsf{T}$.

It is more computationally efficient to calculate the nullspace projectors recursively, with $\overline{\boldsymbol{N}}_0 = \boldsymbol{I}$ and

$$\overline{\boldsymbol{N}}_p = \overline{\boldsymbol{N}}_{p-1}\,\text{Null}\big(\boldsymbol{A}_p\overline{\boldsymbol{N}}_{p-1}\big). \tag{30}$$

At this point, the solution to the $p+1$ prioritized task, which does not compromise the higher-priority tasks, can be found by solving a QP problem. This, similarly to (26), is written as

$$\begin{aligned} \min_{\boldsymbol{\xi}_{p+1}} \quad & \frac{1}{2}\boldsymbol{\xi}_{p+1}^\mathsf{T}\boldsymbol{Q}_{p+1}\boldsymbol{\xi}_{p+1} + \boldsymbol{p}_{p+1}^\mathsf{T}\boldsymbol{\xi}_{p+1} \\ \text{s.t.} \quad & \widehat{\boldsymbol{C}}_{p+1}\boldsymbol{\xi}_{p+1} \le \widehat{\boldsymbol{d}}_{p+1}. \end{aligned} \tag{31}$$

Here, $\boldsymbol{\xi}_{p+1} = \begin{bmatrix}\boldsymbol{x}_{p+1}^\mathsf{T} & \boldsymbol{s}_{i,p+1}^\mathsf{T}\end{bmatrix}^\mathsf{T}$, and

$$\boldsymbol{Q}_{p+1} = \begin{bmatrix}\overline{\boldsymbol{N}}_p^\mathsf{T}\boldsymbol{A}_{p+1}^\mathsf{T}\boldsymbol{A}_{p+1}\overline{\boldsymbol{N}}_p + \sigma\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix} \tag{32}$$

$$\boldsymbol{p}_{p+1} = \begin{bmatrix}\overline{\boldsymbol{N}}_p^\mathsf{T}\boldsymbol{A}_{p+1}^\mathsf{T}\left(\boldsymbol{A}_{p+1}^\mathsf{T}\overline{\boldsymbol{x}}^*_p - \boldsymbol{b}_{p+1}\right)\\ \boldsymbol{0}\end{bmatrix} \tag{33}$$

$$\widehat{\boldsymbol{C}}_{p+1} = \begin{bmatrix}\boldsymbol{C}_{p+1}\overline{\boldsymbol{N}}_p & -\boldsymbol{I}\\ \overline{\boldsymbol{C}}_p\overline{\boldsymbol{N}}_p & \boldsymbol{0}\end{bmatrix} \tag{34}$$

$$\widehat{\boldsymbol{d}}_{p+1} = \begin{bmatrix}\boldsymbol{d}_{p+1} - \boldsymbol{C}_{p+1}\overline{\boldsymbol{x}}^*_p\\ \overline{\boldsymbol{d}}_p - \overline{\boldsymbol{C}}_p\overline{\boldsymbol{x}}^*_p + \overline{\boldsymbol{s}}^*_{i,p}\end{bmatrix}. \tag{35}$$

The scalar σ is a regularization factor that guarantees that the $\boldsymbol{Q}_{p+1}$ matrix is positive definite. Similarly to $\overline{\boldsymbol{A}}_p$, the matrices $\overline{\boldsymbol{C}}_p$, $\overline{\boldsymbol{d}}_p$, and $\overline{\boldsymbol{s}}^*_{i,p}$ are the vertical stack of the corresponding matrices or vectors from the highest priority to p

$$\overline{\boldsymbol{C}}_p = \begin{bmatrix}\boldsymbol{C}_1\\ \vdots\\ \boldsymbol{C}_p\end{bmatrix}, \quad \overline{\boldsymbol{d}}_p = \begin{bmatrix}\boldsymbol{d}_1\\ \vdots\\ \boldsymbol{d}_p\end{bmatrix}, \quad \overline{\boldsymbol{s}}^*_{i,p} = \begin{bmatrix}\boldsymbol{s}^*_{i,1}\\ \vdots\\ \boldsymbol{s}^*_{i,p}\end{bmatrix}. \tag{36}$$

C. WBC Optimization Vector

To define the WBC, it is required to choose the optimization vector $\boldsymbol{x}$, the set of tasks, in the form (28), to be solved hierarchically, and their priorities.

The majority of quadrupedal robots have all their limb joints actuated. The EoM (2) can be split into the unactuated floating-base part and the actuated joints part

$$\begin{cases} \boldsymbol{M}_u(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{h}_u(\boldsymbol{q},\boldsymbol{v}) = \boldsymbol{J}_{c,u}^\mathsf{T}(\boldsymbol{q})\boldsymbol{f}_c & (37)\\ \boldsymbol{M}_a(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{h}_a(\boldsymbol{q},\boldsymbol{v}) = \boldsymbol{\tau} + \boldsymbol{J}_{c,a}^\mathsf{T}(\boldsymbol{q})\boldsymbol{f}_c. & (38) \end{cases}$$

It is chosen to optimize over the generalized accelerations $\dot{\boldsymbol{v}}$ and contact forces $\boldsymbol{f}_c$. This keeps the computational complexity limited since the optimal joint torques vector can be computed using (38), i.e.,

$$\boldsymbol{\tau}^* = \boldsymbol{M}_a(\boldsymbol{q})\dot{\boldsymbol{v}}^* + \boldsymbol{h}_a(\boldsymbol{q},\boldsymbol{v}) - \boldsymbol{J}_{c,a}^\mathsf{T}(\boldsymbol{q})\boldsymbol{f}_c^*. \tag{39}$$

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

In addition, to take into consideration the contact compliance (4) in the WBC, the desired feet–terrain penetrations $\boldsymbol{\delta}_d$ are added into the optimization vector. Here, notice that the desired penetration $\boldsymbol{\delta}_d$ differs conceptually from the actual penetration $\boldsymbol{\delta}$. The former is the desired quantity that will be obtained from the optimization together with the generalized accelerations and contact forces required to attain the requested performance, specified by the tasks. The latter, instead, is the actual foot–terrain penetration that can be measured if perfect knowledge of the terrain is available.

Thus, our optimization vector is

$$\boldsymbol{x} = \begin{bmatrix}\dot{\boldsymbol{v}}^\top & \boldsymbol{f}_c^\top & \boldsymbol{\delta}_d^\top\end{bmatrix}^\top \in \mathbb{R}^{n_v+3n_c+n_c} \tag{40}$$

where n_v is the dimension of the generalized velocity vector and n_c is the number of feet currently in contact with the terrain.

D. WBC Tasks Set

The tasks are defined by the matrices $\boldsymbol{A}_p$, $\boldsymbol{b}_p$, $\boldsymbol{C}_p$, and $\boldsymbol{d}_p$ of (28), i.e., the elements of $\mathcal{T}_s$. In case a task only requires either equality or inequality constraints, it is called a monotype. For such tasks, the absent part (equality or inequality) is represented by empty matrices that are omitted in the following description.

1) Physical Consistency: This task constrains the robot motion and the contact forces to lie on the manifold described by the unactuated floating base dynamics (37), ensuring that the solution is dynamically consistent. This leads to the equality constraint task

$$\boldsymbol{A}_1 = \begin{bmatrix}\boldsymbol{M}_u & -\boldsymbol{J}_{c,u}^\top & \boldsymbol{0}\end{bmatrix}, \qquad \boldsymbol{b}_1 = -\boldsymbol{h}_u. \tag{41}$$

2) Torque Limits: The WBC must compute joint torques within the limits of the actuators. The task is

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max}. \tag{42}$$

The joint torques vector is a function of the optimization vector, as written in (38). Substituting (38) into (42), leads to the inequality task

$$\boldsymbol{C}_{2,tl} = \begin{bmatrix}\boldsymbol{M}_a & -\boldsymbol{J}_{c,a}^\top & \boldsymbol{0}\\ -\boldsymbol{M}_a & \boldsymbol{J}_{c,a}^\top & \boldsymbol{0}\end{bmatrix}$$
$$\boldsymbol{d}_{2,tl} = \begin{bmatrix}\boldsymbol{\tau}_{\max} - \boldsymbol{h}_a\\ -\boldsymbol{\tau}_{\min} + \boldsymbol{h}_a\end{bmatrix}. \tag{43}$$

3) Friction Cone Limits: This task reduces the tangential contact forces so that the friction constraints are not violated. To limit the ratio of the tangential and normal components of the contact force, a Coulomb friction model is assumed, which yields

$$f_t \leq \mu f_n. \tag{44}$$

This describes a so-called friction cone that can be approximated by an n-sided pyramid with equal-length sides, which is linear in its parameters and thus suitable for a QP formulation. The value of n is a tradeoff between modeling accuracy and computational cost

$$\boldsymbol{C}_{2,fl} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{H}_e\cos(\theta) + \boldsymbol{L}_a\sin(\theta) - \mu\boldsymbol{N}_o & \boldsymbol{0}\end{bmatrix} \tag{45}$$
$$\boldsymbol{d}_{2,fl} = \boldsymbol{0} \tag{46}$$

where $\theta = 2k\pi/n$, $k = 0, 1, \ldots, n-1$. The matrices $\boldsymbol{H}_e$, $\boldsymbol{L}_a$, $\boldsymbol{N}_o \in \mathbb{R}^{n_c\times 3n_c}$ are the heading, lateral, and normal matrices, respectively. The ith block on the diagonal of $\boldsymbol{H}_e$ and $\boldsymbol{L}_a$ contains the projection of the heading and lateral directions on the local terrain plane in the ith contact point. Similarly, $\boldsymbol{N}_o$ is the block diagonal matrix containing the local terrain normal direction of the contact points. For example, assuming a horizontal terrain at all contact points and with ϕ the base yaw angle, the heading matrix with three feet in contact with the terrain is

$$\boldsymbol{H}_e = \begin{bmatrix}\boldsymbol{h}_e & \boldsymbol{0}_{1,3} & \boldsymbol{0}_{1,3}\\ \boldsymbol{0}_{1,3} & \boldsymbol{h}_e & \boldsymbol{0}_{1,3}\\ \boldsymbol{0}_{1,3} & \boldsymbol{0}_{1,3} & \boldsymbol{h}_e\end{bmatrix}, \quad \boldsymbol{h}_e = \begin{bmatrix}\cos\phi & -\sin\phi & 0\end{bmatrix}. \tag{47}$$

Similarly, the lateral and normal matrix would be

$$\boldsymbol{L}_a = \operatorname{diag}(\boldsymbol{l}_a, \boldsymbol{l}_a, \boldsymbol{l}_a) \quad \boldsymbol{l}_a = \begin{bmatrix}\sin\phi & \cos\phi & 0\end{bmatrix} \tag{48}$$
$$\boldsymbol{N}_o = \operatorname{diag}(\boldsymbol{n}_o, \boldsymbol{n}_o, \boldsymbol{n}_o) \quad \boldsymbol{n}_0 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}. \tag{49}$$

4) Force Modulation: The normal component of the contact forces should be restricted within a minimum and a maximum value for limiting impacts. By periodically varying these bounds throughout time, gradual lift-off and touchdowns can be ensured

$$\boldsymbol{C}_{2,fm} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{N}_o & \boldsymbol{0}\\ \boldsymbol{0} & -\boldsymbol{N}_o & \boldsymbol{0}\end{bmatrix}, \qquad \boldsymbol{d}_{2,fm} = \begin{bmatrix}\boldsymbol{F}_{n,\max}(\phi)\\ -\boldsymbol{F}_{n,\min}(\phi)\end{bmatrix}. \tag{50}$$

5) Base Linear Motion: This enforces the tracking of the desired linear trajectory of the robot base

$$\boldsymbol{A}_{3,bl} = \begin{bmatrix}\boldsymbol{J}_{b,\text{lin}} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}$$
$$\boldsymbol{b}_{3,bl} = \Big[\ddot{\boldsymbol{r}}_b^{\text{des}} + \boldsymbol{K}_{d,\text{lin}}\left(\dot{\boldsymbol{r}}_b^{\text{des}} - \boldsymbol{v}_b\right) + \boldsymbol{K}_{p,\text{lin}}\left(\boldsymbol{r}_b^{\text{des}} - \boldsymbol{r}_b\right) - \dot{\boldsymbol{J}}_{b,\text{lin}}\boldsymbol{v}\Big]. \tag{51}$$

6) Base Angular Motion: This task deals with the tracking of the angular reference trajectory of the robot base

$$\boldsymbol{A}_{3,ba} = \begin{bmatrix}\boldsymbol{J}_{b,\text{ang}} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}$$
$$\boldsymbol{b}_{3,ba} = \Big[\boldsymbol{K}_{d,\text{ang}}\,\boldsymbol{\omega}_b^{\text{des}} + \boldsymbol{K}_{p,\text{ang}}\left(\boldsymbol{q}_b^{\text{des}} \ominus \boldsymbol{q}_b\right)\Big]. \tag{52}$$

7) Swing Feet Motion: The tracking of the reference trajectory of the feet that are in swing phase is ensured by

$$\boldsymbol{A}_{3,sw} = \begin{bmatrix}\boldsymbol{J}_s & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}$$
$$\boldsymbol{b}_{3,sw} = \Big[\ddot{\boldsymbol{r}}_s^{\text{des}} + \boldsymbol{K}_{d,\text{swing}}\left(\dot{\boldsymbol{r}}_s^{\text{des}} - \boldsymbol{v}_s\right) + \boldsymbol{K}_{p,\text{swing}}\left(\boldsymbol{r}_s^{\text{des}} - \boldsymbol{r}_s\right) - \dot{\boldsymbol{J}}_s\boldsymbol{v}\Big]. \tag{53}$$

8) Soft Contact Constraint: The WBC must interact with the environment through forces that are consistent with the contact model of choice. If a rigid contact model is used, the joint accelerations should be constrained to prevent any movement of the contact point.

Here, since the monodirectional soft model (5) is used, the contact vertical kinematics is constrained by the feet deformations and, thus, the following must be satisfied:

$$-\ddot{\boldsymbol{\delta}} = \boldsymbol{J}_{c,n}\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_{c,n}\boldsymbol{v} \tag{54}$$

where $\boldsymbol{J}_{c,n} = \boldsymbol{N}_o \boldsymbol{J}_c$, $\boldsymbol{J}_{c,hl} = \begin{bmatrix} \boldsymbol{H}_e & \boldsymbol{L}_a \end{bmatrix}^T \boldsymbol{J}_c$, and similarly for $\dot{\boldsymbol{J}}_c$. This constraint together with (5) ensures consistency of the robot kinematics (54) with the KV model. Additionally, an inequality constraint binds the foot only to compressive deformations, i.e., the foot cannot be pulled by the terrain

$$\boldsymbol{A}_4 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{N}_o & -\boldsymbol{K}_p - 1/\Delta t \boldsymbol{K}_d \\ \boldsymbol{J}_{c,n} & \boldsymbol{0} & 1/\Delta t^2 \boldsymbol{I} \\ \boldsymbol{J}_{c,hl} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$
$$\boldsymbol{b}_4 = \begin{bmatrix} -1/\Delta t \boldsymbol{K}_d \boldsymbol{\delta}^*_{d,k-1} \\ -\dot{\boldsymbol{J}}_{c,n}\boldsymbol{v} + 2/\Delta t^2 \boldsymbol{\delta}^*_{d,k-1} - 1/\Delta t^2 \boldsymbol{\delta}^*_{d,k-2} \\ -\dot{\boldsymbol{J}}_{c,hl}\boldsymbol{v} \end{bmatrix}$$
$$\boldsymbol{C}_4 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}, \qquad \boldsymbol{d}_4 = \begin{bmatrix} \boldsymbol{0}_{n_c} \end{bmatrix} \tag{55}$$

where the first and second-order derivatives of the compenetration are obtained by means of backward differencing ($\boldsymbol{\delta}^*_{d,k-1}$ and $\boldsymbol{\delta}^*_{d,k-2}$ are the desired compenetration vectors at the two previous time-steps)

$$\dot{\boldsymbol{\delta}}_d = \frac{\boldsymbol{\delta}_d - \boldsymbol{\delta}^*_{d,k-1}}{\Delta t} \tag{56}$$
$$\ddot{\boldsymbol{\delta}}_d = \frac{\boldsymbol{\delta}_d - 2\boldsymbol{\delta}^*_{d,k-1} + \boldsymbol{\delta}^*_{d,k-2}}{\Delta t^2}. \tag{57}$$

For more details on its derivation, see the Appendix.

9) Energy Optimization: This task minimizes the square of the joint torques, which is roughly proportional to the energy cost of the control

$$\boldsymbol{A}_{5,\,ec} = \begin{bmatrix} \boldsymbol{M}_a & -\boldsymbol{J}^{\mathsf{T}}_{c,\,a} & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{b}_{5,\,ec} = -\boldsymbol{h}_a. \tag{58}$$

10) Contact Force Minimization: This task minimizes the square of the contact forces. Minimizing the tangential component of the contact forces is useful for avoiding unnecessary lateral forces and reducing slippage. In addition, minimizing the normal component under the higher-priority constraints results in equally distributing the contact forces among the feet in the stance phase

$$\boldsymbol{A}_{5,\,cf} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{b}_{5,\,cf} = \boldsymbol{0}. \tag{59}$$

Having defined all the necessary tasks for our WBC HO, the list of prioritized tasks is shown in Table II. The priorities are assigned based on the tasks' importance. Physical consistency has the highest priority since respecting the system dynamics is of utmost importance. The actuation limits and the friction cone limits (together with the force modulation) have the second highest priority because the robot cannot exert torques larger than its motors' capabilities and because violating the friction cone constraints leads to slippage and instability. It follows the contact constraint, with a lower priority because it represents a model and not a hard and precise limit. The reference tracking tasks have lower priorities because they can and should only be satisfied when they do not violate the previous ones. Finally, the partially redundant constraints on the energy minimization and the contact forces minimization improve the performance of the controller; however, they are not essential and need to be at the lowest-priority level.

TABLE II
PRIORITIES OF THE LOCOMOTION TASKS

Priority	Tasks
1	Physical consistency
2	Actuation torque limits Contact friction cone limits Force modulation
3	Contact constraints
4	Base linear trajectory tracking Base angular trajectory tracking Swing feet trajectory tracking
5	Energy minimization Contact forces minimization

VI. SIMULATIONS

This section describes the simulations used to test the method's effectiveness, along with a comparison with SoA techniques. Here, the acronyms RC and RP are introduced to indicate the standard LIP-based MP, the rigid motion planner (RP), and the WBC with rigid contacts, i.e., the rigid tracking controller (RC), respectively. These two controllers are based on state-of-the-art model-based controllers and implemented to be as similar as possible to the proposed approach, minus the novelties previously described. They serve as a baseline for the comparison with the proposed approach. This allows for a fair comparison among similar methods and highlights the improvements granted by considering the softness. RP is implemented based on the work in [36]. The planner employs MPC and the LIP model to generate feasible trajectories of the CoM and the swing feet. The RC is based on [34]. It consists in a controller employing QP and HO, but using the classical rigid contact hypothesis. The proposed control approach, composed of the SBIP model and the WBC with soft contacts, is divided into the soft motion planner (SP) and the soft tracking controller (SC), which are described in Sections IV and V, respectively.

After a brief description of the simulation setup and an overview of the simulations, Section VI-A describes the key performance indicators (KPIs) used to measure the performance. Section VI-B validates the proposed SC + SP on simulated soft and rigid terrains. Section VI-C uses the previously defined KPI to compare the proposed approach with the SoA methods. In particular, the four combinations (RC + RP, SC + RP, RC + SP, and SC + SP) are compared against each other in order to assess how each component affects the performance. Finally, in Section VI-D, the proposed SC + SP approach is again compared to the RC + RP method by assessing their ability to avoid losing balance while increasing locomotion velocities.

The proposed controller was implemented and validated with simulations using a laptop with an Intel Core i7–8565U processor. The control ran in a Docker container with ROS 2 Humble and Gazebo Classic 11. ODE, the default physics engine, was used, following the results in [37]. The MP and the WBC were implemented in C++ as ROS 2 controllers, with the MPC block running at a frequency of 100 Hz and the other blocks at 200 Hz (in Fig. 2 f_{MPC} and f_{WBC}, respectively). The robot kinematics and dynamics were computed using the Pinocchio library [38], [39]. The QP problems were solved with a modified version of the quadprog library [40].

A. Key Performance Indexes

The heading and lateral root mean square error (RMSE) are

$$h_{\mathrm{RMSE}} = \sqrt{\frac{\sum_{i=1}^{N}\left(v_{h,i} - v_{h,i}^{\mathrm{des}}\right)^2}{N}} \tag{60}$$

$$l_{\mathrm{RMSE}} = \sqrt{\frac{\sum_{i=1}^{N}\left(v_{l,i} - v_{l,i}^{\mathrm{des}}\right)^2}{N}} \tag{61}$$

where $v_{h,i}$ and $v_{h,i}$ are the measured and commanded base velocity in the heading direction at the ith instant, respectively, and similarly for the lateral direction; while N is the total number of measurements.

The slippage along a trajectory is computed as the integral of the feet' horizontal movements while in contact with the terrain

$$\mathrm{slip} = \sum_{f=1}^{4}\sum_{i=1}^{N} c_f \|v_{f,i}\| \Delta t \tag{62}$$

where c_f is equal to 1 if the fth foot is in contact with the terrain and 0 otherwise; the velocity of the fth foot at the ith instant is $v_{f,i}$.

The cost of transport (CoT) is computed by assuming that most of the energy is lost due to thermal effects and not due to the mechanical work of the actuators. It is computed with

$$\mathrm{CoT} = \frac{\sum_{i=1}^{n} \boldsymbol{v}_{\mathrm{hor},i} \cdot \boldsymbol{v}_i^{\mathrm{des}} / \|\boldsymbol{v}_i^{\mathrm{des}}\|}{\sum_{i=1}^{n} \boldsymbol{\tau}_i^2} \tag{63}$$

where $\boldsymbol{v}_{\mathrm{hor},i}$ and $\boldsymbol{v}_i^{\mathrm{des}}$ are, respectively, the horizontal velocity and horizontal reference velocity at the ith instant.

B. Controller Behavior

1) Simulation Description: The goal of these simulations is to compare the performance of the proposed SC + SP framework with the traditional RC + RP approach. This is done by testing the locomotion behavior over rigid and soft terrains.

The simulations have been performed on two robotic platforms: 1) ANYmal C [11] and 2) SOLO12 [41]. Two terrain types have been considered: 1) a rigid terrain ($k_p = 10^6$ N/m, $k_p = 100$ Ns/m) and 2) a soft terrain. Since the weights of the two tested robots are different, different soft terrain parameters are used to achieve analogous conditions. For ANYmal's tests, the terrain is ($k_p = 2000$ N/m, $k_d = 100$ Ns/m), while it is ($k_p = 400$ N/m, $k_d = 15$ Ns/m) for SOLO's ones. The simulations compare SC + SP with the SoA, represented by RC + RP.

The task consists of following a linear velocity reference command equal to $\boldsymbol{v}_{\mathrm{ref}} = \begin{bmatrix}0.15 & 0\end{bmatrix}^{\mathsf{T}}$ m/s for SOLO12. The KPI are computed over a time duration equal to 10 s; for each configuration, three runs are performed and averaged.

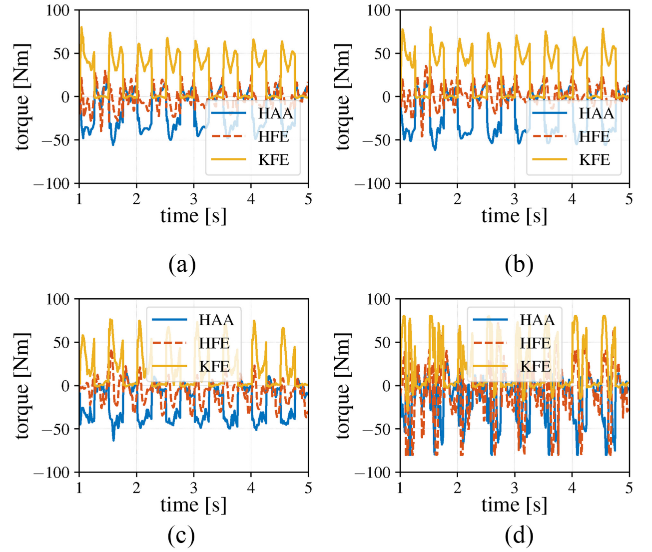


Fig. 4. Commanded left-front leg joint torques during a simulation of the ANYmal C robot performing a walking trot. HAA, HFE, and KFE stands for hip adduction–abduction, hip flexion–extension, and knee FE. (a) RC + RP with rigid terrain. (b) SC + SP with rigid terrain. (c) RC + RP with soft terrain. (d) SC + SP with soft terrain.

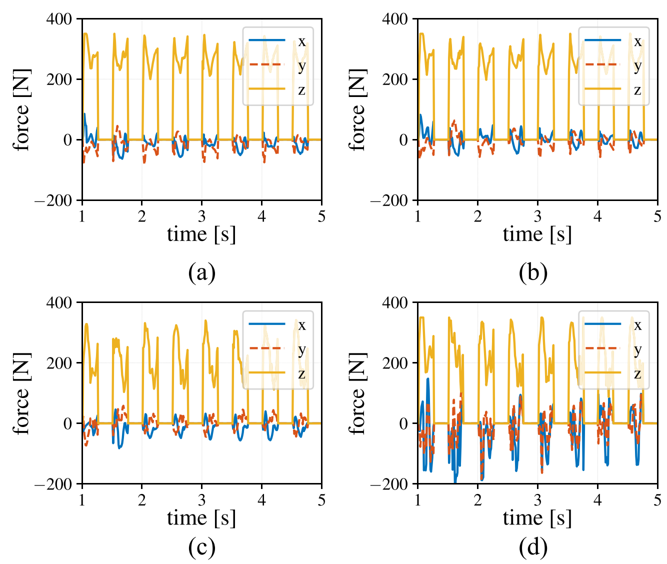
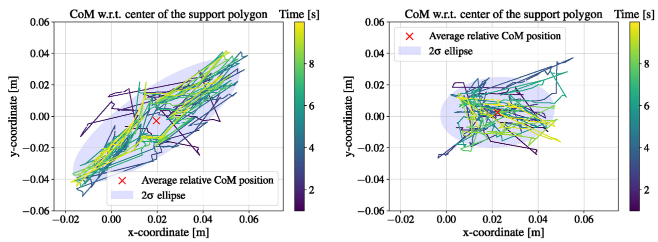


Fig. 5. Commanded left-front leg contact forces during a simulation of the ANYmal C robot performing a walking trot. (a) RC + RP with rigid terrain. (b) SC + SP with rigid terrain. (c) RC + RP with soft terrain. (d) SC + SP with soft terrain.

2) Results and Discussion: Fig. 4 collects the joint torques of ANYmal C locomotion in four different conditions. Fig. 4(a) and (c) show the joint torques employing the RC + RP controller over rigid and soft terrains, respectively. Fig. 4(b) and (d) show the joint torques employing the SC + SP controller over rigid and soft terrains, respectively. Fig. 5 shows the contact forces in the same four cases. Similar behaviors are obtained in the SOLO 12 simulations, but they are not reported here for the sake of space.

In Figs. 4(a) and (b), and 5(a) and (b), it can be observed that, in the rigid terrain case, both the joint torques and the contact forces do not vary significantly between the two



(a) (b)

Fig. 6. Commanded left-front leg contact forces during a simulation of the ANYmal C robot performing a walking trot. (a) RC + RP with soft terrain. (b) SC + SP with soft terrain.

TABLE III
KPI RELATIVE CHANGE WITH RESPECT TO THE RC + RP

Terrain – Robot	h_{RMSE}	l_{RMSE}	slip	CoT
Rigid – ANYmal C	+3%	+4%	+0.1%	+1%
Soft – ANYmal C	−12%	−36%	+3.6%	−10%
Rigid – SOLO12	+0.9%	+0.6%	−0.4%	+1%
Soft – SOLO12	−10%	−50%	+1%	−58%

control approaches. Conversely, when locomoting on soft terrains, more significant changes can be appreciated. When using the SC + SP approach, the joint torques vary more abruptly and the contact forces are higher. Indeed, due to the characteristics of the proposed approach, the robot is forced to respond to the terrain softness by adapting its desired trajectory and contact forces. This causes the robot to exert more effort but, as explained in Section VI-C, results in an overall performance increment.

Table III summarizes the results of all the tests. In particular, it reports the percentage change of the KPI with the SC + SP approach w.r.t. the RC + RP one, i.e., $(\text{KPI}_{\text{SC SP}} - \text{KPI}_{\text{RC RP}})/(\text{KPI}_{\text{RC RP}})$. The analysis of these KPI gives insights into our approach's performance in two different cases. First of all, adopting the SBIP-based MP and the soft WBC does not produce significant changes in the four KPI of both ANYmal C and SOLO12 while locomoting on the rigid terrain. The KPI changes are limited to some percentage points or even less. On the other hand, the SC + SP approach reduces the heading RMSE by about 10% and the lateral RMSE by an even larger margin. The slippage is almost unchanged, while the CoT is significantly reduced. As discussed in Section VI-C in more detail, this results in an overall performance improvement.

The walking trot is a dynamically stable gait; it requires and exploits statically unstable configurations to achieve higher velocities and efficiency. Nonetheless, the base displacement w.r.t. the support polygon gives insights into the robot's stability during the locomotion. Fig. 6 captures the movement of the robot base w.r.t. the geometrical center of the support polygon of the two approaches. From Fig. 6 it can be observed that the proposed approach does not reduce stability, having an average CoM displacement similar to the SoA approach and a comparable variance. These results were consistent across multiple runs, not reported here for brevity. It is worth noting that the proposed approach achieves better performance and velocity tracking (see Table III), a benefit obtained without hindering stability.

Fig. 7 represents two sequences of ANYmal C and SOLO12 trotting on soft grounds. For more details, please refer to the supplementary material. From them, it can be observed that the contact feet significantly penetrate the terrain: the proposed approach adapts by changing the desired feet trajectory and achieves stable locomotion.

C. Performance Comparison With SoA

1) Simulation Description: Assessing the performance of the proposed approach is particularly important, especially in the case of very challenging soft terrains. As such, this section provides a quantitative comparison between the proposed approach and the SoA. The KPI are obtained by commanding a horizontal linear velocity equal to $\boldsymbol{v}_{\text{ref}} = \begin{bmatrix}0.2 & 0\end{bmatrix}^{\mathsf{T}}$m/s and a zero yaw rate for a time duration equal to 10 s.

The four controllers' configurations (i.e., RC + RP, SC + RP, RC + SP, and SC + SP) are compared on two different soft terrains: 1) ($k_p = 1750$ N/m, $k_p = 75$ Ns/m) and 2) ($k_p = 1750$ N/m, $k_p = 150$ Ns/m). The first terrain has a considerably lower damping and is much more challenging. Three runs are performed for each configuration and terrain.

2) Results and Discussion: Fig. 8 summarizes the results of all 24 simulations. In particular, it shows the KPI mean and standard deviation among the three trials in the eight cases.

The heading and lateral RMSE values display that the SC improves the tracking of the velocity command on soft terrains, both with the rigid and soft planners. Similarly, the SP improves the velocity tracking compared to the RP with both TCs. Moreover, it can be observed that the improvement is more significant with softer terrains.

The total slippage increases when using the SP compared to the RP. However, since the SP guarantees better-velocity tracking, the slippage normalized by the distance traveled is reduced. The SC slightly reduces the measured slippage. In both cases the influence on slippage is limited.

The CoT highlights that the SC improves the locomotion efficiency by a considerable amount. On the other hand, the SC has a much lower-observed impact on the CoT.

Finally, these results are consistent with the ones obtained in simulations and reported in Table III.

D. Fall Performance Comparison With SoA

1) Simulation Description: The ability to maintain stability in complex scenarios is even more fundamental than locomoting rapidly and efficiently. As such, this simulations' goal is to evaluate the performance of the controller in this regard.

The stability performance of the robot is evaluated by performing tests on soft terrains with a varying commanded velocity that gradually increases. The robot starts with a null velocity reference, and every 10 s the forward velocity is increased by 0.1 m/s. The test stops when the robot falls, i.e., the robot's base comes in contact with the terrain. The falling speed is the speed at which the robot loses stability.

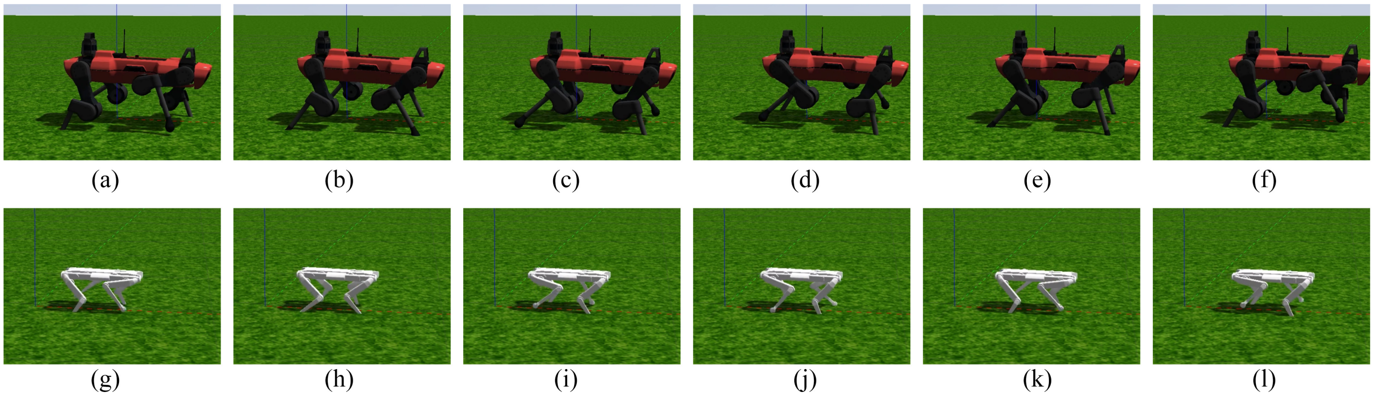


Fig. 7. Sequences of ANYmal C (a)–(f) and SOLO12 (g)–(l) performing the walking trot on two soft grounds ($k_p = 2000$ N/m, $k_d = 100$ Ns/m and $k_p = 400$ N/m, $k_d = 15$ Ns/m). For more details, please refer to the attached video.

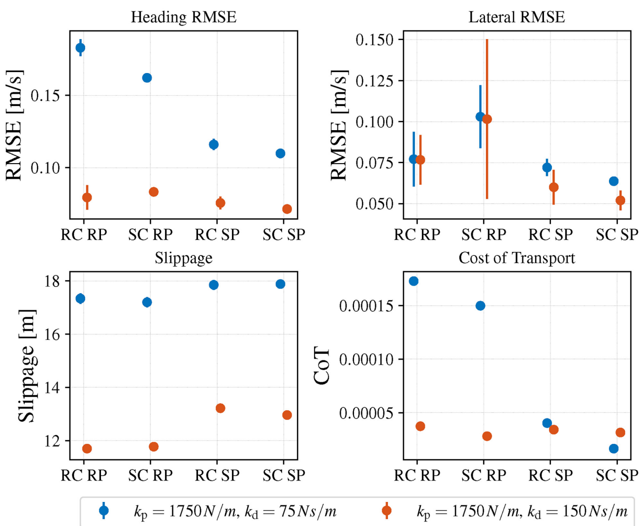


Fig. 8. KPI means and standard deviations over three simulations of different planner-controller configurations on different terrains.

TABLE IV
MAXIMUM COMMANDED VELOCITY BEFORE FALLING

Terrain	SC SP	RC RP
$k_p = 2000\,\mathrm{N/m}$, $k_d = 100\,\mathrm{Ns/m}$	$0.9\,\mathrm{m/s}$	$0.5\,\mathrm{m/s}$
$k_p = 1500\,\mathrm{N/m}$, $k_d = 75\,\mathrm{Ns/m}$	$0.6\,\mathrm{m/s}$	$0.4\,\mathrm{m/s}$

The tests are performed with ANYmal C locomoting over two particularly challenging soft terrains, which are ($k_p = 2000$ N/m, $k_d = 100$ Ns/m) and ($k_p = 1500$ N/m, $k_d = 75$ Ns/m) respectively. The comparison is between the SC + SP and RC + RP configurations.

2) Results and Discussion: Table IV reports the results of the test, showing the maximum commanded velocity before falling in each of the four cases.

These results demonstrate that the proposed approach achieves more stable motions over very soft terrains, enabling the robot to maintain stability at higher speeds before failing.

VII. EXPERIMENTS

This section details the experimental validation of the proposed control strategy, deployed on the SOLO12 [41] quadrupedal robot. The goal of the experiments was to validate the approach outside of a simulated environment and with compliant elements not perfectly described by a linear contact model. In addition, they aimed to confirm the performance improvements granted by the control strategy in comparison to the state-of-the-art. The experiments compared the RP + RC and the SP + SC configurations, i.e., the state-of-the-art and the proposed controller, respectively, during locomotion on compliant terrains.

Due to the additional complexity introduced by soft grounds, the robot's pose data was obtained using the Qualisys Arqus[1] motion capture system, without relying on proprioceptive state estimation. Eight cameras tracked the position of SOLO12's base, using the default attachment with 13 markers, with an estimated accuracy of less than 1 mm.

The experimental setup was composed of foam rubber sheets with a thickness of 10 cm and a density of 30 kg/m^3. The foam's stiffness was measured to be roughly 1000 N/m. Due to SOLO12's low weight, layering multiple foams did not significantly affect the ground stiffness.

1) Experiment Description: The two controllers are compared on three different inclinations, each experiment repeated three times, for a total of 18 experiments. The results of the three run were averaged to ensure the reliability of the data. Similarly to Section VI-C, the robot is commanded to perform a walking trot at a constant forward speed equal to 0.1 m/s for a time duration of 10 s. To assess the robustness of the approach and its behavior in different conditions, this was performed with the ground inclined by 0°, 5°, and 10°.

2) Results and Discussion: Fig. 9 shows some snapshots of the robot performing the walking gait on the soft ground with 0° and 10° of inclination. It can be observed that the proposed approach demonstrates a stable gait even in the presence of significant terrain compliance and challenging inclinations, showcasing its robustness and effectiveness.

[1]More information available at https://www.qualisys.com/cameras/arqus/.

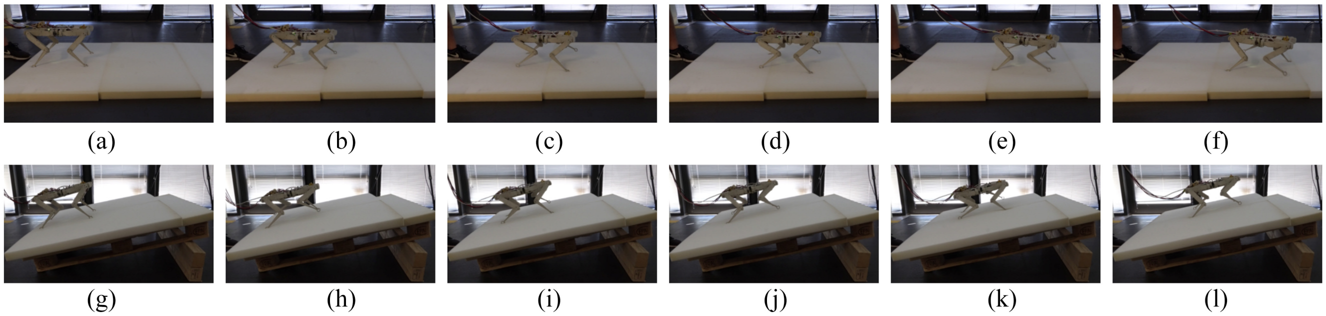


Fig. 9. Sequences of SOLO12 performing the walking trot on soft ground inclined by 0° (a)–(f) and 10° (g)–(l), respectively. For more details, please refer to the attached video.

TABLE V
EXPERIMENTS KPI RELATIVE CHANGE WITH RESPECT TO THE RC + RP

Inclination	$\mathrm{h_{RMSE}}$	$\mathrm{l_{RMSE}}$	CoT
0°	−69%	−80%	−27%
5°	−29%	−58%	−25%
10°	−13%	+3%	−10%

Furthermore, Table V reports the relative change of the KPIs of the proposed control approach with respect to the RP + RC configuration. More specifically, the heading (60) and lateral RMSE (61) and the CoT (63) presented in Section VI-A are reported. The proposed approach outperforms the state-of-the-art in all cases, both on flat terrains and on inclined terrains. Moreover, the obtained results are coherent with the ones obtained in the simulation. In addition to validating the proposed control strategy, this also confirms the reliability of simulation data to evaluate locomotion on soft grounds, despite employing a very simple soft-contact model.

VIII. CONCLUSION

This work extends the state-of-the-art quadrupedal locomotion control to account for contact compliance. It adopts a two-control-blocks approach and considers it in its entirety, integrating the compliance into both blocks for the first time. The SBIP model is proposed and employed in an MPC-based MP. Furthermore, a Hierarchical-Optimization-based WBC that accounts for the terrain-feet contact softness is presented. The control approach was validated with simulations on various robotic platforms over rigid and compliant terrains. A quantitative analysis of the performance improvements achieved by incorporating the compliance in the MP and the TC, both individually and in combination, was conducted. Finally, the proposed control architecture was experimentally validated on a compliant terrain with the SOLO12 robot. The results show that the proposed controller achieves better performance than state-of-the-art model-based solutions on terrains whose compliance is considerable.

Future works will focus on and improve the different blocks of the proposed control architecture. The MP will be extended to take into account a varying pendulum height, the feet's inertia, or the rotational effects. The TC will be adapted by considering different soft contact models, and the terrain parameters can be measured or estimated online. Finally, the state estimator module, which was not considered in this work, will be extended to consider soft contacts.

APPENDIX

The monodirectional spring damper model, used for all the legs in contact with the terrain, is described by the following equations (obtained from (54) for the deformation and (5) for the forces and nonslippage constraint):

$$\begin{aligned} -\ddot{\boldsymbol{\delta}}_d &= \boldsymbol{J}_{c,n}\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_{c,n}\boldsymbol{v} \\ \boldsymbol{N}_a\boldsymbol{f}_c &= \boldsymbol{K}_p\boldsymbol{\delta}_d + \boldsymbol{K}_d\dot{\boldsymbol{\delta}}_{,d} \\ \boldsymbol{H}_e\ddot{\boldsymbol{p}}_c &= \boldsymbol{H}_e\big(\boldsymbol{J}_c\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_c\boldsymbol{v}\big) = 0 \\ \boldsymbol{L}_a\ddot{\boldsymbol{p}}_c &= \boldsymbol{L}_a\big(\boldsymbol{J}_c\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_c\boldsymbol{v}\big) = 0. \end{aligned} \tag{64}$$

By numerically differentiating the deformations to obtain the deformation velocity (56) and acceleration (57) and substituting those in (64), the following equations are obtained:

$$\begin{aligned} -\frac{\boldsymbol{\delta}_d - 2\boldsymbol{\delta}^*_{d,k-1} + \boldsymbol{\delta}^*_{d,k-2}}{\Delta t^2} &= \boldsymbol{J}_{c,n}\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_{c,n}\boldsymbol{v} \\ \boldsymbol{N}_a\boldsymbol{f}_c &= \boldsymbol{K}_p\boldsymbol{\delta}_d + \boldsymbol{K}_d\frac{\boldsymbol{\delta}_d - \boldsymbol{\delta}^*_{d,k-1}}{\Delta t} \\ \boldsymbol{H}_e\ddot{\boldsymbol{p}}_c &= \boldsymbol{H}_e\big(\boldsymbol{J}_c\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_c\boldsymbol{v}\big) = 0 \\ \boldsymbol{L}_a\ddot{\boldsymbol{p}}_c &= \boldsymbol{L}_a\big(\boldsymbol{J}_c\dot{\boldsymbol{v}} + \dot{\boldsymbol{J}}_c\boldsymbol{v}\big) = 0. \end{aligned} \tag{65}$$

Conversely, (65) is put in matrix form to get $\boldsymbol{A}_4$ and $\boldsymbol{b}_4$ in (55). Finally, matrix $\boldsymbol{C}_4$ and $\boldsymbol{d}_4$ in (55) are obtained by putting in matrix form $\boldsymbol{\delta}_d \leq \boldsymbol{0}$.

REFERENCES

[1] T. A. McMahon, "The role of compliance in mammalian running gaits," *J. Exp. Biol.*, vol. 115, no. 1, pp. 263–282, 1985.

[2] T. A. McMahon and G. C. Cheng, "The mechanics of running: How does stiffness couple with speed?" *J. Biomech.*, vol. 23, no. 1, pp. 65–78, 1990.

[3] R. M. Alexander, "Elastic mechanisms in primate locomotion," *Z. Morphol. Anthropol.*, vol. 78, no. 3, pp. 315–320, 1991.

[4] M. H. Dickinson, C. T. Farley, R. J. Full, M. Koehl, R. Kram, and S. Lehman, "How animals move: An integrative view," *Science*, vol. 288, no. 5463, pp. 100–106, 2000.

[5] E. Schumann, N. Smit-Anseeuw, P. Zaytsev, R. Gleason, K. A. Shorter, and C. D. Remy, "Effects of foot stiffness and damping on walking robot performance," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 3698–3704.

[6] Z. Li, Q. Ge, W. Ye, and P. Yuan, "Dynamic balance optimization and control of quadruped robot systems with flexible joints," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 10, pp. 1338–1351, Oct. 2016.

[7] C. Semini, N. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. Caldwell, "Design of HyQ-A hydraulically and electrically actuated quadruped robot," *Proc. Inst. Mech. Eng. Part-I, J. Syst. Control Eng.*, vol. 225, pp. 831–849, Aug. 2011.

[8] C. Semini et al., "Design of the hydraulically actuated, torque-controlled quadruped robot HyQ2Max," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 635–646, Apr. 2017.
[9] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for highly efficient quadrupeds and implementation on the MIT cheetah robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3307–3312.
[10] G. A. Folkertsma, S. Kim, and S. Stramigioli, "Parallel stiffness in a bounding quadruped with flexible spine," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 2210–2215.
[11] M. Hutter et al., "ANYmal-A highly mobile and dynamic quadrupedal robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 38–44.
[12] C. Piazza et al., "Toward an adaptive foot for natural walking," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, 2016, pp. 1204–1210.
[13] M. G. Catalano et al., "HRP-4 walks on soft feet," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 470–477, Apr. 2021.
[14] M. G. Catalano et al., "Adaptive feet for quadrupedal walkers," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 302–316, Feb. 2022.
[15] S. Hauser, P. Eckert, A. Tuleu, and A. Ijspeert, "Friction and damping of a compliant foot based on granular jamming for legged robots," in *Proc. 6th IEEE Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, 2016, pp. 1160–1165.
[16] A. Ranjan, F. Angelini, T. Nanayakkara, and M. Garabini, "Design guidelines for bioinspired adaptive foot for stable interaction with the environment," *IEEE/ASME Trans. Mechatronics*, vol. 29, no. 2, pp. 843–855, Apr. 2024.
[17] F. Angelini et al., "Robotic monitoring of habitats: The natural intelligence approach," *IEEE Access*, vol. 11, pp. 72575–72591, 2023.
[18] S. Godon, C. Prados, A. Chemori, A. Ristolainen, and M. Kruusmaa, "Walking in mud: Modelling, control and experiments of quadruped locomotion," TechRxiv, Preprint, Jul. 2024. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.172047167.78902329/v1
[19] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, 2016, pp. 558–564.
[20] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, "STANCE: Locomotion adaptation over soft terrain," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 443–457, Apr. 2020.
[21] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expand. Soc. Role Robot. Next Millennium*, 2001, pp. 239–246.
[22] O. Urbann, I. Schwarz, and M. Hofmann, "Flexible linear inverted pendulum model for cost-effective biped robots," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots (Humanoids)*, 2015, pp. 128–131.
[23] L. Lanari, O. Urbann, S. Hutchinson, and I. Schwarz, "Boundedness approach to gait planning for the flexible linear inverted pendulum model," in *Proc. 20th RoboCup Robot World Cup*, 2017, pp. 58–70.
[24] M. Shahbazi, R. Babuška, and G. A. Lopes, "Unified modeling and control of walking and running on the spring-loaded inverted pendulum," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1178–1195, Oct. 2016.
[25] D. Kim, S. J. Jorgensen, J. Lee, J. Ahn, J. Luo, and L. Sentis, "Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 936–956, 2020.
[26] V. Vasilopoulos, I. S. Paraskevas, and E. G. Papadopoulos, "Control and energy considerations for a hopping monopod on rough compliant terrains," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 4570–4575.
[27] V. Vasilopoulos, K. Machairas, and E. G. Papadopoulos, "Quadruped pronking on compliant terrains using a reaction wheel," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 3590–3595.
[28] A. H. Chang, C. M. Hubicki, J. J. Aguilar, D. I. Goldman, A. D. Ames, and P. A. Vela, "Learning to jump in granular media: Unifying optimal control synthesis with Gaussian process-based regression," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 2154–2160.
[29] M. Azad and M. N. Mistry, "Balance control strategy for legged robots with compliant contacts," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 4391–4396.
[30] T. Flayols, A. Del Prete, M. Khadiv, N. Mansard, and L. Righetti, "Reactive balance control for legged robots under visco-elastic contacts," *Appl. Sci.*, vol. 11, no. 1, p. 353, 2020.
[31] M. Neunert et al., "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1458–1465, Jul. 2018.
[32] S. Fahmi, C. Mastalli, M. Focchi, and C. Semini, "Passive whole-body control for quadruped robots: Experimental validation over challenging terrain," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2553–2560, Jul. 2019.
[33] S. Choi et al., "Learning quadrupedal locomotion on deformable terrain," *Sci. Robot.*, vol. 8, no. 74, 2023, Art. no. eade2256.
[34] C. D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2017, pp. 3359–3365.
[35] M. Khadiv, S. A. A. Moosavian, A. Yousefi-Koma, M. Sadedel, A. Ehsani-Seresht, and S. Mansouri, "Rigid vs compliant contact: An experimental study on biped walking," *Multibody Syst. Dyn.*, vol. 45, no. 4, pp. 379–401, Apr. 2019.
[36] G. Xin et al., "Robust footstep planning and LQR control for dynamic quadrupedal locomotion," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4488–4495, Jul. 2021.
[37] D. Kang and J. Hwangho. "SimBenchmark." Accessed: Nov. 28, 2024. [Online]. Available: https://leggedrobotics.github.io/SimBenchmark/
[38] J. Carpentier et al. "Pinocchio: Fast forward and inverse dynamics for poly-articulated systems." Accessed: Nov. 28, 2024. [Online]. Available: https://stack-of-tasks.github.io/pinocchio
[39] J. Carpentier et al., "The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *Proc. IEEE Int. Symp. Syst. Integr. (SII)*, 2019, pp. 614–619.
[40] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Math. Program.*, vol. 27, no. 1, pp. 1–33, Sep. 1983.
[41] F. Grimminger et al., "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3650–3657, Apr. 2020.

Davide De Benedittis received the B.S. degree in mechanical engineering and the M.S. degree (cum laude) in automation and robotics engineering from the University of Pisa, Pisa, Italy, in 2018 and 2022, respectively, where he is currently pursuing the Ph.D. degree in robotics engineering.

His main research interests are the control of legged robots, multirobots control, and optimization techniques.

Franco Angelini (Member, IEEE) received the B.S. degree in computer engineering, the M.S. degree (cum laude) in automation and robotics engineering, and the Ph.D. degree (cum laude) in robotics from the University of Pisa, Pisa, Italy, in 2013, 2016, and 2020, respectively.

He is currently an Assistant Professor with University of Pisa. His main research interests are control of soft robotic systems, grasping, and impedance planning.

Manolo Garabini (Member, IEEE) received the B.S. degree in mechanical engineering, the M.S. degree in mechanical engineering, and the Ph.D. degree in robotics from the University of Pisa, Pisa, Italy, in 2007, 2010, and 2014, respectively.

He is currently employed as an Assistant Professor with the University of Pisa. He contributed to the realization of modular variable stiffness actuators, to the design of the humanoid robot WALK-MAN and, to the development of efficient and effective compliance planning algorithms for interaction under uncertainties. He is currently the Principal Investigator in the THING H2020 EU Research Project for the University of Pisa, and the Coordinator of the Dysturbance H2020 Eurobench Subproject. Finally, he is the Coordinator of the H2020 EU Research Project Natural Intelligence. His main research interests include the design, planning, and control of soft adaptive robots.

Open Access funding provided by 'Università di Pisa' within the CRUI CARE Agreement