Robust Bipedal Walking With Closed-Loop MPC: Adios Stabilizers

Antonin Dallard, Mehdi Benallegue, Nicola Scianca, Fumio Kanehiro, and Abderrahmane Kheddar, *Fellow, IEEE*

***Abstract*—In this article, we propose a novel walking control scheme based on the dynamics of the linear inverted pendulum (LIP) model. Pattern generation incorporates a model of contact forces, enabling closed-loop control of the humanoid robot's state, including the center-of-mass position, velocity, and zero moment point. No additional control policies are required to maintain static and dynamic balance. Our approach also includes dynamic replanning of step locations and timings, thus preserving the LIP's boundedness condition. We validated this controller on five different humanoid robots, testing its robustness through various disturbances, including sudden pushes during walking and static phases. In addition, our controller demonstrated effective locomotion over uneven and compliant terrain. Both simulation and experimental results confirm the effectiveness and robustness of this controller.**

***Index Terms*—Force control, humanoid and bipedal locomotion, humanoid robots, legged robots.**

I. Introduction

WALKING control strategies for biped and humanoid robots are diverse, as discussed in Section II. The complexity of this problem emerges from the interplay of several key components, including: 1) reactive footstep planning; 2) whole-body switched control under balance constraints; 3) stability (as defined in control theory) of the overall system; and 4) perception for understanding the surrounding environment and the robot's state.

This article focuses on a particular class of walking control strategies: the use of the linear inverted pendulum model (LIPM), first introduced in [1] and later popularized by the Honda humanoid robots (see [2], [3], [4], and [5]). The LIPM is still widely used in many humanoid applications, though it has evolved in various ways (see Section II).

Received 14 January 2025; revised 31 May 2025; accepted 17 June 2025. Date of publication 10 July 2025; date of current version 7 August 2025. This article was recommended for publication by Associate Editor Y. Zhao and Editor K. Mombaur upon evaluation of the reviewers' comments. *(Corresponding author: Antonin Dallard.)*

Antonin Dallard is with the Wandercraft, 75004 Paris, France (e-mail: antonin.dallard@gadz.org).

Mehdi Benallegue and Fumio Kanehiro are with the CNRS-AIST Joint Robotics Laboratory, IRL3218, Tsukuba, Japan.

Nicola Scianca is with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza University of Rome, 00185 Rome, Italy.

Abderrahmane Kheddar is with the CNRS-AIST Joint Robotics Laboratory, IRL3218, Tsukuba, Japan, and also with the CNRS-University of Montpellier, LIRMM, UMR5506, Montpellier, France.

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TRO.2025.3588452

The standard implementation of LIPM-based models focuses on the dynamics of the center of mass (CoM) to facilitate fast planning and control. Typically, control relies on the zero moment point (ZMP) to maintain dynamic balance. The ZMP represents a constraint on contact forces applied to the robot, remaining within the support polygon during motion. In linear systems, the ZMP can either be treated as the causality output (in the cart table model [6]) or as the input (in the LIPM), linking it to the CoM. However, instability inherent in this model means that, for arbitrary ZMP trajectories, the corresponding CoM trajectory can diverge.

The intrinsically stable model preview control (IS-MPC) [7] is a walking gait generator derived from the LIPM that explicitly incorporates stability constraints. These constraints ensure recursive feasibility, meaning that it is always possible to find a solution that satisfies the balance requirements, keeping the CoM trajectory bounded with respect to the ZMP. The IS-MPC is capable of updating footstep locations and has been extended to adapt the duration of steps as well [8].

However, reduced models such as LIPM cannot fully capture the complexity of the robot's multibody dynamics. As a result, discrepancies are expected between the behavior predicted by these models and the actual behavior of the robot. The inverted pendulum model, derived from centroidal dynamics, essentially controls the contact forces applied to the robot. Different ground conditions, such as uneven or compliant surfaces, can lead to different behaviors even when the same controller is used (see Section II). To address such modeling discrepancies, a correction policy is often introduced, which compensates for the differences between the model and the real robot state, often including a model of the contact forces, as in [9]. This correction, often referred to as the *stabilizer* in humanoid robotics, typically comes with several tunable parameters. Tuning these parameters is a nontrivial task, especially when combined with the controller's parameters. Moreover, this correction policy is inherently short-sighted compared to model-predictive control (MPC), which relies on state prediction, potentially undermining some of the advantages of prediction, as discussed in Section III-A.

One might argue that MPC should address these issues due to its inherent robustness. However, as with any controller, MPC's robustness stems from its closed-loop nature, meaning that the predicted trajectory is recalculated at each time step using the measured state. If the contact force model is not considered,

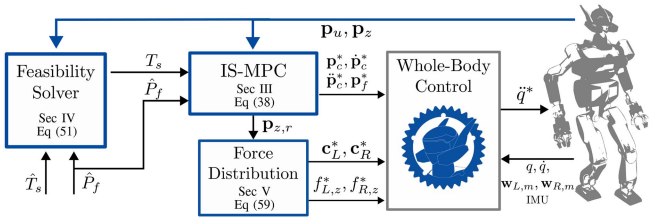

Fig. 1. Global control scheme. The superscript "$*$" represents the control references for the whole body control or the robot. The swing foot trajectory planning or the force control scheme is included in the whole-body control box as they are outside the article's contribution. The blue elements highlight our adding w.r.t existing approaches.

the walking gait may still be infeasible, forcing reliance on an external *stabilizer*.

This article addresses this issue by presenting a humanoid walking control scheme offering the following key contributions:

1) a reformulation of the IS-MPC that includes a model of the contact forces, effectively closing the loop on the robot's state (CoM, CoM velocity, and ZMP). This approach eliminates the need for an external correction policy of the MPC references (the so-called *stabilizer*) to enforce balance (see Section III). This is the main contribution of this article;
2) an LIPM feasibility solver, based on an extended stability constraint, to adapt footstep locations and timings over the entire planning horizon using the robot's current state (see Section IV);
3) a force distribution scheme to plan contact forces during double-support phases (see Section V);
4) extensive experimental validation on five different humanoid robot (as in Fig. 19), all controlled with the same software, to assess the robustness, versatility, and portability of our algorithms (see Sections VI and VII).

Fig. 1 highlights our contributions, the order of execution, sections, and the main equations for each subproblem.

II. Background

Bipedal and humanoid locomotion is an active area of research within robotics. For comprehensive reviews of the topic, we recommend the recent works in [10], [11], [12], and [13], and the monographs in [6] and [14].

Our work targets motorized humanoid robots rather than passive bipedal locomotion [15]. Specifically, our interest lies in robots capable of dynamic noncyclic locomotion. While research on passive walkers offers valuable insights into motorized gait energy optimization, it falls outside the scope of this article. We also exclude quadrupedal and hexapod locomotion studies. Although there are similarities with bipedal locomotion [16], [17], multilegged systems exhibit more complex gait patterns and specificities (e.g., balance and mass distribution), which differ from those in humanoid robots. In the field of computer graphics and animation, notable works exist that simulate human-like bipedal walking based on human-captured data [18], [19], [20], [21]. However, transferring these techniques to humanoid robots is challenging, as they do not account for sensor noise, uncertainties, and physical limitations.

Various methods for walking gait generation based on full dynamic and kinematic constraints have been proposed. For instance, semi-infinite optimization [22], differential dynamic programming [23], hybrid zero dynamics [12], and learning-based approaches [24], [25], [26] are among the prominent strategies. However, other approaches focus primarily on centroidal gait generation, which involves computing a feasible trajectory for the CoM.

This approach traces its origins to the LIPM [27], where the CoM dynamics are constrained both vertically and in terms of angular momentum. Over time, control strategies for centroidal dynamics have evolved to incorporate variations in vertical height [28], [29] and angular momentum around the CoM [30], [31]. Although full centroidal dynamics can be captured [32], [33], nonlinear solvers are required to account for all constraints simultaneously. The LIPM remains a useful model because of its linearity and simplicity, allowing for easier analysis of stability and its application to real humanoid robots.

Our work builds upon the LIPM, specifically leveraging the contributions in [7]. To maintain consistency, we adopt the same notation and conventions from the original work.

1) Superscripts $x^{\square}$ refer to indices in a sequence, unless there is no confusion with powers of x.
2) The superscript $\square^*$ refers to all current references applied to the robot.
3) Lower-script $x_{\square}$ indicates the value type (e.g., ZMP z, CoM c, DCM u, foot f, etc.).
4) x, y, and z represent the components of a position in the x, y, and z axes, respectively.
   In cases where this is unambiguous, the axis component may also be specified as a subscript, e.g., a 3-D vector $\mathbf{b}_{\square} = \begin{pmatrix} b_{\square,x} & b_{\square,y} & b_{\square,z} \end{pmatrix}^T$.

A. Inverted Pendulum Mode

The LIPM is derived from the Newton–Euler equations, assuming that angular momentum around the CoM is constant and that the CoM experiences no vertical acceleration. This leads to a relationship between the CoM projection $\mathbf{p}_c = (x_c \quad y_c)^T$ and the ZMP $\mathbf{p}_z = (x_z \quad y_z)^T$ on the (x, y) plane (corresponding to the ground)

$$\ddot{\mathbf{p}}_c = \omega^2(\mathbf{p}_c - \mathbf{p}_z) \tag{1}$$

where $\omega^2 = g/\hat{z}_c$, with g being the gravitational acceleration magnitude and $\hat{z}_c$ the constant difference between the CoM and the ZMP heights, with the ZMP set at the stance foot height.

This model, initially introduced for humanoid locomotion by Kajita et al. [27], was formulated as a linear–quadratic preview controller that generates a CoM trajectory based on a pregenerated ZMP objective trajectory. An extension of this model [34] allows for both the CoM and ZMP to be treated as decision variables. Constraints are applied to enforce balance and kinematic feasibility, ensuring that generated trajectories respect ZMP constraints and step parameters [7], [35], [36].

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

By introducing the following change of variables:

$$x_u = x_c + \frac{1}{\omega}\dot{x}_c \tag{2}$$

(1) can be rewritten as

$$\dot{x}_u = \omega(x_u - x_z) \tag{3}$$

where x_u represents the divergent component of motion (DCM), introduced in [2]. The DCM's dynamics are such that while the CoM converges exponentially toward the DCM, the DCM itself diverges exponentially from the ZMP. Controlling the first-order dynamics between the ZMP and DCM is fundamental to walking gait generation [7], [8], [36], [37], as it simplifies the system's dynamics and extends the balance stability criterion for gait generation.

Using the DCM dynamics, for a bounded ZMP trajectory $x_z(t)$ within an interval $t \in [\tilde{t}, \infty]$, the CoM dynamics remain bounded if and only if the initial DCM $x_u(\tilde{t})$ satisfies the following condition, as detailed in [38]:

$$x_u(\tilde{t}) = x_u^\star(\tilde{t}, x_z) \triangleq \omega \int_{\tilde{t}}^{\infty} x_z(\tau) e^{-\omega(\tau-\tilde{t})}\, d\tau. \tag{4}$$

This equation is the *stability condition*. Satisfying this condition ensures that the CoM remains bounded (provided that the ZMP or its velocity is bounded) at all times.

The stability condition offers several useful properties.

1) It provides a linear relationship between the current DCM and the ZMP trajectory.
2) As the ZMP location depends on the future step positions and durations (for both double- and single-support phases), the feasibility of a walking plan is inherently captured.

This condition can be seen as an alternative to penalizing divergent trajectories in the cost function (e.g., by minimizing CoM jerk), a heuristic solution that may depend on the prediction window of the MPC. It forms the foundation of the IS-MPC proposed in [7]. In [8], this condition is linearized with respect to step durations for forward walking, decoupling the constraints along the axes. Similarly, Khadiv et al. [37] use DCM dynamics (in point contact feet) to derive a bounding condition that allows simultaneous computation of the next step's timing and location.

Note that the DCM can be used as a baseline for relaxing the constraints on pendulum height, thereby creating a variable height pendulum model [28], [39].

To control a humanoid under LIPM assumptions, one must ensure that the robot's dynamics follow the generated gait. To mitigate the effects of modeling uncertainties and disturbances, a complementary policy based on DCM tracking feedback can be added to the ZMP control. This feedback ensures that the DCM converges exponentially to its planned reference [9], [40]. This feedback loop, commonly called the *stabilizer*, serves to "stabilize" the walking gait generated by the open-loop system, addressing the need for the gait to remain feasible despite uncertainties. Even though certain control schemes in the literature incorporate the robot's state in gait generation [7], [34], only a subset of the state is provided to the planner, as ZMP dynamics are not considered. As a result, a complementary policy is still necessary

B. Disturbed Pendulum

The ZMP in the LIPM represents the contact forces that can be controlled. However, additional contact forces may be applied to the system and must be considered.

To model the centroidal dynamics with the inclusion of an external disturbance wrench, let $(\mathbf{n}, \mathbf{f}) = (n_x \quad n_y \quad n_z \quad f_x \quad f_y \quad f_z)^T$ represent the external force and moment, applied at a contact point $\mathbf{p} = (p_x \quad p_y \quad p_z)^T$. The pendulum dynamics become as follows (refer to [41, eqs. (1)–(4)] for the derivation)

$$\ddot{x}_c = \omega^2 \left(x_c - \kappa x_z + \Delta x_c'\right) \tag{5}$$

$$\omega^2 = \frac{g + \ddot{z}_c}{z_c - z_z} \tag{6}$$

$$\kappa = 1 - \frac{f_z}{m(g + \ddot{z}_c)} \tag{7}$$

$$\Delta x_c' = \frac{1}{m(g + \ddot{z}_c)}[(p_z - z_z) f_x - p_x f_z + n_y - \dot{L}_{c,y}]. \tag{8}$$

Here, $\dot{\mathbf{L}}_c = \begin{pmatrix} \dot{L}_{c,x} & \dot{L}_{c,y} \end{pmatrix}^T$ is the derivative of the angular momentum at the CoM, z_c and z_z are the vertical components of the CoM and the ZMP, respectively, and m represents the mass of the robot. For the y-axis, $\Delta y_c'$ is similar to $\Delta x_c'$ and is given by

$$\Delta y_c' = \frac{1}{m(g + \ddot{z}_c)}[(p_z - z_z) f_y - p_y f_z - n_x + \dot{L}_{c,x}]. \tag{9}$$

By combining (2), (3), and (5) with a constant pendulum frequency ω, the DCM dynamics can be rewritten as

$$\dot{x}_u = \omega(x_u - (\kappa x_z - \Delta x_c')). \tag{10}$$

An external disturbance is thus modeled as an offset in the ZMP dynamics, with a proportional effect on the ZMP trajectory

C. Force Control

In a high-gain kinematic-controlled robot, the forces at the feet are regulated using admittance control, which tracks a reference wrench $\mathbf{w}_r$ by displacing the feet in the direction opposite to the desired force.

Let the 6-D spatial velocity vector of a robot link be represented as $(\boldsymbol{\omega}^T \quad \mathbf{v}^T)^T$. We use an acceleration-based tracking law [42], which relies on a reference spatial velocity that writes

$$\begin{pmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{v}} \end{pmatrix} = -K_d \left[\begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\omega}_r \\ \mathbf{v}_r \end{pmatrix} \right] \tag{11}$$

where K_d is the damping coefficient, and

$$\begin{pmatrix} \boldsymbol{\omega}_r^T & \mathbf{v}_r^T \end{pmatrix}^T = K_a(\mathbf{w}_m - \mathbf{w}_r) \tag{12}$$

where $\mathbf{w}_r$ and $\mathbf{w}_m$ represent the reference and measured 6-D wrenches, respectively, in the link frame, and K_a is the admittance gain.

For force control, the reference spatial velocity is defined such that [9]

$$\begin{pmatrix} \boldsymbol{\omega}_r^T & \mathbf{v}_r^T \end{pmatrix}^T = K_a(\mathbf{w}_m - \mathbf{w}_r) \tag{13}$$

where $\mathbf{w}_r$ and $\mathbf{w}_m$ represent the reference and measured 6-D wrench in the link frame, respectively, and K_a is the admittance gain. To control the robot's ZMP, we must regulate the center of pressure (CoP) at the contact points. This is achieved by controlling the moments in the x and y directions in the contact frame, as well as the vertical force. For moment control, we use (13) in two dimensions only. The vertical forces during the double-support phase are controlled using foot force difference control (FFDC) [43], [44]. Let (v_L, v_R) denote the vertical velocities of the left and right feet, respectively. FFDC with gain K_z updates the reference velocities $(v_{L,r}, v_{R,r})$ as follows [9]:

$$v_{L,r} \leftarrow v_{L,r} - 0.5 v_{\delta f} \tag{14}$$

$$v_{R,r} \leftarrow v_{R,r} + 0.5 v_{\delta f} \tag{15}$$

$$v_{\delta f} \triangleq K_z \left[(f_{L,r} - f_{R,r}) - (f_{L,m} - f_{R,m}) \right]. \tag{16}$$

When evaluating the performance of such a force control scheme, the deformation of the contact surface (either the foot sole or the ankle) must be taken into account. This holds true regardless of the type of control employed (kinematics or torque). In addition, many humanoid robots are equipped with a passive-compliant shock-absorbing mechanism between the ankle and the feet, with a thin layer of soft sole material. This flexibility can be modeled as a second-order system relating the torque at the contact point to the deformation [45]. Even in configurations without a shock-absorbing mechanism, our force control scheme, which relies on the position of an end-effector to achieve the desired force, this second-order behavior between the torque at the contact point and the deformation reflects the coupling between the desired CoP and the reference trajectory fed into the force control scheme. To simplify this, a first-order dynamic model has been empirically used to approximate the relationship between the CoP and its reference, with a force control law as defined in (13) [43].

III. Walking Gait Using Closed-Loop IS-MPC

A. Closed-Loop Control of the Pendulum—Problem Statement via a Simple Problem

We demonstrate the application of DCM feedback control and its challenges by analyzing the dynamics of an idealized 1-D pendulum, both analytically and via simulation. The pendulum is initialized with an arbitrary DCM and ZMP at time $t^0 = 0$, and the ZMP is set at $t = t^0 + \Delta t$ (with $\Delta t = 0.5$ s) to bring the pendulum to rest. While the pendulum's final position is irrelevant for this illustrative example, the same methodology can be applied when the final position is of interest. From (4), setting $x_z(\Delta t) = x_u(\Delta t)$ stabilizes the pendulum, as shown in Fig. 2(a). In this idealized case, the ZMP is instantaneously aligned with the DCM. However, this is physically impossible due to causality constraints.

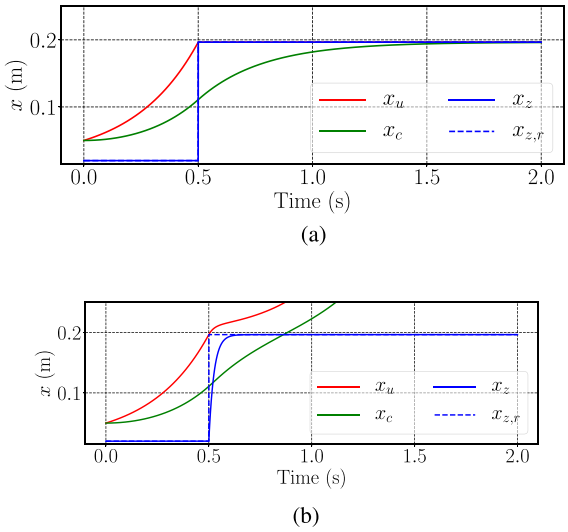

Fig. 2. LIPM behaviors for the simple example. (a) LIPM without ZMP bounds: the DCM can be captured by placing the ZMP on it. (b) Once the ZMP dynamics is introduced, the real DCM cannot be captured.

Previous works suggest considering a ZMP behavior $x_z(t)$ to track its reference $x_{z,r}$ with at least a linear first-order model, e.g., [43], that is, with a parameter λ

$$\dot{x}_z(t) = -\lambda(x_z(t) - x_{z,r}). \tag{17}$$

Integrating such ZMP dynamics to the LIPM, by setting $x_{z,r} = x_u(\Delta t)$, leads to instability [see Fig. 2(b)]. Indeed, a delay in getting the ZMP induces a deviation of the DCM; subsequently, the pendulum's dynamics is not anticipated at its initial pose.

One could then correct the ZMP reference using a DCM feedback control policy, which could be a simple proportional–integral–derivative (PID) on the DCM error. This strategy leads to an exponential convergence of the measured DCM toward the reference one [see Fig. 3(a)] [9]. The downside of this policy is to consider only the current state of the pendulum. Moreover, as the control references are changing, the higher level planning is disconnected from the state of the real robot, which could lead to violations of the planning constraints. Fig. 3 shows a gait generated using the IS-MPC in [7], where the corrective ZMP is out of the predefined constraints. Finally, it can be seen from [41] that if $\lambda < \omega$, this DCM feedback policy becomes unstable.

Our solution models the ZMP dynamics explicitly into the walking pattern generation [see Fig. 3(b)]. This amounts to solving (4) and (17) to compute a reference ZMP such that the real ZMP trajectory would follow the one computed by the MPC. This captures the combined dynamics with a single reference ZMP value. To our best knowledge, this is a novel approach that has never been considered in previous works.

The difference between the common control scheme in the literature and ours is illustrated in Fig. 3. Note that in real-case applications, the scheme shown in Fig. 3(a) can be used with a semiclosed-loop component by providing the DCM state to the pattern generator. However, because the force tracking dynamics

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

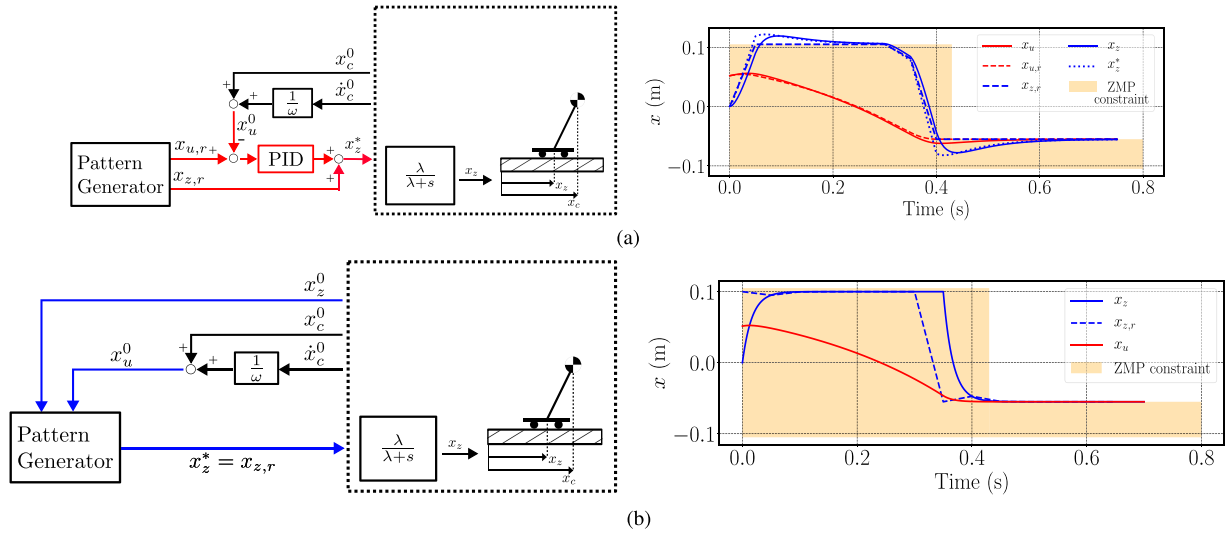

Fig. 3. LIPM (toy example) control schemes, highlighting our proposed approach. (a) Using a DCM feedback control $x_z^* = x_{z,r} + \text{PID}(x_u^0 - x_{u,r})$, we can correct the ZMP reference to be stable. (b) If the dynamic of the ZMP is known, the DCM can be captured using (4).

are not considered, such applications still necessitate an additional corrective policy (for instance based on DCM feedback) to ensure proper tracking of the pendulum state.

B. Control Variables

The IS-MPC generates a feasible CoM gait to realize a *footstep plan*, which includes both the location and orientation of the footsteps, as well as their durations, over a preview horizon (T_p) that extends beyond the MPC control horizon.

To make the robot stand still, we can generate a gait based on an empty footstep plan. This allows us to decouple the walking phase from the standing phase. Transitions between these phases can be triggered manually by the user or automatically (see Section IV-E).

The IS-MPC formulation initially employs the LIPM by using $\dot{x}_z$ (or equivalently $\dot{y}_z$) as the input term. The following dynamics are defined from (1), with the state $(x_c \quad \dot{x}_c \quad x_z)^T$:

$$\begin{pmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{x}_z \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ \dot{x}_c \\ x_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{x}_z. \tag{18}$$

However, for closed-loop operation on the robot's full state (including the ZMP and DCM), it is necessary to model the relationship between the ZMP reference forwarded to the force controller (e.g., admittance control) and the measured ZMP value. We model this interaction as a first-order system, similar to (17), with a parameter λ and a delay δ_d such that

$$\dot{x}_z = -\lambda(x_z(t) - x_{z,r}(t - \delta_d)). \tag{19}$$

The reason behind adding δ_d is an empirical observation resulting from experiments. Indeed, such a delay takes into account the lag between the instant the command is computed and the instant it is applied. It includes software latency and joint torque tracking response time. The dynamics in (18) can then be rewritten as follows (that also apply to the y-axis).

Delay δ_d is introduced based on empirical observations, accounting for the lag between the computation of the command and its actual application. This delay includes software latency and the joint torque tracking response time. With this, the dynamics in (18) can be rewritten as (same for the y-axis)

$$\begin{pmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{x}_z \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & -\lambda \end{pmatrix} \begin{pmatrix} x_c \\ \dot{x}_c \\ x_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \lambda \end{pmatrix} x_{z,r}(t - \delta_d). \tag{20}$$

At time t^0 with constant δ_d, the ZMP trajectory is given by

$$x_z(t) = x_z^0 + (x_z^* - x_z^0)(1 - e^{-\lambda(t-t^0)}), \quad \text{for } t \in [t^0; t^0 + \delta_d] \tag{21}$$

where x_z^0 is the current ZMP estimate and x_z^* is the commanded ZMP; we set $\bar{x}_z^0 = x_z(t^0 + \delta_d)$.

The reference $x_{z,r}(t)$ serves as the input to the force control loop. We consider it to be piecewise constant over a duration $\Delta t > \delta_d$, with the inputs u_x^k applied over $[t^k; t^{k+1}]$, where $t^k = t^0 + k\Delta t$. Thus, on each interval, we have

$$\begin{aligned} x_{z,r}(t) &= \bar{x}_z^0 + \sum_{l=0}^{k} u_x^l \\ \bar{x}_z^0 &= x_z^0 + (x_z^* - x_z^0)(1 - e^{-\lambda\delta_d}). \end{aligned} \tag{22}$$

Integrating the ZMP dynamics in (19) with the reference from (22) yields the following ZMP trajectory for $t \in [t^k + \delta_d; t^{k+1} + \delta_d]$:

$$x_z(t) = \bar{x}_z^0 + \sum_{l=0}^{k} u_x^l \left[1 - e^{-\lambda(t - t^l - \delta_d)}\right]. \tag{23}$$

Each $u_\square^l$ represents a first-order step delta input, with a delay of t_d^k. Fig. 4 illustrates the expected behavior of this model.

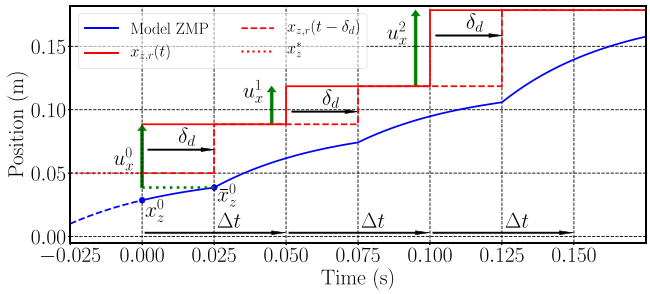

Fig. 4. Chosen first-order ZMP model with $\lambda = 25$. The green dashed line indicates the starting point of the first arrow.

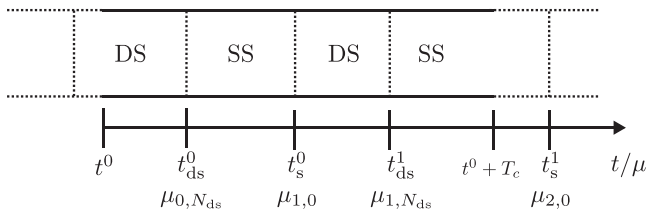

Fig. 5. Disposition of the planning time variables.

We define the control inputs as $\mathbf{u}^k = \begin{pmatrix} u_x^k & u_y^k \end{pmatrix}^T$ and the future contact location as $\mathbf{p}_f^i$, where i is the step number.

For each iteration, the closed-loop IS-MPC considers a control horizon of C samples, each with duration Δt. The footstep plan defines the number of steps N_s within the control horizon. From this, the following values are computed.

1) The reference inputs $U = \begin{pmatrix} \mathbf{u}^{0T} & \dots & \mathbf{u}^{C-1T} \end{pmatrix}^T$.
2) The footstep locations $P_f = \begin{pmatrix} \mathbf{p}_{f^1}^T & \dots & \mathbf{p}_f^{N_s T} \end{pmatrix}^T$, where $\mathbf{p}_f^0$ and $\mathbf{p}_f^{-1}$ represent the current and previous support foot locations at time t^0, respectively.

The fixed time parameters are given by $T_s = [(t_{\text{ds}}^0, t_{\text{s}}^0), \dots, (t_{\text{ds}}^{N_s-1}, t_{\text{s}}^{N_s-1})]$ which correspond to the start and end times of the single-support duration for each step.

Fig. 5 shows the organization of the time variables. $\hat{P}_f$ is the reference footstep location. The control horizon time is $T_c = C\Delta t$. Once computed, the linear inverted pendulum dynamics defined in (20) is integrated using (22) under computed input (u_x^0, u_y^0) at the sampling rate $\delta t \leq \Delta t$ (δt is the sampling period of the whole-body controller). Finally, the MPC solves the quadratic optimization problem (QP) [see (38)] under various linear constraints that enforce dynamic balance.

C. Constraints

1) ZMP Constraints: At time t^k, the ZMP position $\mathbf{p}_z(t^k) = \begin{pmatrix} x_z^k & y_z^k \end{pmatrix}^T \triangleq \mathbf{p}_z^k$ is constrained to lie within a convex polygon. This polygon is either located under the support foot or between both feet, depending on the current time t. Let $N(t)$ and $O(t)$ represent the normal matrix and the offset vector of the constraint, respectively. The constraint is given by

$$N(t^k)\mathbf{p}_z^k \leq O(t^k) \tag{24}$$

which defines C ZMP constraints in the form of (24).

For each MPC computation, during the current walking phase (single or double support), the ZMP constraint polygon is fully defined by the current robot foot positions and does not depend on any decision variables. Hence, the constraints can be set directly. Otherwise, $N(t)$ and $O(t)$ define the admissible ZMP position over the entire horizon and depend on the footstep positions P_f. Therefore, if we set N and O to cover the full support area, the constraint (24) becomes nonlinear with respect to P_f.

To maintain linearity between the ZMP constraints and P_f, the ZMP-allowed region is modeled as a rectangle with dimensions (dx, dy) that slides from one foot to the other. For each sample taken after t_s^0, the timing can be rewritten as a function of the step index $i \geq 1$ and an index $j \geq 0$ as follows:

$$t^k = t_s^{i-1} + j\Delta t. \tag{25}$$

As a result, we have $N(t^k = t_s^{i-1} + j\Delta t)$. Since the ZMP constraints are rectangular, the normal matrix is given by

$$N(t^k = t_s^{i-1} + j\Delta t) = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}^T R(t^k) \tag{26}$$

where $R(t^k = t_{\text{s}}^{i-1} + j\Delta t)$ is a 2×2 rotation matrix that defines the orientation of the rectangle. In double support, the orientation is defined by a linear interpolation between steps $i-1$ and i. In single support, the orientation of the rectangle is aligned with the support foot i.

The offset vector $O(t)$ is expressed within the interval $[t_{\text{s}}^{i-1}, t_{\text{s}}^i]$ as

$$O(t^k = t_{\text{s}}^{i-1} + j\Delta t) = \frac{1}{2}\begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + N(t^k)\left(\alpha^{i,j}\mathbf{p}_f^i + (1-\alpha^{i,j})\mathbf{p}_f^{i-1}\right) \tag{27}$$

$$\alpha^{i,j} = \frac{j\Delta t}{t_{\text{ds}}^i - t_{\text{s}}^{i-1}} \tag{28}$$

where $\alpha^{i,j} \in [0; 1]$ is the weighting variable that allows the rectangle to slide from one contact to the other.

2) Footstep Constraints: Within the leg's reachable space, the difference between two consecutive footsteps must be bounded. This can be enforced using N_{kin}^i and O_{kin}, where N_{kin}^i is a 4×2 normal matrix, and O_{kin} is a 4×1 offset vector for the step kinematic constraints. For $1 \leq i \leq N_s$, the constraint is given by

$$N_{\text{kin}}^{i-1}(\mathbf{p}_f^i - \mathbf{p}_f^{i-1}) \leq O_{\text{kin}}^i \tag{29}$$

$$O_{\text{kin}}^i = \frac{1}{2}\begin{pmatrix} dx_f & dy_f - l^i & dx_f & dy_f + l^i \end{pmatrix}^T \tag{30}$$

where l^i is an offset, which is positive if $\mathbf{p}_f^{i-1}$ corresponds to a step from the right foot, and negative otherwise. The matrix N_{kin}^{i-1} represents the normals of a rectangle oriented according

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

to $\mathbf{p}_f^{i-1}$ and is defined as in (26)

$$N_{\text{kin}}^{i-1} = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}^T R(t_s^{i-1}). \tag{31}$$

3) Stability Constraint: Using (4) to bound the CoM trajectory, the generated ZMP trajectory $x_z(t)$ and the current DCM position x_u^0 are constrained by

$$\begin{aligned} x_u^0 &= x_u^\star(t^0, \qquad\qquad x_z) \text{ and} \\ x_u^\star(t^0, x_z) &= \underbrace{\omega \int_{t^0}^{t^0+\delta_d} x_z(\tau) e^{-\omega(\tau-t^0)}\, d\tau}_{\bar{x}_\phi} \\ &\quad + \omega \int_{t^0+\delta_d}^{t^0+\delta_d+T_c} x_z(\tau) e^{-\omega(\tau-t^0)}\, d\tau \\ &\quad + \omega \int_{t^0+\delta_d+T_c}^{\infty} x_z(\tau) e^{-\omega(\tau-t^0)}\, d\tau. \end{aligned} \tag{32}$$

The term $x_u^\star(t^0, x_z)$ can be split into three integrals:

1) the part of the ZMP trajectory that is delayed and cannot be controlled (the interval $[t^0; t^0+\delta_d]$), denoted as $\bar{x}_\phi$;
2) the controlled part of the ZMP trajectory within the interval $[t^0+\delta_d; t^0+\delta_d+T_c]$;
3) the imposed ZMP trajectory in the interval $[t^0+\delta_d+T_c; \infty]$, denoted as $\tilde{x}_z(t)$, which represents the *tail* of the trajectory beyond the horizon, as defined in [7].

The stability constraint (32) becomes

$$x_u^0 - \bar{x}_\phi = \omega \int_{t^0+\delta_d}^{\infty} x_z(\tau) e^{-\omega(\tau-t^0)}\, d\tau. \tag{33}$$

Substituting the expression for $x_z(t)$ from (21), we obtain

$$\bar{x}_\phi = \omega \int_{t^0}^{t^0+\delta_d} x_z^0 + (x_z^* - x_z^0)\left[1 - e^{-\lambda(\tau-t^0)}\right] e^{-\omega(\tau-t^0)} d\tau. \tag{34}$$

We set $x_z(t)$ from (23). We define the control inputs beyond the horizon as $\tilde{u}_x^k$ for $k \geq C$. Equation (33) becomes

$$\begin{aligned} x_u^0 - \bar{x}_\phi &= \omega \int_{t^0+\delta_d}^{\infty} \bar{x}_z^0 e^{-\omega(\tau-t^0)}\, d\tau \\ &\quad + \sum_{k=0}^{C-1} \omega \int_{t^0+\delta_d+k\Delta t}^{\infty} u_x^k (1 - e^{-\lambda(\tau-t^k-\delta_d)}) e^{-\omega(\tau-t^0)}\, d\tau \\ &\quad + \underbrace{\sum_{k=C}^{\infty} \omega \int_{t^0+\delta_d+k\Delta t}^{\infty} \tilde{u}_x^k (1 - e^{-\lambda(\tau-t^k-\delta_d)}) e^{-\omega(\tau-t^0)}\, d\tau}_{\tilde{x}_\phi}. \end{aligned} \tag{35}$$

Finally, we have

$$x_u^0 - \tilde{x}_\phi - \bar{x}_\phi = e^{-\omega\delta_d}\left[\bar{x}_z^0 + \frac{\lambda}{\lambda+\omega} \sum_{k=0}^{C-1} u_x^k e^{-\omega k \Delta t}\right]. \tag{36}$$

4) Tailing: Tailing refers to the behavior of $\tilde{x}_z(t)$ beyond the control horizon, which can vary depending on the expected future motion (see [7] for more details).

1) *Truncated* tailing sets the ZMP to a stop beyond the control horizon, i.e.,

$$\forall i \geq C, \quad \tilde{u}_x^i = 0, \quad \tilde{x}_\phi = 0.$$

2) *Periodic* tailing assumes that the ZMP trajectory repeats periodically every T_c. Hence, $\tilde{x}_z(t)$ follows inputs $\tilde{u}_x^j$, where $\tilde{u}_x^j = u_x^i$ and $j \equiv i \pmod C$.
The stability condition is rewritten as

$$\begin{aligned} &x_u^0 - \bar{x}_\phi \\ &= e^{-\omega\delta_d}\left[\bar{x}_z^0 + \frac{\lambda}{(\lambda+\omega)(1-e^{-\omega C\Delta t})} \sum_{k=0}^{C-1} u_x^k e^{-\omega k \Delta t}\right]. \end{aligned} \tag{37}$$

3) *Anticipative* tailing uses the footstep plan beyond the control horizon to generate a reference ZMP trajectory. It allows $\tilde{x}_z(t)$ to track the desired behavior based on the walking plan. After that, the truncated or periodic policy is added to complete the tail, as described in [7].

D. Cost Function

The cost function is designed to generate a ZMP trajectory that satisfies the following weighted objectives.

1) *Predefined ZMP trajectory objective, $x_{z,obj}$:* It is also possible to set a component of the ZMP velocity magnitude, with an objective velocity set to zero.
2) *DCM trajectory objective:* This is obtained by computing the stable DCM $x_u^\star(t^0, x_{z,\text{obj}})$ and integrating it over the LIPM using (18), with $x_{z,\text{obj}}$ as the input. A DCM velocity objective can also be obtained using (3).
3) *Step location objective:* This aims to bring the step locations as close as possible to the reference values.

In summary, the decision variables over the horizon are

$$U = \begin{pmatrix} \mathbf{u^0}^T & \dots & \mathbf{u}^{C-1^T} \end{pmatrix}^T, P_f = \begin{pmatrix} \mathbf{p}_{f1}^T & \dots & \mathbf{p}_f^{N_s^T} \end{pmatrix}^T.$$

We use similar notation for the ZMP and DCM position and velocity sequences. The IS-MPC solves the following QP:

$$\begin{aligned} U, P_f = \underset{U,P_f}{\arg\min}\ & \beta_{\dot{z}} \|\dot{P}_z\|_2 + \beta_z \|P_z - P_{z,\text{obj}}\|_2 \\ & + \beta_f \|P_f - \hat{P}_f\|_2 + \beta_u \|P_u - P_{u,\text{obj}}\|_2 \\ & + \beta_{\dot{u}} \|\dot{P}_u - \dot{P}_{u,\text{obj}}\|_2 \end{aligned} \tag{38}$$

under the following constraints:

1) ZMP (24);
2) footsteps (29);
3) stability (36).

E. External Forces

From (10), if the humanoid is subjected to expected or unexpected external forces, the dynamics of the pendulum must be controlled under a new state $\kappa x_z - \Delta x'_c$. If the duration $\delta_p > \delta_d$ for which these external forces are active is known, the stability condition (4) can be rewritten as follows:

$$\begin{aligned} x_u^\star(t^0, x_z) = \omega \int_{t^0}^{\infty} (\kappa x_z(\tau) - \Delta x'_c) e^{-\omega(\tau - t^0)} d\tau \\ -\omega \int_{t^0+\delta_p}^{\infty} (\kappa x_z(\tau) - \Delta x'_c) e^{-\omega(\tau-(t^0+\delta_p)+\delta_p)} d\tau \\ +\omega \int_{t^0+\delta_p}^{\infty} x_z(\tau) e^{-\omega(\tau-(t^0+\delta_p)+\delta_p)} d\tau. \end{aligned} \tag{39}$$

Using the definition in (4), we obtain the following expression for the stability condition under the perturbation:

$$\begin{aligned} x_u^0 = x_u^\star(t^0, \kappa x_z - \Delta x'_c) - e^{-\omega\delta_p} x_u^\star(t^0 + \delta_p, \kappa x_z - \Delta x'_c) \\ + e^{-\omega\delta_p} x_u^\star(t^0 + \delta_p, x_z). \end{aligned} \tag{40}$$

Thus, the stability condition is updated to account for the effect of these external forces on the dynamic model.

IV. Step Timing and Position Preoptimization

We describe the process for updating step planning when the current plan is not feasible. Let N_s denote the number of planned steps. Given the current constraints in (38), the feasible region of the current DCM can be computed. The stability condition is reformulated by combining (4) with (24). To do so, we assume that the normals N defined in (26) are constant and denote them as $\bar{N}$.

Theorem 1: If the ZMP constraint is defined as $\bar{N}\mathbf{p}_z(t) \leq O(t)$ for $t \in [t^0; \infty]$, then the stability condition is satisfied iff

$$\bar{N}\mathbf{p}_u^0 \leq \omega \int_{t^0}^{\infty} O(\tau) e^{-\omega(\tau - t^0)}\, d\tau. \tag{41}$$

The proof of this theorem is given in the Appendix.

This condition, derived from the IS-MPC constraints, ensures that the current DCM remains within a polygon (referred to as the *feasibility region*) generated by (41). This guarantees that the ZMP trajectory will keep the DCM trajectory bounded, thus ensuring a solution for the IS-MPC. The *feasibility region* represents the planned motion of the pendulum, as $O(t)$ specifies the location and size of the ZMP constraint, which depends on the step location. In addition, since the ZMP constraints are time dependent, the stability condition is also influenced by step durations.

For instance, Fig. 6 illustrates three different step plans. The target step is represented by the dashed rectangles, with their respective step frequencies. The solid colored rectangles show the resulting *feasibility region*. In cases (a) and (b), if the DCM position along the x-axis is outside the feasibility region (e.g., due to a push in the $+x$ direction), the planned step becomes infeasible, and a forward step is required [as in case (c)]. Furthermore, in case (b), if the DCM is outside the pink region along the y-axis due to a push in the $-y$ direction, changing the footstep location can be challenging, as it may cause a collision between the feet. However, reducing the step frequency [as in case (a)] can lead to a feasible solution. Thus, the ability to update the footstep plan in terms of step position and/or step timings significantly enhances the humanoid's balance capabilities.

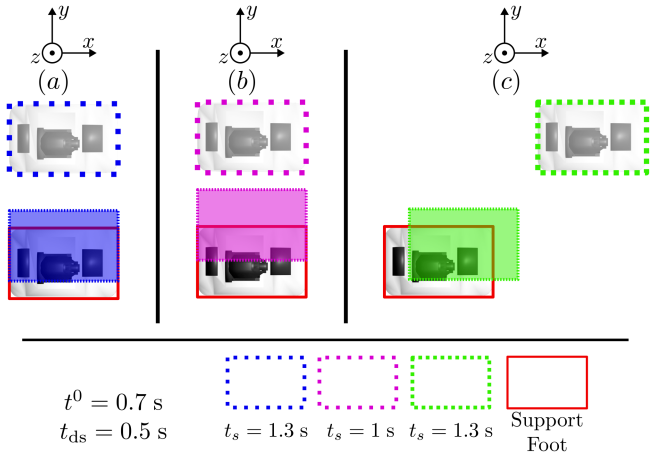

Fig. 6. Top view of the foot and footprints with the feasibility region as colored rectangles for three cases of step target and step frequency. The current phase at t^0 is single support.

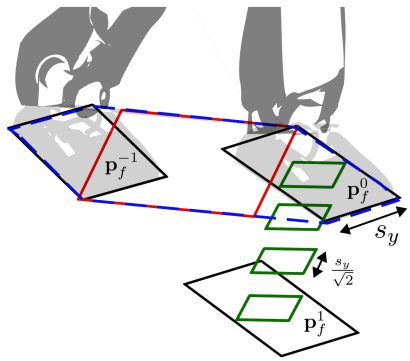

Fig. 7. View of the foot and footprints with the computation of the ZMP constraint rectangle shape. The red rectangle, once computed, defines $\bar{N}$, and the green squares represent future ZMP constraints oriented accordingly.

A. Problem Constraints

We discretize (41) and define the kth ZMP constraint at time t_c^k such that $\forall t \in [t_c^k; t_c^{k+1}]$, $\bar{N}\mathbf{p}_z(t) \leq O(t = t_c^k)$. Note that the duration $t_c^{k+1} - t_c^k$ is not necessarily constant for all k.

The stability condition in (41) is rewritten as

$$\bar{N}\mathbf{p}_u^0 \leq \sum_{k=0}^{\infty} O(t = t_c^k) \left[e^{-\omega(t_c^k - t^0)} - e^{-\omega(t_c^{k+1} - t^0)} \right]. \tag{42}$$

We also introduce slack variables $\mathbf{s}$ into the constraints (44) to retrieve the parameters bound with the activated constraints. We model the ZMP constraint as a rectangular region of dimensions (dx, dy) inside the support foot during the single-support phase, which slides linearly to the other foot during the double-support phase. This path is discretized into a fixed number of positions, $N_{\text{ds}} + 1$, for each step. Fig. 7 illustrates this case for $N_{\text{ds}} = 3$. We define t_c^k as a function of the ith step and the jth ZMP

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

constraint, denoted as $t^{i,j}$, as follows:

$$\begin{aligned} t^{i,j} &= t_c^{i(N_{\text{ds}}+1)+j} \\ t^{i,0} &= t_s^{i-1}, \quad \text{for } i \geq 1 \\ t^{i,N_{\text{ds}}} &= t_{\text{ds}}^{i}, \quad \text{for } i \geq 0. \end{aligned} \tag{43}$$

Fig. 5 shows how the indices are assigned relative to the walking phases. We split the sum in (42) into three intervals:

1) single-support phases (intervals $[t^{i,N_{\text{ds}}}; t^{i+1,0}]$);
2) double-support phases (intervals $[t^{i,j}; t^{i,j+1}]$);
3) intervals beyond the footstep plan horizon.

We note $\mu^{i,j} = e^{-\omega t^{i,j}}$, $O^{i,j} = O(t^{i,j})$, and $\alpha_j = j/N_{\text{ds}}$ and rewrite the stability condition, for a plan of N_s steps as follows:

$$\begin{aligned} e^{-\omega t^0}\bar{N}\mathbf{p}_u^0 \leq \sum_{i=0}^{N_s-1}\Bigg[& O^{i,N_{\text{ds}}}(\mu^{i,N_{\text{ds}}} - \mu^{i+1,0}) \\ & + \sum_{j=0}^{N_{\text{ds}}-1} O^{i,j}(\mu^{i,j} - \mu^{i,j+1})\Bigg] + \tilde{O}. \end{aligned} \tag{44}$$

The offset ZMP constraints are given by

$$O^{i,j} = \frac{1}{2}\begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + \bar{N}\left[\alpha_j \mathbf{p}_f^i + (1-\alpha_j)\mathbf{p}_f^{i-1}\right]. \tag{45}$$

For the case where the current walking phase is in single support, we set

$$\forall j \in [0, N_{\text{ds}}],\ \mu^{0,j} = e^{-\omega t^0}.$$

. We can identify three parts in the inequality (44).

1) The first sum represents the single-support phase, where $\mu^{i,N_{\text{ds}}} = e^{-\omega t_{\text{ds}}^i}$ and $\mu^{i+1,0} = e^{-\omega t_s^i}$; thus, $O^{i,N_{\text{ds}}} = O^{i+1,0}$, which corresponds to the ZMP constraint under the foot $\mathbf{p}_f^i$.
2) The inner sum represents the double-support phase, where the ZMP constraint slides from $\mathbf{p}_f^{i-1}$ to $\mathbf{p}_f^i$.
3) $\tilde{O}$ represents the tailing, which defines the ZMP constraints after the footstep plan horizon.

Except for $i = 0$, the $O^{i,j}$ terms in (45) are similar to those in (27); however, during the double-support phase, their number is constant, N_{ds}, whereas in Section III, the number of terms varies with the double-support phase duration divided by the sampling rate Δt.

For $i = 0$, the positions of the footsteps $\mathbf{p}_f^{-1}$ and $\mathbf{p}_f^0$ are fixed. We set $O^{0,j}$ constant for all j to ensure that the ZMP constraint in the current walking phase best approximates the current support polygon. Fig. 7 illustrates this, and Section IV-B provides additional details.

As in (36), the tailing $\tilde{O}$ accounts for the planning beyond the horizon. Only a truncated or periodic tailing can be used, as there are no further steps planned beyond this horizon. Since our plan is defined with a sufficiently long horizon (more than three steps), we set a truncated tail such that the ZMP constraint reaches the midpoint between the last two contacts as follows:

$$\tilde{O} = \sum_{j=0}^{N_{\text{ds}}-1} O^{N_s,j}(\mu^{N_s,j} - \mu^{N_s,j+1}) + O^{N_s,N_{\text{ds}}}\mu^{N_s,N_{\text{ds}}}$$

$$O^{N_s,j} = \frac{1}{2}\begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + \bar{N}\left[\frac{\alpha_j}{2}\mathbf{p}_f^{N_s} + (1-\frac{\alpha_j}{2})\mathbf{p}_f^{N_s-1}\right]. \tag{46}$$

From (42), we obtain a constraint that relates the positions of the footsteps and their timings to guarantee the stability condition. Therefore, we can formulate the following optimization problem with:

1) the step timings $\mu^{i,j}$;
2) the footstep positions $\mathbf{p}_f^i$

as decision variables. The step timings can be retrieved as

$$\begin{aligned} t_{\text{ds}}^i &= -\frac{1}{\omega}\ln(\mu^{i,N_{\text{ds}}}) \\ t_{\text{s}}^i &= -\frac{1}{\omega}\ln(\mu^{i+1,0}) \end{aligned} \tag{47}$$

which gives the timing plan illustrated in Fig. 5.

We also add constraints to this optimization problem on: 1) the step locations, as defined in Section III-C2, and 2) the step phase durations, such that:

$$\begin{aligned} \Delta t_{\text{ds,m}} \leq t_{\text{ds}}^i - t_{\text{s}}^{i-1} &\leq \Delta t_{\text{ds,m}} \\ \Delta t_{\text{ss,m}} \leq t_{\text{s}}^i - t_{\text{ds}}^i &\leq \Delta t_{\text{ss,m}} \\ \Delta t_{\text{s,m}} \leq t_{\text{s}}^i - t_{\text{s}}^{i-1} &\leq \Delta t_{\text{s,m}} \end{aligned} \tag{48}$$

where $\Delta t_{\text{ds,m}}$, $\Delta t_{\text{ss,m}}$, and $\Delta t_{\text{s,m}}$ are the minimum double-support, single-support, and total step durations, respectively. The superscript M denotes the upper bound of these constraints.

Let $\mu_{\text{ds,m}} = e^{-\omega \Delta t_{\text{ds,m}}}$ and similarly for the other duration bounds. The timing constraints are

$$\begin{aligned} \mu_{\text{ds,m}}\mu^{i,0} \geq \mu^{i,N_{\text{ds}}} &\geq \mu_{\text{ds,m}}\mu^{i,0} \\ \mu_{\text{ss,m}}\mu^{i,N_{\text{ds}}} \geq \mu^{i+1,0} &\geq \mu_{\text{ss,m}}\mu^{i,N_{\text{ds}}} \\ \mu_{\text{s,m}}\mu^{i,0} \geq \mu^{i+1,0} &\geq \mu_{\text{s,m}}\mu^{i,0}. \end{aligned} \tag{49}$$

B. ZMP Constraint Region Choice

Since the normals $\bar{N}$ are constant, the size of the ZMP constraints may vary, but their orientations remain similar.

The first ZMP constraint, represented by the red rectangle in Fig. 7, covers the largest area of the current support polygon [46]. The remaining constraints are defined as shown by the green rectangles in Fig. 7.

The size of the rectangle is chosen to always fit within the upcoming support polygon. A conservative approach is to define the rectangle as a square with side length $s_y/\sqrt{2}$, where s_y is the width of the robot's foot.

By doing so, the allowed ZMP region is reduced over the horizon, but the ZMP constraint still covers the largest possible area in the current walking phase. Since the contribution of the

ZMP trajectory to the stability condition decays exponentially as the horizon extends [due to the time-dependent exponential term in (4)], this approach provides a reasonable approximation of the feasible region. This is especially effective as optimization is performed at the MPC update frequency within the control loop.

C. Problem Formulation

The decision variables are $P_f = \begin{pmatrix} \mathbf{p}_{f1}^T & \dots & \mathbf{p}_f^{N_s T} \end{pmatrix}^T$ and $\mu = \begin{pmatrix} \mu_{i,j} \; i \in [\![0; N_s - 1]\!], j \in [\![0; N_{\text{ds}}]\!] \end{pmatrix}^T$. The desired step positions are given by $\hat{P}_f = \begin{pmatrix} \hat{\mathbf{p}}_f^{1\,T} & \dots & \hat{\mathbf{p}}_f^{N_s\,T} \end{pmatrix}^T$. Using the desired step durations $\hat{T}_s = [(\hat{t}_{\text{ds}}^0, \hat{t}_{\text{s}}^0), \dots, (\hat{t}_{\text{ds}}^{N_s-1}, \hat{t}_{\text{s}}^{N_s-1})]$, we compute the desired values of $\hat{\mu}^{i,j}$ as follows:

$$\hat{\mu}^{i,j} = e^{-\omega \left[\hat{t}_s^{i-1} + \frac{j}{N_{\text{ds}}} (\hat{t}^i{}_{\text{ds}} - \hat{t}_s^{i-1}) \right]}. \tag{50}$$

We then formulate the optimization problem as

$$\begin{aligned} P_f, \mu, s = \underset{\mathbf{p}_f^i, \mu^{i,j}, \mathbf{s}}{\arg\min} \; & \| \sum_{i=0}^{N_s-1} \sum_{j=0}^{N_{ds}} \hat{\mu}^{i,j} - \mu^{i,j} \|_2^2 + w_s \|\mathbf{s}\|_2^2 \\ & + \sum_{i=1}^{N_s-1} K_s^i \|\hat{\mathbf{p}}_f^i - \mathbf{p}_f^i\|_2^2 \times \|\mu^{i,0} - \mu^{i-1,0}\|_2^2 \\ \text{s.t.} \; & \text{stability } (44) \\ & \text{kinematics } (29) \\ & \text{timing } (49) \\ & \mu^{i,j} \leq e^{-\omega t^0}. \end{aligned} \tag{51}$$

The objective function in problem (51) consists of three main components.

1) The first term minimizes the deviation from the desired step durations.
2) The second term, weighted by $w_s \gg 1$, penalizes the slack variables, which are nonzero only when the constraints are not satisfied.
3) The third term minimizes the difference between the current and desired step positions. It is scaled by the factor $\|\mu^{i,0} - \mu^{i-1,0}\|_2^2$, which encourages faster steps when there is a larger deviation from the reference.

The gain K_s^i is chosen such that

$$K_s^i = \frac{K_s}{\|\hat{\mu}^{i,0} - \hat{\mu}^{i-1,0}\|_2^2}. \tag{52}$$

Here, the denominator normalizes K_s^i relative to the term $\|\mu^{i,0} - \mu^{i-1,0}\|_2^2$. The output timings correspond to the reference step durations T_s used in the IS-MPC framework described in Section III.

D. Solving Using Block Coordinate Descent

The constraint defined by (44) is quadratic and nonconvex, making the problem complex and computationally expensive with respect to our MPC. However, this constraint becomes linear if we treat either the positions or the timings of the steps as independent decision variables.

To overcome this challenge, we adopt a block coordinate descent method [47], where the problem is solved alternately with fixed step timings or fixed step locations. This approach allows us to reformulate the entire problem as two sequential QP optimizations. By framing the problem as two successive QPs, we first solve for the step locations, followed by the step durations. The rationale behind this order is that, without slack variables or constraints on kinematics and step duration, problems with fixed step durations always have a solution, while problems with fixed step locations might not. Therefore, solving for step locations first is more likely to yield a feasible initial solution. Finally, since the IS-MPC updates the footstep locations at each iteration, it is more advantageous to conclude the alternating process by computing the step durations.

This alternating optimization procedure has been employed in other research, such as in [48], to solve similar problems. The process may be iterated until a satisfactory solution emerges. At each iteration, the constraints (44) and the cost function for each QP are formulated using the previously computed μ and P_f, respectively. We have chosen to perform this alternating procedure twice (i.e., solving four QPs) as it efficiently strikes a balance between optimizing step duration and step location, within a short computation time. The footstep plan is updated at each IS-MPC iteration. During this alternating process, for any set (μ, P_f) that satisfies all constraints, the result at the next iteration will, at worst, be identical to the previous set. Consequently, the algorithm cannot alternate indefinitely between two solutions.

It is important to note that if the initial walking plan is feasible, the output remains unchanged by design. Otherwise, the primary goal is to obtain a new feasible walking plan. In this case, our method aims primarily at improving the humanoid's balance.

E. Stepping Recovery While Standing

When the humanoid is standing still, the region defined in (44) represents a geometric constraint on the position of the DCM, indicating where it can be located without necessitating a step. Since we use a rectangular shape constraint for the ZMP, the DCM's stability region is likewise rectangular. By determining which vertices of the stability region the DCM violates, we can plan the appropriate support foot to use for recovery. This decision depends on the current foot positions.

This strategy proves effective for forward and backward pushes, as discussed in Section VII. Under lateral pushes, the support foot is chosen to be the one furthest from the current DCM position. However, for improved balance—particularly in the case of lateral disturbances—additional recovery strategies must be developed. This could involve techniques such as using angular momentum, as suggested in [49], or other advanced methods.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

V. Robot Force Control

The gait generated in Section III is passed to a whole-body task-space controller, which regulates both the CoM trajectory and the contact forces to achieve stable walking behavior.

A. CoP Regulation

The ZMP dynamics $\mathbf{p}z(t) = \begin{pmatrix} x_z(t) & y_z(t) \end{pmatrix}^T$ is modeled as a first-order system with delay relative to the reference ZMP $\mathbf{p}z, r(t) = \begin{pmatrix} x_{z,r}(t) & y_{z,r}(t) \end{pmatrix}^T$ computed by the IS-MPC controller in Section III. The admittance control law applies a reference net wrench to track the ZMP dynamics, as given by (23). During single-support phases, the admittance control applies a wrench to achieve the desired CoP, $\mathbf{p}_{z,r}$, at the contact point. The delay in the ZMP dynamics causes this reference to sometimes lie outside the support polygon, resulting in a faster change in the ZMP position. During the double-support phase, the total reference wrench is distributed across both feet, and a separate CoP reference is computed for each foot. We model each foot's contact as having a first-order dynamic response to the actual CoP, characterized by a time constant λ_c, where ideally $\lambda_c \simeq \lambda$. We denote $\mathbf{c}L, r$ and $\mathbf{c}R, r$ as the reference CoPs for the left and right feet, respectively, and $\mathbf{c}_L$ and $\mathbf{c}_R$ as the modeled CoPs. This model, with a piecewise-constant reference, exhibits dynamics similar to (19) and is described by the following differential equation for the left foot:

$$\dot{\mathbf{c}}_L(t) = -\lambda_c(\mathbf{c}_L(t) - \mathbf{c}_{L,r}(t - \delta_d)) \tag{53}$$

which, when integrated over the time interval $[t^0 + \delta_d; t^0 + \delta_d + \Delta t]$, yields

$$\begin{aligned} \mathbf{c}_L(t) &= \mathbf{c}_{L,r}(t^0) + (\bar{\mathbf{c}}_L^0 - \mathbf{c}_{L,r}(t^0))e^{-\lambda_c(t-(t^0+\delta_d))} \\ \bar{\mathbf{c}}_L^0 &= \mathbf{c}_L^* + (\mathbf{c}_L^0 - \mathbf{c}_L^*)e^{-\lambda_c\delta_d}. \end{aligned} \tag{54}$$

Here, the superscript $\square^*$ refers to the current reference values applied to the robot. Similarly, the vertical forces at the left and right feet, f_L and f_R, are modeled as follows:

$$\begin{aligned} f_L(t) &= f_{L,r} + (\bar{f}_L^0 - f_{L,r})e^{-\lambda_f(t-(t^0+\delta_d))} \\ \bar{f}_L^0 &= f_L^* + (f_L^0 - f_L^*)e^{-\lambda_f\delta_d}. \end{aligned} \tag{55}$$

The same formulation holds for the right foot. Then, we set $\mathbf{c}_{L,r}$ and $\mathbf{c}_{R,r}$ to have the same overall ZMP, i.e.,

$$\frac{f_L(t)\mathbf{c}_L(t) + f_R(t)\mathbf{c}_R(t)}{f_L(t) + f_R(t)} = \mathbf{p}_z(t). \tag{56}$$

This equation couples the CoPs and ZMP. Assuming that the vertical forces are known in advance, this relationship is linear. There are multiple ways to distribute the vertical forces and CoPs to achieve the desired ZMP. In this work, we minimize the moments at the ankles.

Let $\mathbf{p}_L$ and $\mathbf{p}_R$ denote the locations of the left and right ankles, respectively, and let m be the mass of the humanoid. To compute the vertical forces $f_L(t)$ and $f_R(t)$, we define $\gamma(t) \in [0, 1]$ as the normalized projection of the vector $\mathbf{p}_z(t) - \mathbf{p}_L$ onto the vector $\mathbf{p}_R - \mathbf{p}_L$, and set

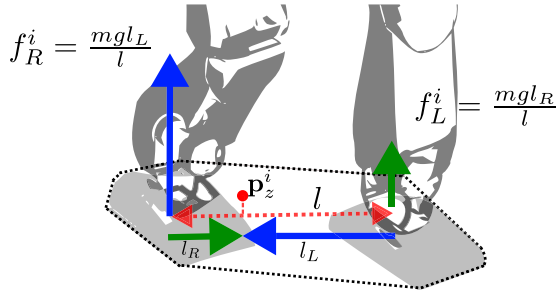

Fig. 8. Vertical force distribution depending on the desired ZMP position at time t^i. We have $\gamma = l_L/l$.

$$\begin{aligned} f_R(t) &= mg\gamma(t) \\ f_L(t) &= mg(1 - \gamma(t)). \end{aligned} \tag{57}$$

Fig. 8 illustrates the force distribution computation. The reference vertical forces are set by (55) using λ_f.

To compute $\mathbf{c}L, r$ and $\mathbf{c}R, r$, we formulate an optimization problem to approximate (56) quadratically. Since the ZMP horizon extends through the entire double-support phase, we consider the force distribution over the entire duration of the double-support phase, which lasts from t^0 to t^0_{ds}.

Let $\mathbf{c}L^i$ and $\mathbf{c}L, r^i$ represent the CoP location and the corresponding reference at time $t^i = t^0 + i\Delta t$, respectively, for the left foot (the notation is similar for the right foot). Using (54) and (53), we compute the CoP recursively

$$\mathbf{c}_L^i = \begin{cases} \mathbf{c}_{L,r}^{i-2}(e^{\lambda\delta_d} - 1)e^{-\lambda\Delta t} + \mathbf{c}_{L,r}^{i-1}(1 - e^{-\lambda(\Delta t - \delta_d)}) \\ \qquad +\mathbf{c}_L^{i-1}e^{-\lambda\Delta t}, \quad \text{if } i \geq 2 \\ \mathbf{c}_L^*(e^{\lambda\delta_d} - 1)e^{-\lambda\Delta t} + \mathbf{c}_{L,r}^0(1 - e^{-\lambda(\Delta t - \delta_d)}) \\ \qquad +\mathbf{c}_L^0 e^{-\lambda\Delta t}, \quad \text{if } i = 1. \end{cases} \tag{58}$$

With the modeled ZMP $\mathbf{p}z^i = \mathbf{p}z(t^i)$ at time $t^i = t^0 + i\Delta t$, the future vertical forces f_L^i and f_R^i are computed using (57), as shown in Fig. 8. Subsequently, we compute the CoPs $\mathbf{c}L, r^i$ and $\mathbf{c}R, r^i$ for $i \in [0, \lfloor \frac{t^0_{\text{ds}} - t^0}{\Delta t} \rfloor]$ by solving the following QP:

$$\begin{aligned} \underset{\mathbf{c}_{L,r}^i, \mathbf{c}_{R,r}^i}{\arg\min} \quad & w_z \| \sum_i \frac{\mathbf{c}_L^i f_L^i + \mathbf{c}_R^i f_R^i}{f_L^i + f_R^i} - \mathbf{p}_z^i \|_2 + \\ & w_d \| \sum_i R_L(\mathbf{c}_L^i - \mathbf{p}_L) - R_R(\mathbf{c}_R^i - \mathbf{p}_R) \|_2 \\ \text{s.t.} \quad & \mathbf{c}_L^i, \mathbf{c}_R^i \in \text{their contact polygon} \end{aligned} \tag{59}$$

with $w_d \ll w_z$; R_R and R_L are the rotation matrices to the right and left contact frame, respectively. The first term of the cost function is the ZMP tracking error minimization, and the second term minimizes the moment at the ankles.

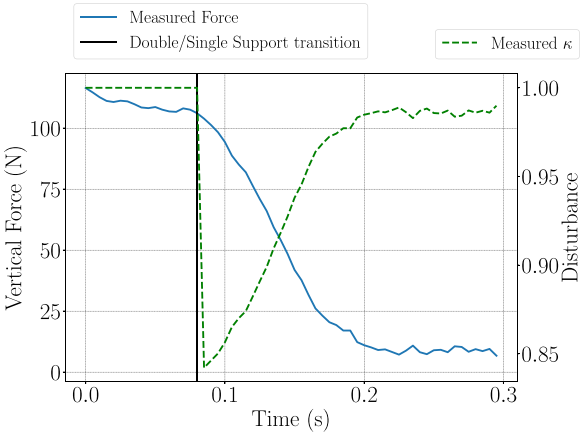

Fig. 9. Measured disturbances during foot contact release. A residual force of approximately 100 N is observed after the swing foot force control is deactivated, causing a change in the pendulum dynamics.

B. Contact Transition

Upon foot contact release, disabling force control can result in residual forces on the released foot, which may disturb the pendulum dynamics (see Fig. 9). To mitigate this, we include these residual forces in the dynamics of the pendulum. We can update the IS-MPC stability constraints using (40) to account for disturbances of duration δ_p (empirically set to 0.1 s based on experimental data).

To estimate the robot's mass, we use force sensors located at each foot in combination with a low-pass filter with a very long cutoff period (≥ 10 s). More accurate estimates of the robot's mass and the vertical acceleration $\ddot{z}_c$ can be obtained by using the raw unfiltered sensor data, which can be useful for updating the robot's model, e.g., when it is holding objects.

VI. Simulations

Real-world experiments cannot be conducted before extensive simulations of all the humanoid robots described in Section VII. In this section, we present our simulation environment and provide a comparative analysis between our closed-loop IS-MPC implementation and its original open-loop version. A more detailed data analysis will be presented in Section VII through real-world experiments.

A. Simulation Environment

The rigid-body simulation of our humanoids and their environment is carried out using the `Choreonoid` framework, integrated with the OpenHRP3 physics engine [50]. The joint commands generated by the whole-body control (see Section VII) are directly transmitted to the simulator, which also incorporates the low-level joint control dynamics. This enables a seamless transition to real robot control once the simulation results are satisfactory. The `Choreonoid` environment also simulates inertial measurement units, force/torque sensors, and potentially other types of sensors. In addition, state observers and estimators embedded within the robot are implemented identically in the simulation. The simulated environments include flat terrains, uneven surfaces, and slopes (see Fig. 10). However, reliable simulation of soft terrains is currently not feasible.

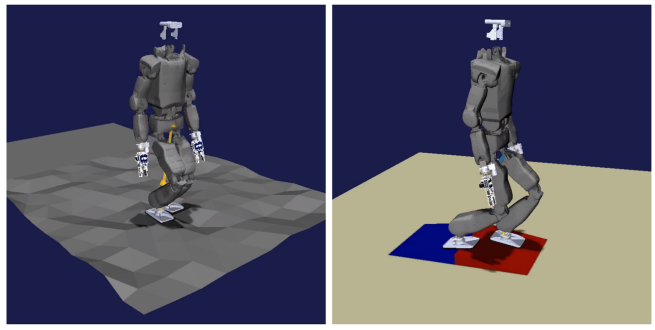

Fig. 10. Choreonoid simulation environment for uneven terrain and slope with RHPS1 robot.

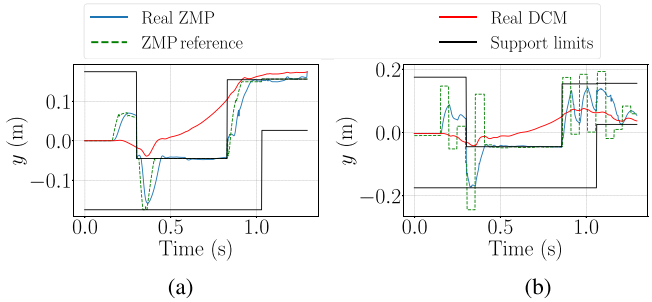

Fig. 11. Simulation comparative between DCM closed-loop IS-MPC and closed-loop IS-MPC. (a) DCM closed-loop IS-MPC. (b) Closed-loop IS-MPC.

B. Stabilizer and Closed-Loop Comparison

In addition to the theoretical advantages of our method, we conducted several comparative studies in simulation, under ideal conditions (no noise, flat terrain). For example, we instructed our robots to initiate walking with high-frequency steps without any adaptation of the footsteps. This was done to emphasize the challenge posed by a short double-support duration, which forces the robot's ZMP to quickly lie beneath the support foot, exposing the limitations of the original *stabilizer* policy. We compared three walking control schemes:

1) open-loop IS-MPC with a stabilizer;
2) DCM closed-loop IS-MPC with a stabilizer;
3) closed-loop IS-MPC (our proposed scheme).

Fig. 11 illustrates the measured DCM and ZMP, as well as the reference ZMP for the two closed-loop cases. Note that the ZMP reference generated by the *stabilizer* does not account for ZMP dynamics, which restricts it to references within the support polygon. Furthermore, as shown in Fig. 11(b), the delay δ_d is negligible. In Fig. 11, the measured ZMP can exceed the constraint due to the support polygon being slightly smaller than the actual one, allowing for safety margins during experimentation.

VII. Experiments

The proposed control scheme is implemented on five different humanoid platforms: HRP-2KAI, HRP-4, HRP-4CR, HRP-5P, and RHPS1 (i.e., all the humanoids in our possession), as shown

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

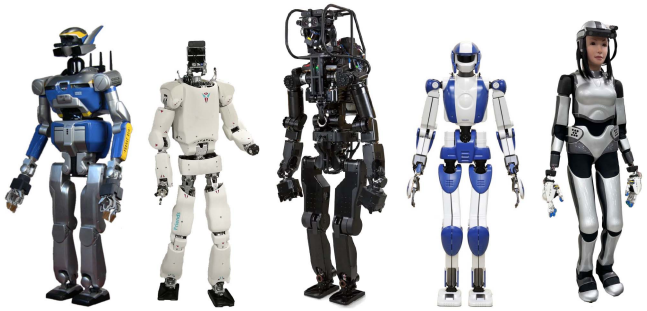

Fig. 12. Humanoid robots used in the experiments: from left to right, HRP-2KAI (modified HRP-2 used in the DRC), RHPS1 (a new humanoid from Kawasaki Robotics co-designed with AIST and CNRS), HRP-5P (latest HRP family prototype designed for torque control and building applications [52]), HRP-4 (used for the Airbus joint project [53]), and HRP-4CR (used in the ANA Avatar XPrize contest [54]).

in Fig. 12. The control scheme is tested across various environments, including:

1) flat floors;
2) flat floors with small obstacles (height $\lessapprox 3$ cm);
3) compliant terrains (not reaching compression limits);
4) outdoor terrains (only HRP-4 and HRP-2KAI).

Disturbances, such as human pushes, are introduced during both walking and standing phases. In addition, the control scheme was used during multiple public demonstrations at the IREX exhibition in November 2023, in a teleoperation scenario [51].

All humanoids are controlled in a high-gain kinematic mode (position control). Whole-body control is achieved using the `mc_rtc` task-space QP control framework,[1] as described in [55]. This framework computes the accelerations of the floating base and joints, which are then integrated twice and sent as joint commands. The goal of these extensive experiments is to develop a "plug-and-play" walking controller—referred to as a *plug-and-walk* software—that can be easily adapted and tested across multiple humanoid platforms. It is well known that most of the existing walking algorithms are highly tuned for specific humanoid or bipedal robots. However, such tunings are rarely explicitly described in academic papers.

For each robot, we experimentally adjust several model parameters, including:

1) the MPC's ZMP model first-order parameters, λ and δ_d;
2) the force distribution first-order parameters, λ_c and λ_f;
3) the admittance gain K_a for CoP tasks and the FFDC gain K_z.

These parameters are interrelated and are tuned by fitting the modeled CoP, vertical forces, or ZMP to the measured ones. Specifically, λ represents the model's first-order response, and increasing K_a fastens the application of the wrench, thereby increasing λ. Similarly, K_a affects vertical force control. We set K_a to its lowest value in order to estimate λ.

At present, the tuning process for these parameters remains empirical; however, for transparency, we provide an ad hoc guideline for the tuning procedure used in our experiments.

[1][Online]. Available: https://jrl-umi3218.github.io/mc_rtc/

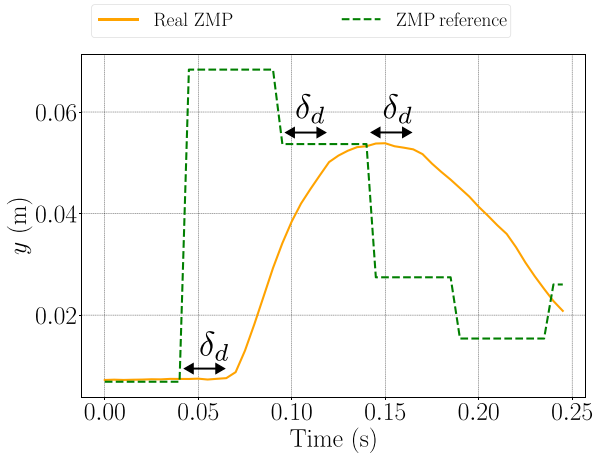

Fig. 13. Delay between the real ZMP and the sent reference on HRP-2KAI.

TABLE I
CHOSEN PARAMETERS FOR EACH ROBOT

	ZMP model		FD model		FFDC	CoP	
Parameters	λ	δ_d	λ_f	λ_c	K_z	K_a	δ
Unit	[s^{-1}]	[s]	[s^{-1}]	[s^{-1}]	[N.s.m^{-1}]	[N.s.m^{-1}]	[ms]
HRP-2KAI	20	0.025	20	20	0.0001	0.02	4
HRP-4CR	5	0.025	10	5	0.0001	0.01	5
HRP-4	7	0.035	15	15	0.0001	0.01	5
HRP-5P	15	0.025	15	15	0.0001	0.01	5
RHPS1	8	0.025	8	10	0.0001	0.01	2

1) Set $K_a = 0.01$, $K_z = 10^{-4}$, and $\lambda = \lambda_f = \lambda_c = 40$, and make the robot step in place.
2) δ_d can be identified from the transition into the walking phase (see Fig. 13).
3) Adjust λ to ensure that the model ZMP aligns well with the measured ZMP in single support.
4) Ensure that the feet in single support do not vibrate; otherwise, increase K_a (and λ).
5) Adjust λ_z such that the model vertical force matches the measured one in double support.
6) Vibration in double support is typically due to vertical force control; adjust K_z and λ_f to ensure smooth and complete transition of the ZMP to the future support foot.
7) Set λ_c equal to λ and adjust to refine the ZMP trajectory in double support.
8) Fine-tune further by having the robot step forward and backward.

The parameters for each humanoid are provided in Table I, which includes the sampling period of the whole-body control δ at which logged data and computed joint angles are transmitted.

The other parts of the control scheme are updated at a sampling period Δt for all robots. Parameters common to all humanoids are shown in Table II.

The IS-MPC weights that handle the DCM trajectory, β_u, and DCM velocity, $\beta_{\dot{u}}$, are set to a nonzero value only when the robot is in standing phase and to zero once the robot switches to a walking phase. This is because those weights affect the compliance of the robot during disturbances and make it less likely

TABLE II
CHOSEN PARAMETERS FOR THE MPC

Δt	β_z	$\beta_{\dot{z}}$	β_u	$\beta_{\dot{u}}$	β_f	T_c	T_p
0.05 s	10	0.001	50	2	1000	1.5 s	10 s

TABLE III
COMPUTATION TIME FOR EACH MODULE ON RHPS1

Feasibility Solver	IS-MPC	Force Distribution	Whole-Body Control
0.5 ms	1.5 ms	0.2 ms	0.4 ms

TABLE IV
TASKS PARAMETERS

Tasks	CoM	CoP	Swing Foot	Chest	Posture
Weight	10000	10^6	5000	200	10
K_p	100	1	200	50	1
K_d	$2\sqrt{K_p}$	150	$2\sqrt{K_p}$	$2\sqrt{K_p}$	$2\sqrt{K_p}$

to trigger steps. The computation performances on one robot are shown in Table III; they may slightly fluctuate depending on the onboard hardware.

The whole-body QP cost function consists of a set of weighted tasks, where the decision variables are the accelerations of the joint angles and floating base [55], [56]. The tasks include:

1) position, velocity, and acceleration of the CoM;
2) contact forces (CoP task);
3) chest orientation relative to foot orientations;
4) swing foot position speed and acceleration during single-support phases;
5) a low-gain posture task to address redundancy and singularities.

These tasks are weighted and their parameters are listed in Table IV. The transition from the swing foot task to the CoP task occurs when the measured force on the swing foot exceeds a certain threshold.

The experiments demonstrate the robustness of the proposed method by enabling the robots to walk long distances, traverse uneven terrains, and recover from disturbances such as operator pushes. We also tested perturbations during the standing phase to trigger stepping recovery.

All plots are displayed in a coordinate frame linked to the robot's floating base, where $\mathbf{z}$ is the vertical axis, $(\mathbf{z}, \mathbf{y})$ is the coronal plane, and $(\mathbf{z}, \mathbf{x})$ is the sagittal plane.

While we aimed to maintain consistent walking conditions across robots, practical limitations sometimes made this challenging. For example, the HRP-4 is located in France, while the other robots are based in Japan, where campus experiments are subject to stricter regulations. The HRP-4CR was tested indoors on a rigid flat floor due to mechanical constraints, as reported in [54]. Multimedia material accompanying this article provides additional context to complement experimental data.

A. Force Control and Model Evaluation

The force control model is evaluated across all the humanoid robots listed. As shown in Fig. 14, the first-order dynamic model of the ZMP with a delay offers a reliable estimation of the real ZMP dynamics. To highlight our objective of modeling ZMP behavior, we adjusted the parameter λ to allow the HRP-2KAI, HRP-5P, and RHPS1 robots to walk on a compliant mattress. Despite the slow response time of the model, the control remains stable, and the robots are able to maintain balance while performing dynamic walking tasks (see Section VII-D).

Fig. 9 illustrates how the measured disturbance from the swing foot is used to adjust the pendulum trajectory during walking. The figure shows the measured κ (defined in Section V-B), which captures the vertical component of the disturbance. Finally, Fig. 15 demonstrates that the vertical forces during the double-support phase align with the expected model. When the model value is constant, the robot is in single support, and vertical force control is no longer active.

The force measurements on the HRP-4 are less accurate due to crosstalk between the measured torque and the vertical forces, which affects the ZMP calculation during the double-support phase. This coupling has been identified and corrected in single support through a corrective plug-in.

B. Footstep and Step Timing Adaptation

Footstep and step timing adaptation are evaluated across all the humanoid robots. Fig. 16(a) illustrates the recovery process following a front push and how the footstep planning is updated in response. In most cases, increasing the step frequency or stepping in the direction of the disturbance helps the robot regain balance. However, in some configurations, the optimal strategy is to extend the duration of the current or next step, as there may be no appropriate step available to counterbalance the disturbance. This is evident in Fig. 16(b), where a lateral disturbance leads to an increased double-support duration, allowing the DCM to reach the next support foot.

C. Static Push Recovery

Each humanoid was subjected to a push during the standing phase to assess the conditions defined in Section IV-E and trigger a recovery step. The size of the ZMP region is intentionally set smaller than the allowed limits, ensuring that a recovery step is activated even when the robot is well balanced. The choice of the supporting foot is determined by the position of the DCM and the contact configuration at the moment of the disturbance. Depending on which side of the balance region is violated, if the disturbance is frontal, the support foot is selected as the one closest to the DCM. Conversely, for nonfrontal disturbances, the support foot is chosen as the one furthest from the DCM. If a recovery step is triggered, no specific footstep plan is predefined. Instead, the recovery steps are entirely determined based on the feasibility region. The effectiveness of this push recovery strategy is primarily limited by the intensity of the perturbation. If the required recovery steps are deemed infeasible, the robot will inevitably fall. In such cases, it becomes necessary to

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

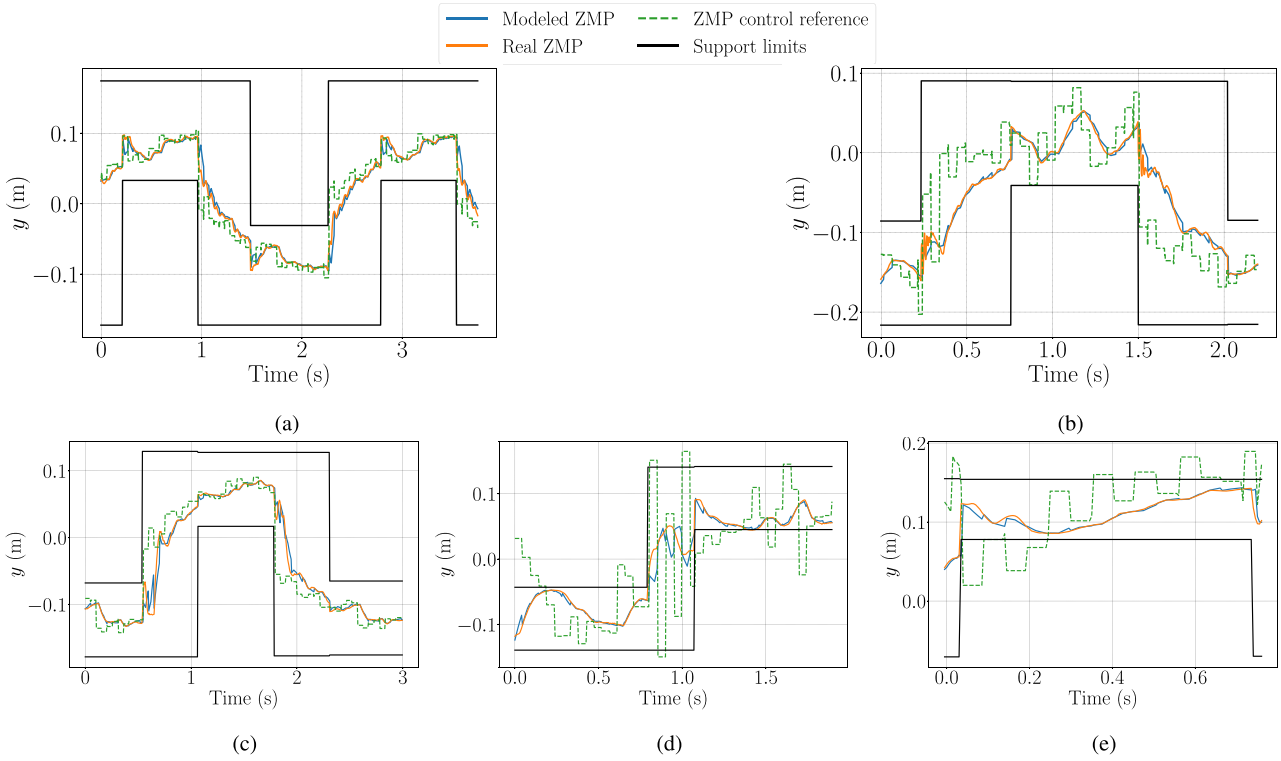

Fig. 14. ZMP model compared with robots' measured ZMP when walking on rigid flat floors; model state is updated at a sampling rate Δt. (a) On HRP-2KAI. (b) On RHPS-1. (c) On HRP-5P. (d) On HRP-4. (e) On HRP-4CR.

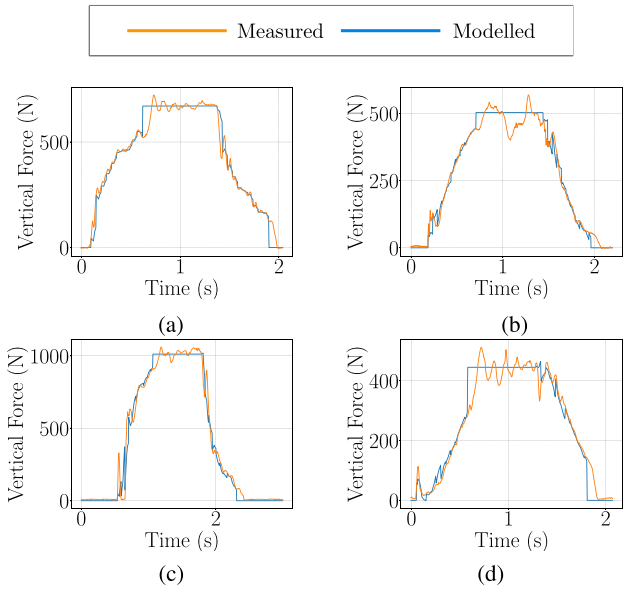

Fig. 15. Comparison between the vertical force model and real measurements (left foot) of the robots walking on rigid flat ground. The model state is updated at sampling rate Δt. (a) On HRP-2KAI. (b) On RHPS-1. (c) On HRP-5P. (d) On HRP-4CR.

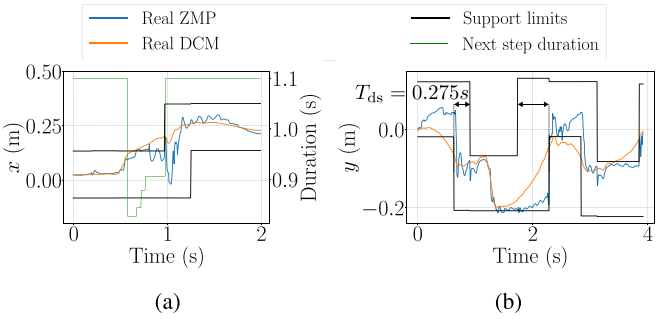

Fig. 16. Step duration planning is updated differently based on the direction of the disturbance (these behaviors apply across all terrain types. (a) Forward push. (b) Lateral push.

introduce a variation in the robot's angular momentum or alter the height of the CoM (i.e., apply vertical acceleration to the CoM) in addition to the existing recovery strategy. In either scenario, the LIPM assumptions no longer hold.

D. Walking on Soft Ground

This experiment was conducted with HRP-2KAI, HRP-5P, and RHPS1. Walking on compliant surfaces poses a significant challenge for bipedal robots, requiring a detailed understanding of how forces behave in relation to control inputs. For example, without the active control scheme, HRP-2KAI could not remain standing on the mattress, even with the CoM initially over the support region. A successful demonstration of locomotion under such conditions has been previously achieved using a torque-controlled humanoid robot [57], where contact compliance and feedforward force control played a crucial role in adapting to the soft terrain.

Our approach, which models the contact force responses, necessitates adjusting the parameters that govern force behavior to

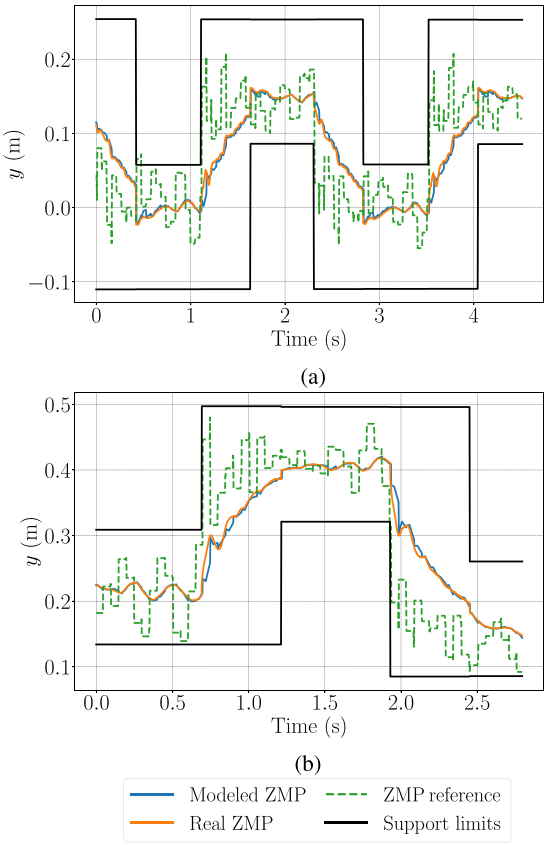

Fig. 17. ZMP model compared with robots' measured ZMP when walking on soft grounds; the model state is updated at sampling rate Δt. (a) On HRP-2KAI, $\lambda = 4$. (b) On RHPS1, $\lambda = 5$.

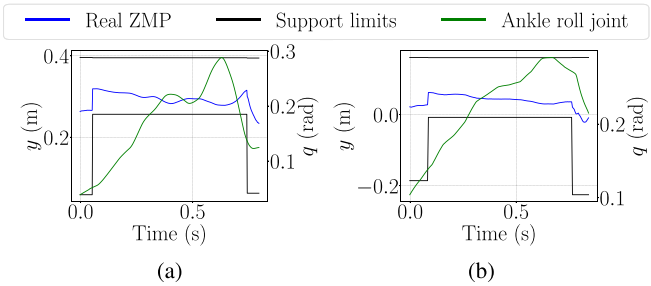

Fig. 18. On similar compliant ground with similar walking, feet spring damper reduces oscillation in single support. (a) On RHPS1. (b) On HRP-2KAI.

ensure stable walking on compliant surfaces. We ensured that the deformation of the mattress during walking never exceeded its limits, even during single- or double-support phases, to prevent the support from transitioning to rigid ground conditions due to excessive compression. The evaluation of the ZMP model is shown in Fig. 17. As discussed in Section VII-A, walking on soft ground required tuning the ZMP model parameter λ. In addition, it was necessary to adjust the FFDC gain K_z to enhance vertical force control. This adjustment was made specifically for HRP-2KAI, where K_z was increased from 0.0001 to 0.0002.

All the humanoids used in the experiments on soft ground were position controlled. However, HRP-2KAI is equipped with

Fig. 19. HRP-2KAI and HRP-4 robots walking outdoors. Experiments with the HRP-4 are done without safety ropes.

a shock-absorbing mechanism consisting of a rubber bush between the foot and ankle actuators. This mechanical difference results in smoother foot motion in single support for HRP-2KAI compared to RHPS1, which has a rigid connection between the ankles and feet. As seen in Fig. 18, under similar conditions (environment, stepping frequency, step length, and ZMP trajectory), RHPS1 exhibits more pronounced foot oscillations, while HRP-2KAI's feet motion is notably smoother.

E. Walking Outdoors

Outdoor experiments underscore the advantages of compliance combined with force control, conducted with HRP-4 and HRP-2KAI, as shown in Fig. 19. Even seemingly flat outdoor terrains exhibit nonnegligible local roughness. In addition, some of the terrains the robots traversed display pseudo-compliant behavior, such as grass or soft ground, providing an opportunity to assess the adaptability of the force behavior model to varying environmental conditions.

The evaluation of the ZMP model on these terrains is presented in Fig. 20. Supplementary multimedia material demonstrates the feet's compliance with the terrain during walking. These terrains also feature uneven surfaces with varying local slopes. When walking uphill, the robot's motion is minimally affected, as early contacts can be detected via force measurements. In contrast, when walking downhill, the uncertainty in contact location requires the leg to continue lowering until contact is detected, which can potentially lead to disturbances. Our force control scheme can effectively handle this scenario, ensuring that the foot continues moving downward until force feedback confirms contact. Alternatively, the FFDC can be employed if the actual contact remains close to the foot. To optimize this approach, the FFDC K_z gain was increased from 0.0001 to 0.00015 on HRP-4 for these experiments. In the HRP-2KAI experiments, strong wind induced additional perturbations, while for HRP-4, significant disturbances occurred when stepping on thick roots.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

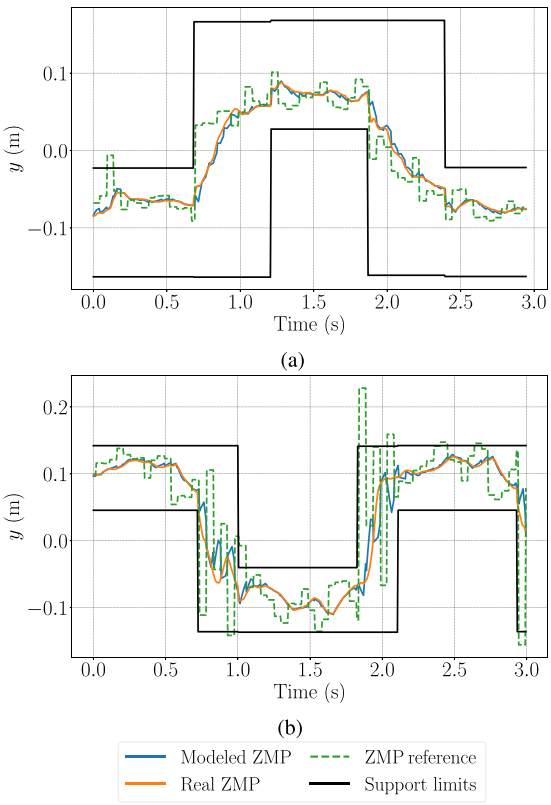

Fig. 20. ZMP model compared to the real robots ZMP walking outdoors; the model state is updated at a sampling rate Δt. (a) On HRP-2KAI, $\lambda = 12$. (b) On HRP-4, $\lambda = 11$.

In practice, the primary cause of failures during these tests was poor contact detection.

VIII. Conclusion

In this article, we presented a new walking control scheme for bipedal robots. A key innovation of this approach is the use of an LIPM-based MPC in a closed-loop configuration, which operates directly on the robot's pendulum state (CoM, CoM velocity, and ZMP) without requiring an additional stabilizer module. In addition, the controller dynamically adjusts step location and duration and distributes the contact wrench resulting from MPC computations.

To demonstrate the versatility and robustness of our control scheme, we tested it on five different humanoid robots across various terrains and scenarios. Our goal was to minimize the number of tunable parameters and provide clear guidelines for their adjustment. While these parameters are primarily robot dependent, they also exhibit a dependence on the mechanical properties of the ground. Our experiments revealed that the *robustness* and *plug-and-walk* functionality of the control could be further enhanced by estimating some of these parameters online or during a calibration phase. This is an area for future work, and we believe it is feasible, as many of these parameters are observable. Furthermore, while we modeled the floor dynamics using a first-order spring–damping model for ZMP, the general behavior of the system reflects a second-order dynamic, requiring a more refined estimation of ZMP velocity.

We also tested the HRP-4 robot walking on river pebbles outdoors, but the controller failed in all trials. This failure was partly due to the flat rigid soles and the inadequate handling of momentum during walking. In future work, we plan to extend our control policy to incorporate full centroidal dynamics, including variable CoM height and angular momentum, as described in previous studies [31], [39]. In addition, we are working on redesigning the robot's soles based on a recent tactile technology [58], inspired by previous work on compliant materials for better ground adaptation [59].

Understanding how these challenges are interconnected is crucial to achieve the *plug-and-walk* goal. Finally, as mentioned in Section II, we are also exploring a machine-learning-based approach to walking, believing that a robust solution will emerge from a hybrid approach that leverages the strengths of both model-based and data-driven methods.

Appendix
Proof of Theorem 1

We suppose that $\bar{N}$ and $O(t)$ are $N_v \times 2$ matrix and $N_v \times 1$ row vector, respectively, such that the ZMP constraints is an N_v vertices convex polygon (with possibly multiple vertices at the same position) $\forall t \in [t^0; \infty]$. Let us assume that $\bar{N}$ and $O(t)$ are ordered such that, at any time t, two consecutive constraints intersect at one vertex of the polygon (the first constraint is considered as the successor of the last one).

To prove the necessity, let $\bar{n}_x^i$ and $\bar{n}_y^i$ be the first and second element of the ith row of $\bar{N}$ respectively, and $o_i(t)$ the value of the ith row of $O(t)$. The ZMP constraint can be written for each row i as

$$\bar{n}_x^i x_z(t) + \bar{n}_y^i y_z(t) \leq o^i(t). \tag{60}$$

Multiplying each side by $e^{-\omega(t-t^0)}$ and integrating on $[t^0; \infty]$, we obtain

$$\bar{n}_x^i \int_{t^0}^{\infty} x_z(\tau) e^{-\omega(\tau - t^0)} d\tau + \bar{n}_y^i \int_{t^0}^{\infty} y_z(\tau) e^{-\omega(\tau - t^0)} d\tau \leq \int_{t^0}^{\infty} o^i(\tau) e^{-\omega(\tau - t^0)} d\tau. \tag{61}$$

Using the definition of the stability condition (4), we have

$$\bar{n}_x^i x_u^0 + \bar{n}_y^i y_u^0 \leq \int_{t^0}^{\infty} o^i(\tau) e^{-\omega(\tau - t^0)} d\tau. \tag{62}$$

Since this result holds for each row i, the last equation can be rewritten as in (41), which proves the necessity.

Let $\mathbf{v}_z^j(t)$ be the 2-D coordinate of the jth vertex of the ZMP polygon. Let $\bar{N}_j$ and $O_z^j(t)$ be the 2×2 square matrix and the 2×1 row vector representing two consecutive constraints crossing $\mathbf{v}_z^j(t)$. $\bar{N}_j$ originates from two rows of $\bar{N}$ and $O_z^j(t)$ from two terms of $O(t)$. We can write

$$\mathbf{v}_z^j(t) = (\bar{N}^j)^{-1} O_z^j(t). \tag{63}$$

Let O_u be [from (41)]

$$O_u = \omega \int_{t^0}^{\infty} O(\tau) e^{-\omega(\tau - t^0)}\, d\tau \tag{64}$$

and let E be the convex set of initial DCM positions $\mathbf{p}_u^0$ such that (41) is true, i.e., $\bar{N}\mathbf{p}_u^0 \leq O_u$, and Z the set of the ZMP trajectories such that $\forall t \in [t^0; \infty]$, $\bar{N}\mathbf{p}_z(t) \leq O(t)$

Lemma: Z is a convex set.

Proof: let $\mathbf{p}_{z,1}$ and $\mathbf{p}_{z,2}$ two elements of Z and $\gamma \in [0; 1]$.

$$\begin{aligned} &\forall t \in [t^0; \infty],\ \bar{N}(\gamma \mathbf{p}_{z,1}(t) + (1-\gamma)\mathbf{p}_{z,2}(t)) \\ &\quad \leq \gamma O(t) + (1-\gamma) O(t) \end{aligned} \tag{65}$$

Any convex combination of $\mathbf{p}_{z,1}$ and $\mathbf{p}_{z,2}$ belongs to Z, hence the convexity. □

We extract O_u^j from O_u similarly to the definition of O_z^j and compute $\mathbf{v}_u^j$, the vertices of the convex polygon representing E

$$\mathbf{v}_u^j = (\bar{N}^j)^{-1} O_{u,j} \tag{66}$$

$$\mathbf{v}_u^j = \omega \int_{t^0}^{\infty} \bar{N}_j^{-1} O_z^j(\tau) e^{-\omega(\tau - t^0)}\, d\tau \tag{67}$$

$$\mathbf{v}_u^j = \omega \int_{t^0}^{\infty} \mathbf{v}_z^j(\tau) e^{-\omega(\tau - t^0)}\, d\tau. \tag{68}$$

To prove sufficiency, let $\mathbf{p}_u^0 \in E$. Since E is convex, $\mathbf{p}_u^0$ is a convex combination of the feasibility region vertices such that

$$\mathbf{p}_u^0 = \sum_{j=1}^{N_v} \gamma^j \mathbf{v}_u^j \tag{69}$$

where $\forall j \in [\![1; N_v]\!]$, $\gamma^j \in [0, 1]$ and $\sum_{j=1}^{N_v} \gamma^j = 1$. We must find a ZMP trajectory $\mathbf{p}_z(t) \in Z$ such that the initial DCM that respect the stability condition in (4) is $\mathbf{p}_u^0$.

Consider the following ZMP trajectory $\forall t \in [t^0; \infty]$:

$$\mathbf{p}_z(t) = \sum_{j=1}^{N_v} \gamma^j \mathbf{v}_z^j(t). \tag{70}$$

This trajectory satisfies the ZMP constraints as it is a convex combination of the vertices of each ZMP constraint polygon. To check if $\mathbf{p}_u^0$ satisfies the stability condition with the ZMP trajectory $\mathbf{p}_z(t)$, we use (4) with (70)

$$\begin{aligned} &\omega \int_{t^0}^{\infty} e^{\omega(\tau - t^0)} \sum_{j=1}^{N_v} \gamma^j \mathbf{v}_z^j(\tau) d\tau \\ &= \omega \sum_{j=1}^{N_v} \gamma^j \int_{t^0}^{\infty} e^{\omega(\tau - t^0)} \mathbf{v}_z^j(\tau) d\tau. \end{aligned} \tag{71}$$

Using (68) and (69), we obtain

$$\sum_{j=1}^{N_v} \gamma^j \omega \int_{t^0}^{\infty} e^{\omega(\tau - t^0)} \mathbf{v}_z^j(\tau) d\tau = \sum_{j=1}^{N_v} \gamma^j \mathbf{v}_u^j = \mathbf{p}_u^0 \tag{72}$$

which proves sufficiency, hence the equivalence. Q.E.D.

REFERENCES

[1] 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.*, 2001, pp. 239–246.

[2] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot-1st report: Walking gait pattern generation-," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St Louis, MO, USA, 2009, pp. 1084–1091.

[3] T. Takenaka, T. Matsumoto, T. Yoshiike, and S. Shirokura, "Real time motion generation and control for biped robot-2nd report: Running gait pattern generation-," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St Louis, MO, USA, 2009, pp. 1092–1099.

[4] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot-3rd report: Dynamics error compensation-," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St Louis, MO, USA, 2009, pp. 1594–1600.

[5] T. Takenaka et al., "Real time motion generation and control for biped robot-4th report: Integrated balance control-," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St Louis, MO, USA, 2009, pp. 1601–1608.

[6] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*. Berlin, Germany: Springer, 2014.

[7] N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, "MPC for humanoid gait generation: Stability and feasibility," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1171–1188, Aug. 2020.

[8] F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "Feasibility-driven step timing adaptation for robust MPC-based gait generation in humanoids," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1582–1589, Apr. 2021.

[9] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the HRP-4 humanoid robot using whole-body admittance control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 277–283.

[10] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016, pp. 1203–1234.

[11] J. Carpentier and P.-B. Wieber, "Recent progress in legged robots locomotion control," *Curr. Robot. Reports*, vol. 2, no. 3, pp. 231–238, 2021.

[12] Y. Gong and J. W. Grizzle, "Zero dynamics, pendulum models, and angular momentum in feedback control of bipedal locomotion," *J. Dyn. Syst. Meas. Control*, vol. 144, no. 12, Oct. 2022, Art. no. 121006.

[13] M. S. Khan and R. K. Mandava, "A review on gait generation of the biped robot on various terrains," *Robotica*, vol. 41, no. 6, pp. 1888–1930, 2023.

[14] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Boca Raton, FL, USA: CRC Press, Oct. 2018.

[15] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

[16] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "BiConMP: A nonlinear model predictive control framework for whole body motion planning," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 905–922, Apr. 2023.

[17] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2261–2268, Jul. 2018.

[18] K. Yin, K. Loken, and M. van De Panne, "SIMBICON: Simple biped locomotion control," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 105–114, Jul. 2007.

[19] S. Coros, P. Beaudoin, and M. van De Panne, "Generalized biped walking control," *ACM Trans. Graph.*, vol. 29, no. 4, Jul. 2010, Art. no. 130.

[20] D. Holden, T. Komura, and J. Saito, "Phase-functioned neural networks for character control," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017, Art. no. 42.

[21] X. B. Peng, P. Abbeel, S. Levine, and M. van De Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 143:1–143:14, Jul. 2018.

[22] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *Int. J. Robot. Res.*, vol. 32, nos. 9/10, pp. 1104–1119, 2013.

[23] E. Dantec et al., "Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2022, pp. 638–644.

[24] Z. Li et al., "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 2811–2817.

[25] H. Duan et al., "Sim-to-real learning of footstep-constrained bipedal dynamic walking," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 10428–10434.
[26] J. Hwangbo et al., "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaau5872.
[27] S. Kajita et al., "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, 2003, pp. 1620–1626.
[28] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 355–368, Apr. 2015.
[29] S. Caron, "Biped stabilization by linear feedback of the variable-height inverted pendulum model," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 9782–9788.
[30] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3D-slip model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 5134–5140.
[31] K. Guan, K. Yamamoto, and Y. Nakamura, "Virtual-mass-ellipsoid inverted pendulum model and its applications to 3D bipedal locomotion on uneven terrains," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1401–1406.
[32] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online non-linear centroidal MPC for humanoid robot locomotion with step adjustment," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 10412–10419.
[33] M. Murooka, M. Morisawa, and F. Kanehiro, "Centroidal trajectory generation and stabilization based on preview control for humanoid multicontact motion," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8225–8232, Jul. 2022.
[34] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 137–142.
[35] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Adv. Robot.*, vol. 24, nos. 5/6, pp. 719–737, 2010.
[36] R. J. Griffin and A. Leonessa, "Model predictive control for dynamic footstep adjustment using the divergent component of motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1763–1768.
[37] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Walking control based on step timing adaptation," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 629–643, Jun. 2020.
[38] L. Lanari, S. Hutchinson, and L. Marchionni, "Boundedness issues in planning of locomotion trajectories for biped robots," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 951–958.
[39] F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "From walking to running: 3D humanoid gait generation via MPC," *Front. Robot. AI*, vol. 9, 2022, Art. no. 876613.
[40] M. Morisawa, S. Kajita, F. Kanehiro, K. Kaneko, K. Miura, and K. Yokoi, "Balance control based on capture point error compensation for biped walking on uneven terrain," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2012, pp. 734–740.
[41] M. Murooka et al., "Humanoid loco-manipulations pattern generation and stabilization control," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5597–5604, Jul. 2021.
[42] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic programming for multirobot and task-space force control," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 64–77, Feb. 2019.
[43] S. Kajita et al., "Biped walking stabilization based on linear inverted pendulum tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 4489–4496.
[44] S. Kajita et al., "Vertical vibration suppression for a position controlled biped robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1637–1642.
[45] M. Benallegue and F. Lamiraux, "Estimation and stabilization of humanoid flexibility deformation using only inertial measurement units and contact information," *Int. J. Humanoid Robot.*, vol. 12, no. 03, 2015, Art. no. 1550025.
[46] C. Knauer, L. Schlipf, J. M. Schmidt, and H. R. Tiwary, "Largest inscribed rectangles in convex polygons," *J. Discrete Algorithms*, vol. 13, pp. 78–85, 2012.
[47] E. G. Birgin and J. M. Martínez, "Block coordinate descent for smooth nonconvex constrained minimization," *Comput. Optim. Appl.*, vol. 83, no. 1, pp. 1–27, Sep. 2022.
[48] S. Brossette and P.-B. Wieber, "Collision avoidance based on separating planes for feet trajectory generation," in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2017, pp. 509–514.
[49] A. S. Habib, F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "Handling non-convex constraints in MPC-based humanoid gait generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 13167–13173.
[50] S. Nakaoka, "Choreonoid: Extensible virtual robot environment built on an integrated GUI framework," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, 2012, pp. 79–85.
[51] M. Benallegue et al., "Humanoid robot RHP friends: Seamless combination of autonomous and teleoperated tasks in a nursing context," *IEEE Robot. Autom. Mag.*, vol. 32, no. 1, pp. 79–90, Mar. 2025.
[52] I. Kumagai et al., "Toward industrialization of humanoid robots: Autonomous plasterboard installation to improve safety and efficiency," *IEEE Robot. Autom. Mag.*, vol. 26, no. 4, pp. 20–29, Dec. 2019.
[53] A. Kheddar et al., "Humanoid robots in aircraft manufacturing: The airbus use-cases," *IEEE Robot. Autom. Mag.*, vol. 26, no. 4, pp. 30–45, Dec. 2019.
[54] R. Cisneros-Limón et al., "A cybernetic avatar system to embody human telepresence for connectivity, exploration, and skill transfer," *Int. J. Social Robot.*, vol. 17, pp. 535–562, Jan. 2024.
[55] M. Djeha, P. Gergondet, and A. Kheddar, "Robust task-space quadratic programming for kinematic-controlled robots," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3857–3874, Oct. 2023.
[56] K. Bouyarmane and A. Kheddar, "On weight-prioritized multi-task control of humanoid robots," *IEEE Trans. Autom. Control*, vol. 63, no. 6, pp. 1632–1647, Jun. 2018.
[57] G. Mesesan, J. Englsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer, "Dynamic walking on compliant and uneven terrain using DCM and passivity-based whole-body control," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2019, pp. 25–32.
[58] Y. Yan, A. Zermane, J. Pan, and A. Kheddar, "A soft skin with self-decoupled three-axis force-sensing taxels," *Nat. Mach. Intell.*, vol. 6, no. 11, pp. 1284–129, 2024.
[59] A. Pajon, S. Caron, G. De Magistri, S. Miossec, and A. Kheddar, "Walking on gravel with soft soles using linear inverted pendulum tracking and reaction force distribution," in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2017, pp. 432–437.

Antonin Dallard received the Engineering Master degree in mechanical and industrial engineering from the Arts Et Métiers Institute of Technology, Paris, France, in 2020 and 2024, and the Ph.D. degree in humanoid robot teleoperation and locomotion from the CNRS-University of Montpellier, LIRMM, Montpellier, France, and the CNRS-AIST Joint Robotics Laboratory, Tsukuba, Japan, in 2024.

He is currently an R&D Control Engineer with Wandercraft, Paris.

Mehdi Benallegue received the Engineering degree in computer science from the National Institute of Computer Science, Alger, Algeria, in 2007, and the master's degree from the University of Paris 7, Paris, France, in 2008, and the Ph.D. degree in robotics from the University of Montpellier, Montpellier, France, in 2011.

He then joined the Franco–Japanese Robotics Laboratory, Tsukuba, Japan, and INRIA Grenoble, Montbonnot-Saint-Martin, France. He was also a Postdoctoral Researcher with the Collège de France, Paris, and LAAS-CNRS, Toulouse, France. He is currently a Research Associate with the CNRS-AIST Joint Robotics Laboratory, Tsukuba, Japan. His research interests include robot estimation and control, legged locomotion, biomechanics, neuroscience, and computational geometry.

Nicola Scianca received the Ph.D. degree in control engineering from the Sapienza University of Rome, Rome, Italy, in 2020.

In 2019, he was a Visiting Student with the Model Predictive Control Laboratory, University of California at Berkeley, Berkeley, CA, USA. He is currently a Researcher with the Department of Computer, Control and Management Engineering, Sapienza University of Rome. His research interests include model-predictive control for humanoid robots.

Fumio Kanehiro received the B.E., M.E., and Ph.D. degrees in engineering from the University of Tokyo, Tokyo, Japan, in 1994, 1996, and 1999, respectively.

In 2000, he joined the Electrotechnical Laboratory, Agency of Industrial Science and Technology, Ministry of Industrial Science and Technology, later reorganized as the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. Since 2007, he was a Visiting Researcher with LAAS-CNRS, Toulouse, France, for one year and three months. He is currently the Director of the CNRS-AIST Joint Robotics Laboratory, International Research Laboratory, AIST, Tsukuba. His research interests include the software frameworks and whole body motion planning of humanoid robots.

Dr. Kanehiro was a Research Fellow of the Japan Society for the Promotion of Science from 1998 to 1999.

Abderrahmane Kheddar (Fellow, IEEE) received the B.S. degree in computer science from the Institut National d'Informatique, Alger, Algiers, and the M.Sc. and Ph.D. degree in robotics from Pierre et Marie Curie, Sorbonne University, Paris, France, in 1990, 1993, and 1997, respectively.

He is currently the Directeur de Recherche with the CNRS-University of Montpellier, Montpellier, France. His research interests include haptics and humanoids at large.

Dr. Kheddar is an Editor for IEEE ROBOTICS AND AUTOMATION LETTER, an Editor for IEEE TRANSACTIONS ON ROBOTICS (2013–2018), and a Founding Member of IEEE TRANSACTIONS ON HAPTICS. He is a Fellow of the Asia-Pacific Artificial Intelligence Association and a Member of the National Academy of Technology of France.