Robust Bipedal Walking with Closed-Loop MPC: Adios Stabilizers

Antonin Dallard, Mehdi Benallegue, Nicola Scianca, Fumio Kanehiro, Abderrahmane Kheddar

Robust Bipedal Walking with Closed-Loop MPC– Adios Stabilizers

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

***Abstract*—We present a new walking control scheme based on the dynamics of the inverted pendulum. Our scheme includes re-planning the step locations and step timings, feet force control, and a walking pattern generation that is closed-loop thanks to feedback in the state of the real humanoid robot pendulum (CoM position/speed and ZMP). No additional control policy is used to maintain the static and dynamic balance of the humanoid. We experimented this framework on five different humanoid robots over multiple disturbances including sudden pushes during walking or in a static state and by achieving locomotion over uneven and compliant grounds.**

I. Introduction

WALKING control strategies and algorithms for bipedal and humanoid robots are various (see Sec. II). The complexity of this problem lies in the interlink between different bricks consisting of (i) reactive footstep planning, (ii) whole-body switched control under balance constraints, (iii) stability (as understood in control theory) of the entire control, and finally (iv) perception to understand the surrounding environment and relative bipedal robot state.

In this paper, we are interested in a particular class of walking control strategies: the use of linear inverted pendulum mode –or model, acronym LIPM, early introduced in [1] and popularized by the first Honda humanoids family (see knowledge reports in [2], [3], [4], [5]). It is still successfully implemented in many humanoid use cases, yet it has evolved in numerous shades and sophistication (Sec. II).

The common implementation of such models is to reason only on the center of mass (CoM) dynamics in order to draw fast planning and control. The latter usually implies the exploitation of the zero moment point (ZMP) location for dynamic balancing. Indeed, the ZMP gives a limitation on the contact forces that can be applied; namely, the ZMP always resides inside the support polygon of the robot. In linear cases, the ZMP can be seen either as the causality output (also known as the Cart Table model [6]) or as input (the LIPM) for a linear dynamic system linking it to the CoM. However, the instability of the model entails that for a generic ZMP trajectory, the associated CoM trajectory might be divergent.

Fig. 1: HRP-2KAI and HRP-4 robots walking outdoors. Experiments with the HRP-4 are done without safety ropes.

The Intrinsically Stable Model Preview Control (IS-MPC) [7] is a walking gait generator, derived from the LIPM, which incorporates a stability constraint explicitly: The presence of this constraint proves recursive feasibility. This means that it is always possible to find a solution satisfying the constraints and controller stability, i.e., the CoM trajectory is always bounded with respect to the ZMP. The IS-MPC can update the footsteps location and has recently improved to adapt also the next step duration [8].

However, in essence, reduced models cannot grasp the complexity of the whole multi-body dynamics. Therefore, behavioral discrepancies are expected to occur between reduced models and the whole-body dynamics. Also, the inverted pendulum is a model derived from the constrained equation of the centroidal dynamics; controlling the pendulum means controlling the contact forces applied to the robot. Different grounds lead to different behaviors for the same controller (see Sec. II). To deal with these discrepancies, one shall devise a correction policy that mitigates the modeling gaps in using the real robot state, and a model of the contact forces

Manuscript received June 30, 2023; revised February 29, 2024; accepted Xxxxxx XX, 20XX. Date of publication Xxxxxxxx X, 20XX; date of current version Xxxxxxxxx X, 20XX. This paper was recommended for publication by Associate Editor X. Xxxxxxxx and Editor K. Mombaur upon evaluation of the reviewers comments.

A. Dallard, M. Benallegue, F. Kanehiro and A. Kheddar are with CNRS-AIST Joint Robotics Laboratory, IRL3218, Tsukuba, Japan.

A. Dallard and A. Kheddar are also with the CNRS-University of Montpellier, LIRMM, UMR5506, Montpellier, France.

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

This paper has supplementary video downloadable material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 00.0000/XXX.202X.0000000

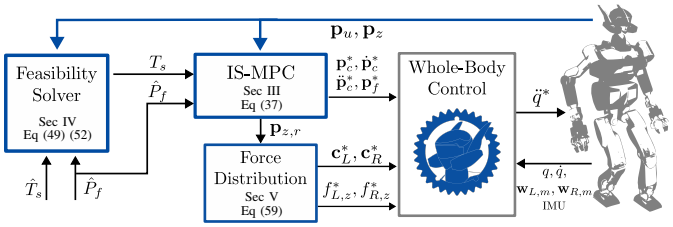


Fig. 2: Global control scheme, the $*$ superscript represents the control references for the whole body control or the robot. The swing foot trajectory planning or the force control scheme is included in the whole-body control box as they are outside the article's contribution. The blue elements highlight our adding w.r.t existing approaches.

behavior, see e.g., [9]. Such mitigation is commonly referred to as the *stabilizer* in the humanoid research jargon (yet not all papers name it so). The *stabilizer* usually comes with several parameters. Tuning it is not trivial, especially when combined with the controller's parameters. This correction policy is short-sighted compared to the MPC policy which is based on state prediction, and might invalidate some benefits of having such prediction, this is well detailed in Sec. III-A.

One might think that MPC should take care of this issue thanks to its inherent robustness. However, as in any controller, the robustness of MPC comes from its closed-loop, i.e., the predicted trajectory is recomputed at each time starting from the measured state. Hence, if the contact force model is not taken into account, the generated walking gait would still be not feasible; hence, a *stabilizer* would still be necessary. This paper proposes a solution to this issue and presents a humanoid walking control scheme with the following newies:

1) A reformulation of the IS-MPC with a model of the contact forces dynamics that closes the loop effectively on the robot state (CoM, CoM velocity and ZMP); which does not need an external correctional policy (the so-called *stabilizer*) to enforce and warrant balance, Sec. III. This is the main contribution;
2) A LIPM feasibility solver based on extended stability constraint to adapt any plan of footsteps location and timing on the entire *horizon* using the state of the robot, Sec. IV;
3) A force distribution scheme to plan the contact forces during double-support phases, Sec. V;
4) Extensive and challenging scenarios using five different humanoid robots, all of them controlled with the same software to assess and illustrate the robustness, versatility and portability of our algorithms, Secs. VI and VII.

Figure 2 highlights our contributions, the order of execution, sections and main equations formulating each subproblem.

II. Background

Bipedal and humanoid locomotion is a rich field of research in the robotics domain. We invite interested readers to refer to the excellent reviews (most of which are recent) in [10], [11], [12], [13], and the monographs [14], [6].

We put aside all the work done in passive bipedal locomotion [15], as we are interested in motorized humanoid robots that do not content in predefined cycled locomotions[1]. We also do not cover research in quadruped or hexapods locomotion; even if there are similarities in some fundamental aspects, see e.g., [16], [17], multiple legs have richer locomotion gait patterns and specificities (e.g., in terms of balance and mass distribution...) that are not found in humanoids and *vice-versa*. In the computer graphics and animation field, there are noticeable papers showing high dexterity skills in bipedal walking of animes from human-captured data, e.g., [18], [19], [20], [21], to cite a few. Yet porting such methods on humanoid robots is not straightforward. This is because they do not account for sensors noise and uncertainties of all kinds, physical bounds, hard auto-collision avoidance, etc.

Walking gait generation using full dynamics and kinematic constraints of the robot, implying non-linear trajectory optimization in most cases, has been proposed in different ways. For example, using full semi-infinite optimization (not real-time) in [22], differential dynamic programming (DDP) [23], hybrid zero-dynamics (HZD) [24], [25], [26], [27], [28], [12] or learning based method [29], [30], [31].

Yet, other persistent approaches rely mainly on centroidal gait generation, which consists of computing a trajectory of the CoM that is feasible. This kind of strategy started with the Linear Inverted Pendulum (LIP) model [32], where the dynamics of the center of mass is constrained both vertically and in terms of angular momentum. Control strategies of the centroidal dynamics have evolved to account for the vertical height variation, e.g., [33], [34], or the angular momentum around the center of mass [35], [36]. The complete centroidal dynamics can now be captured [37], [38], but it requires non-linear solvers if one must account for all the constraints simultaneously. The LIPM still represents an interesting approach because its assumptions can be enforced on a real humanoid, and it is linear; it also allows reasoning relatively easily on the stability of such a model.

Our work uses the LIPM and builds on the contribution of [7]. Hence, we kept the notations as close as possible to the original ones. The overall notation has the following rules:

- Super-scripts $x^{\square}$ refer to indexes in a sequence, when it designates the power of x there will be no possible confusion;
- The super-script $\square^*$ refers to all the current references applied to the robot
- Lower-script $x_{\square}$ indices refers to the value type (for ZMP we use z, for CoM c, for DCM u, for foot f, etc.);
- x, y, z as variables refer to components of a position in the x, y, z axis respectively.
  The axis component can also be specified in some cases on the lower script without possible confusion such as a vector $\mathbf{b}_{\square}$ would be read in a 3D case as: $\mathbf{b}_{\square} = \begin{pmatrix} b_{\square,x} & b_{\square,z} & b_{\square,z} \end{pmatrix}^T$.

A. Inverted Pendulum Mode

The LIPM is derived from the Newton-Euler equations under the assumption that the angular momentum around

[1]Research in passive walkers is however useful to have insights in motorized gait energy optimization.

the CoM is constant, and that the CoM is at a constant height. It leads to a relationship between the CoM projection $\mathbf{p}_c = \begin{pmatrix} x_c & y_c \end{pmatrix}^T$ and the ZMP $\mathbf{p}_z = \begin{pmatrix} x_z & y_z \end{pmatrix}^T$ 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 the gravitational acceleration magnitude and $\hat{z}_c$ the constant difference between the CoM and the ZMP height; with the ZMP set at the stance foot height.

This model was initially introduced for humanoid locomotion by [32] formulated as an LQ preview controller. The latter generates a CoM trajectory under a pre-generated ZMP objective trajectory. An extension of this model in [39] allows having both the CoM and the ZMP as decision variables. Generated trajectories are enforced by constraints on the ZMP, to sustain contact balance, and on the steps to keep their kinematic feasibility, see e.g., [7], [40], [41].

Using the following change of variable:

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

equation (1) can be rewritten as

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

where x_u is the Divergent Component of Motion (DCM), introduced in [2]; its dynamic is such that, while the CoM is converging exponentially towards the DCM, this latter is diverging exponentially from the ZMP. Controlling this first-order dynamics between the ZMP and the DCM is used extensively in walking gait generation, e.g., [41], [42], [7], [8], as it simplifies the dynamics and extends the balance stability criterion to the gait generation.

Using the DCM dynamics, for a bounded ZMP trajectory $x_z(t)$ in an interval $t \in [\tilde{t}, \infty]$, The CoM dynamics is bounded if and only if the initial DCM $x_u(\tilde{t})$ fulfills the following condition, see details in [43]:

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

with e being the exponential function, i.e., $\mathrm{e}^{(\cdot)} \triangleq \exp(\cdot)$.

Equation (4) is the *stability condition*. Using (4) guarantees the boundness of the CoM (if the ZMP or its velocity is bounded) at all times.

This stability condition brings interesting features:

1) It provides a linear constraint between the current DCM and the ZMP trajectory to generate;
2) As the ZMP-allowed location depends on the future step positions and step durations parameters (i.e., for both double and single support durations), the feasibility of a walking plan is also captured.

This feasibility condition is an alternative to penalizing divergent trajectories as part of the cost function (commonly minimizing the CoM jerk), a heuristic solution that can be dependent on the duration of the prediction window of the MPC. It is the basis of the IS-MPC proposed in [7], In [8], this condition is linearized w.r.t the step durations in the case of forward walking (where the constraints are decoupled for the axes). It allows a change in the incoming step duration. Similarly, [42] uses the dynamics of the DCM (in point contact feet), which simplifies the derivation of a boundness condition to compute the next step timing and the next step location simultaneously.

Note that the DCM is used as a baseline to relax the constraints on the pendulum height and creates a variable height pendulum model in [33], [44].

When it comes to controlling a humanoid under LIPM assumptions, one must guarantee that the robot's dynamics follow the generated gait. In order to mitigate modeling uncertainties that are prone to errors and disturbances, a complementary policy based on the DCM tracking feedback can be added to the control of the ZMP to enforce the DCM to converge exponentially to its planned reference, e.g., [45], [9]. This DCM feedback control loop is often named *stabilizer* as its purpose is to 'stabilize' the reference walking gait. This concedes the open-loop aspect of the gait generators previously mentioned, as one must *stabilize* the generated gait to keep the motion feasible. However, even if some control schemes in the literature consider the state of the robot in the gait generation [7], [39], only part of the state is provided to the planning as the ZMP dynamics is not considered. Therefore, such a complementary policy is still needed.

B. Disturbed Pendulum

The ZMP in the LIPM represents the contact forces one can control. Yet, supplementary contact forces can be added to our system and should be considered.

Writing the centroidal dynamics and adding an external disturbance wrench, $(\mathbf{n}, \mathbf{f}) = \begin{pmatrix} n_x & n_y & n_z & f_x & f_y & f_z \end{pmatrix}^T$ applied at a contact point $\mathbf{p} = \begin{pmatrix} p_x & p_y & p_z \end{pmatrix}^T$, the pendulum dynamics become as follows (see equations (1) to (4) from [46] for the derivation):

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

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

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

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

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

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

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

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

An external disturbance is then seen in the pendulum dynamics as an offset on the ZMP with a proportional coefficient on its trajectory.

C. Force control

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

Having the 6D spatial velocity motion vector of a robot link $\begin{pmatrix}\boldsymbol{\omega}^T & \mathbf{v}^T\end{pmatrix}^T$, we use an acceleration-based tracking law [47] with only a reference spatial velocity.

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

K_d is the damping coefficient. In the case of force control, we set the reference spatial velocity such that [47]:

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

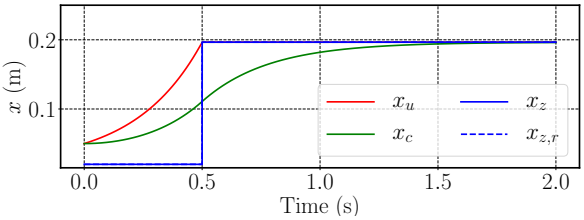
where $\mathbf{w}_r$, $\mathbf{w}_m$ are the reference and measured 6D wrench in the link frame respectively; K_a is the admittance gain. To control the robot ZMP, we must control the Center of Pressure (CoP) of the current contacts. This is done by controlling the moment in the x and y direction in the contact frame and the vertical force. To control the moments, we use (12) in two dimensions. However, the control of the vertical forces during a double support phase is done using a foot force difference control (FFDC) [48], [49]. Noting (v_L, v_R) the vertical velocity of the left and right foot respectively, FFDC with gain K_z updates the reference velocity ($v_{L,r}$,$v_{R,r}$) [9] as follows:

$$v_{L,r} \leftarrow v_{L,r} - 0.5 v_{\delta f} \tag{13}$$

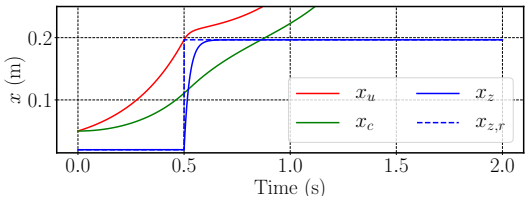
$$v_{R,r} \leftarrow v_{R,r} + 0.5 v_{\delta f} \tag{14}$$

$$v_{\delta f} \triangleq K_z\left[(f_{L,r} - f_{R,r}) - (f_{L,m} - f_{R,m})\right] \tag{15}$$

When it comes to evaluating the tracking of such a force control scheme, one must consider the deformation of either the contact surface of the foot sole or ankle. Moreover, many of the (our) humanoid robots (e.g., HRP-4CR, HRP-4, HRP-2KAI) are equipped with a passive-compliant shock absorbing mechanism between the ankle and the feet, with a thin layer of soft sole to each foot. This flexibility can be seen as a 2nd order model between the torque at the contact point and the deformation [50]. Moreover, even in a configuration without a shock-absorbing mechanism, similar behavior can be drawn if we consider the floor as a compliant material that deforms under the contacts (the flexibility at the ankle can be approximated as a pure spring whereas the floor exhibits rather a spring damper behavior). As our force control relies on the position of an end-effector to obtain the desired force, this 2nd order behavior between the torque at the contact point and the deformation reflects the behavior between the desired CoP and the references forwarded into the force control scheme. A simpler model empirically evaluates the behavior between the CoP and its reference (using a force control defined in (12)) using first-order dynamics [48].



(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.

Fig. 3: LIPM behaviors for the toy example.

III. Walking gait using closed-loop IS-MPC

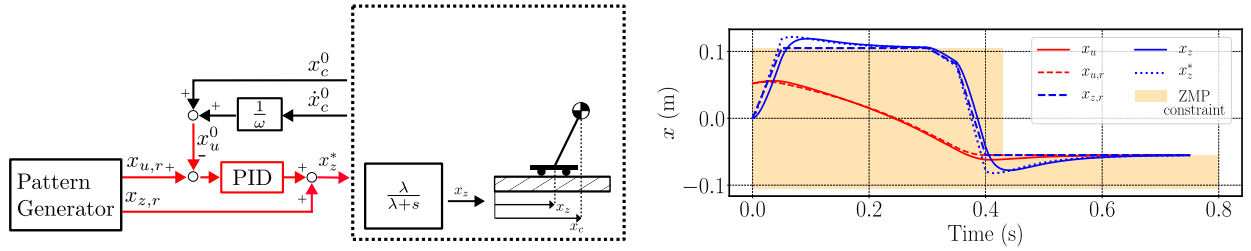
A. Closed-loop control of the pendulum– A toy example

To showcase the use of the DCM feedback control and how to overcome it, we analyze the analytical and simulated dynamics of a perfect pendulum model in one dimension. We initialize the pendulum with an arbitrary and distinct DCM and ZMP at initial time $t^0 = 0$ and choose the ZMP at $t = t^0 + \Delta t$ (here $\Delta t = 0.5$ s) to bring the pendulum to a stop. Since it is a toy example, the final position of the pendulum does not matter; yet the same reasoning extends to the case where it does. From (4), having $x_z(\Delta t) = x_u(\Delta t)$ stabilizes the pendulum, see Fig. 3a. In this case, the ZMP jumps instantly to the DCM. But this is impossible because of physics causality. 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., [48], that is, with a parameter λ:

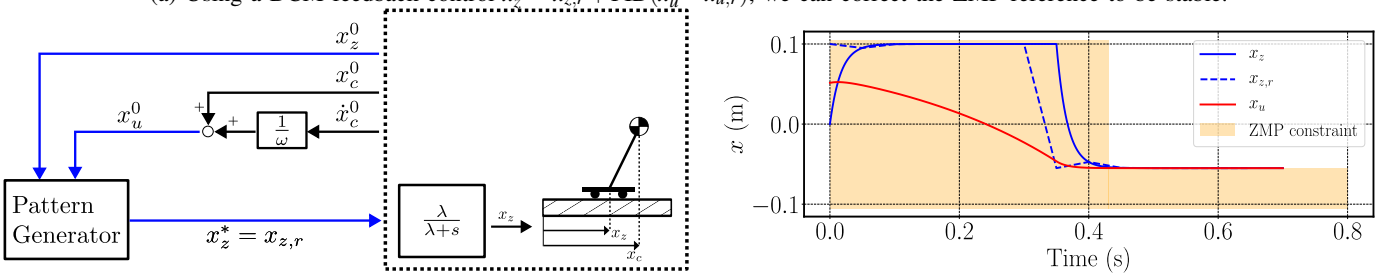
$$\dot{x}_z(t) = -\lambda\left(x_z(t) - x_{z,r}\right) \tag{16}$$

Integrating such ZMP dynamics to the LIPM, by setting $x_{z,r} = x_u(\Delta t)$, leads to instability (see Fig. 3b). 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 is a common strategy leading to an exponential convergence of the measured DCM towards the reference one (see Fig. 4a) [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. Figure 4 shows a gait generated using the IS-MPC in [7] where the corrective ZMP is out of the



(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).

Fig. 4: Linear inverted pendulum models (toy example) control schemes, highlighting our proposed approach.

predefined constraints. Finally, it can be seen from [46] that if $\lambda < \omega$, this DCM feedback policy becomes unstable.

> Our solution uses the dynamics of the ZMP directly and explicitly as part of the walking pattern generation, see Fig. 4b. In our example, this amounts to solving (4) and computing the value of the reference ZMP such that the real ZMP would converge to it at the same time as the DCM. 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. 4. Note that in real-case applications, Fig. 4a's scheme can be used with a semi-closed-loop component by providing the DCM state to the pattern generator. However, this still requires a DCM feedback policy to guarantee proper tracking of the pendulum state.

B. IS-MPC

The IS-MPC generates a feasible CoM gait to achieve a *footstep plan*. This footstep plan consists of footstep prints (both location and orientation) and footstep durations over a preview horizon (T_p) longer than the MPC control one.

If we want the robot to stand still, the generated gait from an empty footstep plan. We can then dissociate the walking phase from the standing phase. Transitions can be triggered manually by the user or automatically (see Sec. IV-F)

The initial description of the IS-MPC uses the LIPM by taking $\dot{x}_z$ (or equivalently $\dot{y}_z$) as the input term, we define the following dynamics from (1) with the state $\begin{pmatrix} x_c & \dot{x}_c & x_z \end{pmatrix}^T$

$$\begin{pmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{x}_z \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ \dot{x}_c \\ x_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{x}_z \tag{17}$$

However, in order to use this MPC in closed-loop on the robot pendulum full state (i.e., including also the ZMP and the DCM), one shall model the relation between the ZMP reference forwarded to the force controller (e.g., admittance) and the measured ZMP value. We model this behavior as a first-order system, similarly to (16), with a parameter λ and a delay δ_d such that:

$$\dot{x}_z = -\lambda\left(x_z(t) - x_{z,r}(t - \delta_d)\right) \tag{18}$$

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 (17) can then be rewritten as follows (that also apply to the y-axis):

$$\begin{pmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{x}_z \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & -\lambda \end{pmatrix} \begin{pmatrix} x_c \\ \dot{x}_c \\ x_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \lambda \end{pmatrix} x_{z,r}(t - \delta_d) \tag{19}$$

Let's consider this system at instant t^0 and δ_d constant,

$$x_z(t) = x_z^0 + (x_z^* - x_z^0)(1 - \mathrm{e}^{-\lambda(t-t^0)}) \text{ for } t \in [t^0; t^0 + \delta_d] \tag{20}$$

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

The reference $x_{z,r}(t)$ is the input to the force control loop. Let's consider it piece-wise constant on a duration $\Delta t > \delta_d$. We define inputs u_x^k on $[t^k; t^{k+1}]$ where $t^k = t^0 + k\Delta t$, as increments added to the ZMP reference $x_{z,r}$ such that on $[t^k; t^{k+1}]$:

$$\begin{aligned} x_{z,r}(t) &= \bar{x}_z^0 + \sum_{l=0}^{k} u_x^l \\ \bar{x}_z^0 &= x_z^0 + (x_z^* - x_z^0)(1 - \mathrm{e}^{-\lambda\delta_d}), \end{aligned} \tag{21}$$

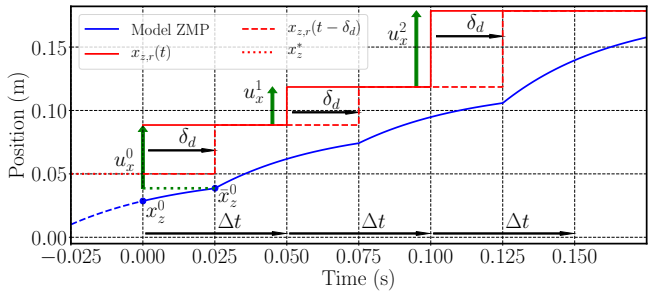


Fig. 5: Chosen 1st order ZMP model with $\lambda = 25$. The green dashed line indicates the starting point of the first arrow.

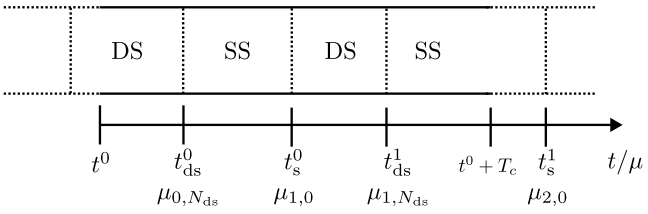


Fig. 6: Disposition of the planning time variables.

The ZMP dynamics in (18) can then be integrated with (21) to obtain the following ZMP trajectory for $t \in [t^k + \delta_d; t^{k+1} + \delta_d]$:

$$x_z(t) = \bar{x}_z^0 + \sum_{l=0}^{k} u_x^l \left[1 - \mathrm{e}^{-\lambda(t - t^l - \delta_d)}\right] \tag{22}$$

One can see each $u_\square^l$ as a sequence of first-order step delta input, each being delayed by t_d^k. Figure 5 presents the expected model behavior.

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. At each iteration, the proposed closed-loop IS-MPC considers a control horizon of C samples of Δt duration each. The provided footstep plan defines the number of steps N_s in the control horizon. We then generate the following values:

1) The reference inputs $U = \begin{pmatrix} \mathbf{u}^{0T} & \dots & \mathbf{u}^{C-1T} \end{pmatrix}^T$;
2) The footsteps locations $P_f = \begin{pmatrix} \mathbf{p}_f^{1T} & \dots & \mathbf{p}_f^{N_s T} \end{pmatrix}^T$. We note $\mathbf{p}_f^0$ and $\mathbf{p}_f^{-1}$ respectively the current and previous support foot location at time t^0.

The fixed time parameters are $T_s = \left[(t^0_{\text{ds}}, t^0_{\text{s}}), \dots, (t^{N_s-1}_{\text{ds}}, t^{N_s-1}_{\text{s}})\right]$ which are the times of the start and end of the single support duration. Figure 6 shows the organization of the time variables. $\hat{P}_f$ are the reference footsteps location. The control horizon time is $T_c = C\Delta t$. Once computed, the LIP dynamics defined in (19) is integrated using (21) under computed input (u_x^0, u_y^0) at sampling rate $\delta t \leq \Delta t$; δt is the sampling period of the whole-body control loop. The MPC solves a quadratic optimization problem (described in (37)) under different linear constraints enforcing dynamic balance.

C. Constraints

1) ZMP constraints: At time t^k, $\mathbf{p}_z(t^k) = \begin{pmatrix} x_z^k & y_z^k \end{pmatrix}^T \triangleq \mathbf{p}_z^k$ is constrained to be within a convex polygon, located either under the support foot or in-between both feet, depending on t. Let $N(t)$ and $O(t)$ be the normal matrix and the offset vector respectively, we have:

$$N(t^k)\mathbf{p}_z^k \leq O(t^k), \tag{23}$$

therefore there are C ZMP constraints in the form of (23).

For each computation of the MPC, at the current walking phase (single or double support), the ZMP constraint polygon is completely defined by the current robot foot positions and does not depend on any decision variables; therefore the constraints can be set at will. Otherwise, $N(t)$ and $O(t)$ define an admissible ZMP position over the entire horizon, they depend on P_f. Hence, if we set N and O to define the whole support area, the constraint (23) is not linear w.r.t. P_f.

To keep the linearity between the next ZMP constraints and P_f, the ZMP-allowed region is a rectangle of dimension (dx, dy) sliding from one foot to the other. For each sample taken after t_s^0, we rewrite its timing as a function of the step-index $i \geq 1$ and an index $j \geq 0$ such that:

$$t^k = t_s^{i-1} + j\Delta t. \tag{24}$$

As a consequence, $N(t^k = t_s^{i-1} + j\Delta t)$. The ZMP constraints being a rectangle, its normal matrix is:

$$N(t^k = t_s^{i-1} + j\Delta t) = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}^T R(t^k) \tag{25}$$

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

Offset $O(t)$ is expressed in the interval $[t_{\text{s}}^{i-1}, t_{\text{s}}^i]$ as:

$$O(t^k = t_{\text{s}}^{i-1} + j\Delta t) = \tfrac{1}{2} \begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + N(t^k) \left(\alpha^{i,j}\mathbf{p}_f^i + (1 - \alpha^{i,j})\mathbf{p}_f^{i-1}\right) \tag{26}$$

$$\alpha^{i,j} = \frac{j\Delta t}{t_{\text{ds}}^i - t_{\text{s}}^{i-1}}, \tag{27}$$

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

2) Footsteps constraints: Within the leg's reachable space, the difference between two consecutive footsteps must be bounded.

This can be achieved using N_{kin}^i and O_{kin} a 4×2 normal matrix and the 4×1 offset vector of the step kinematic constraints such that for $1 \leq i \leq N_s$

$$N_{\text{kin}}^{i-1}(\mathbf{p}_f^i - \mathbf{p}_f^{i-1}) \leq O_{\text{kin}}^i \tag{28}$$

$$O_{\text{kin}}^i = \frac{1}{2}\begin{pmatrix} dx_f & dy_f - l^i & dx_f & dy_f + l^i \end{pmatrix}^T \tag{29}$$

where l^i is an offset, positive if $\mathbf{p}_f^{i-1}$ is a step from the right foot and negative otherwise. Matrix N_{kin}^{i-1} are the normals of a

rectangle oriented according to $\mathbf{p}_f^{i-1}$; subsequently it is defined as in (25),

$$N_{\text{kin}}^{i-1} = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}^T R(t_s^{i-1}) \tag{30}$$

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:

$$\begin{aligned} x_u^0 &= x_u^\star(t^0, x_z) \text{ and} \\ x_u^\star(t^0, x_z) &= \underbrace{\omega \int_{t^0}^{t^0+\delta_d} x_z(\tau) \mathrm{e}^{-\omega(\tau-t^0)}\, \mathrm{d}\tau}_{\bar{x}_\phi} \\ &\quad + \omega \int_{t^0+\delta_d}^{t^0+\delta_d+T_c} x_z(\tau) \mathrm{e}^{-\omega(\tau-t^0)}\, \mathrm{d}\tau \\ &\quad + \omega \int_{t^0+\delta_d+T_c}^{\infty} x_z(\tau) \mathrm{e}^{-\omega(\tau-t^0)}\, \mathrm{d}\tau \end{aligned} \tag{31}$$

$x_u^\star(t^0, x_z)$ can be split as the sum of three integrals:

- The part of the ZMP trajectory delayed and that cannot be controlled (interval $[t^0; t^0+\delta_d]$); we note it $\bar{x}_\phi$;
- The controlled part of the ZMP trajectory in the interval $[t^0+\delta_d; t^0+\delta_d+T_c]$;
- The imposed ZMP trajectory in the interval $[t^0+\delta_d+T_c; \infty]$ that are set and noted $\tilde{x}_z(t)$; it is the *tail* of the trajectory beyond the horizon defined in [7].

The stability constraint (31) becomes:

$$x_u^0 - \bar{x}_\phi = \omega \int_{t^0+\delta_d}^{\infty} x_z(\tau) \mathrm{e}^{-\omega(\tau-t^0)}\, \mathrm{d}\tau \tag{32}$$

And using the value of $x_z(t)$ in (20)

$$\bar{x}_\phi = \omega \int_{t^0}^{t^0+\delta_d} x_z^0 + (x_z^* - x_z^0)\left[1 - \mathrm{e}^{-\lambda(\tau-t^0)}\right] \mathrm{e}^{-\omega(\tau-t^0)} d\tau \tag{33}$$

We set $x_z(t)$ from (22). We note $\tilde{u}_x^k$ the control inputs beyond the horizon (i.e, $k \geq C$). Equation (32) writes:

$$\begin{aligned} x_u^0 - \bar{x}_\phi &= \omega \int_{t^0+\delta_d}^{\infty} \bar{x}_z^0 \mathrm{e}^{-\omega(\tau-t^0)}\, \mathrm{d}\tau \\ &\quad + \sum_{k=0}^{C-1} \omega \int_{t^0+\delta_d+k\Delta t}^{\infty} u_x^k (1 - \mathrm{e}^{-\lambda(\tau-t^k-\delta_d)}) \mathrm{e}^{-\omega(\tau-t^0)}\, \mathrm{d}\tau \\ &\quad \underbrace{+ \sum_{k=C}^{\infty} \omega \int_{t^0+\delta_d+k\Delta t}^{\infty} \tilde{u}_x^k (1 - \mathrm{e}^{-\lambda(\tau-t^k-\delta_d)}) \mathrm{e}^{-\omega(\tau-t^0)}\, \mathrm{d}\tau}_{\tilde{x}_\phi} \end{aligned} \tag{34}$$

To finally have:

$$x_u^0 - \tilde{x}_\phi - \bar{x}_\phi = \mathrm{e}^{-\omega\delta_d} \left[\bar{x}_z^0 + \frac{\lambda}{\lambda+\omega} \sum_{k=0}^{C-1} u_x^k \mathrm{e}^{-\omega k \Delta t} \right] \tag{35}$$

4) Tailing: Refers to the behavior of $\tilde{x}_z(t)$ that can be different depending on the expected future motion (see [7] for more details):

- *Truncated* tailing sets the ZMP to a stop beyond the control horizon, that is:

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

- *Periodic* tailing expects the ZMP trajectory to periodically repeat every T_c. Therefore, $\tilde{x}_z(t)$ follows inputs $\tilde{u}_x^j$, that is $\tilde{u}_x^j = u_x^i$, where j is congruent to i modulo C.
  The stability condition is rewritten as:

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

- *Anticipative* tailing uses the part of the footstep plan defined beyond the control horizon to generate a reference ZMP path. It allows $\tilde{x}_z(t)$ to track the desired behavior from the walking plan. After that, the truncated or periodic policy is added to complete the tail, see [7].

D. Cost Function

The cost function aims at generating a ZMP trajectory that satisfies the following weighted objectives:

1) a predefined ZMP trajectory objective $x_{z,\text{obj}}$. It is also possible to set a component of the ZMP velocity magnitude with an objective velocity set as zero;
2) a DCM trajectory objective. The latter is obtained from computing the stable DCM $x_u^\star(t^0, x_{z,\text{obj}})$ and integrate it over the LIPM using (17) with $x_{z,\text{obj}}$ as input. One can also obtain a DCM velocity objective using (3).
3) step locations objective aiming to have them as close as possible to the references ones.

To summarize, the decision variables on the horizon are:

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

We use a similar notation for the ZMP and DCM position and velocity sequence. The IS-MPC solves the following quadratic optimization problem (QP):

$$\begin{aligned} U, P_f = \underset{U,P_f}{\arg\min}\ & \beta_{\dot{z}} \|\dot{P}_z\|_2 + \beta_z \|P_z - P_{z,\text{obj}}\|_2 + \\ & \beta_f \|P_f - \hat{P}_f\|_2 + \beta_u \|P_u - P_{u,\text{obj}}\|_2 + \\ & \beta_{\dot{u}} \|\dot{P}_u - \dot{P}_{u,\text{obj}}\|_2 \end{aligned} \tag{37}$$

under the constraints:

- ZMP, (23);
- Footsteps, (28);
- Stability, (35).

E. External forces

From (10), if the humanoid is under expected or unexpected external forces, one must control the dynamics of the pendulum under a new state $\kappa x_z - \Delta x_c'$. If we know the duration $\delta_p > \delta_d$ for how long these forces are active, we can rewrite the stability condition, (4), as follows:

$$\begin{aligned} x_u^\star(t^0, x_z) &= \omega \int_{t^0}^{\infty} (\kappa x_z(\tau) - \Delta x_c') \mathrm{e}^{-\omega(\tau-t^0)} d\tau \\ &\quad - \omega \int_{t^0+\delta_p}^{\infty} (\kappa x_z(\tau) - \Delta x_c') \mathrm{e}^{-\omega(\tau-(t^0+\delta_p)+\delta_p)} d\tau \\ &\quad + \omega \int_{t^0+\delta_p}^{\infty} x_z(\tau) \mathrm{e}^{-\omega(\tau-(t^0+\delta_p)+\delta_p)} d\tau \end{aligned} \tag{38}$$

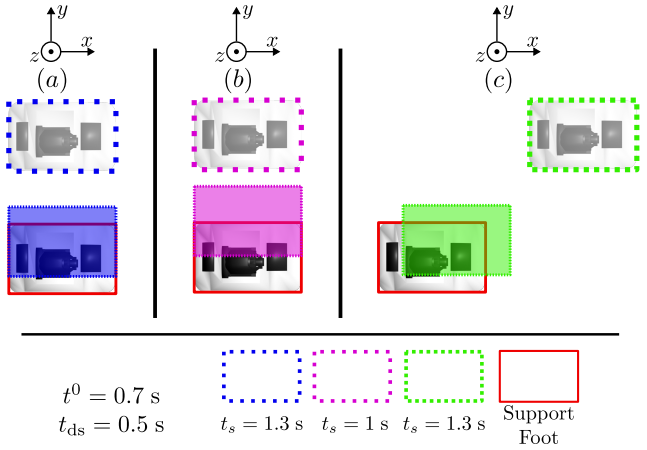


Fig. 7: Top view of the foot and footprints with the feasibility region as colored rectangles for three cases of step target and step frequency. The current phase at t^0 is single support.

Using the definition in (4) we have

$$x_u^0 = x_u^\star(t^0, \kappa x_z - \Delta x_c') - \mathrm{e}^{-\omega\delta_p} x_u^\star(t^0 + \delta_p, \kappa x_z - \Delta x_c') + \mathrm{e}^{-\omega\delta_p} x_u^\star(t^0 + \delta_p, x_z). \tag{39}$$

The stability condition is updated to account for these forces in the dynamic model.

IV. Steps Timing and Position Pre-Optimization

We describe how to update the step planning if not feasible. Let N_s be the number of planned steps. Knowing all the current constraints in (37), the current DCM's feasible region can be computed. We rewrite the stability condition by combining (4) with (23). To do so, we assume that the normals N defined in (25) are constant; they are noted $\bar{N}$.

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 can be satisfied iff:*

$$\bar{N}\mathbf{p}_u^0 \leq \omega \int_{t^0}^{\infty} O(\tau) e^{-\omega(\tau - t^0)} \,\mathrm{d}\tau \tag{40}$$

The proof of this theorem is given in the Appendix.
This condition (resulting from the IS-MPC constraints) requires the current DCM to be inside a polygon (called *feasibility region* and generated by (40)) to have a ZMP trajectory that keeps the DCM trajectory bounded and guarantees a solution for the IS-MPC. The *feasibility region* captures the planned motion of the pendulum, as $O(t)$ represents the location and the size of the ZMP constraint, which in turn depends on the step location. Finally, as the ZMP constraint locations are timed, the stability constraint also depends on step durations.

For example, Fig. 7 shows three different step plans where the target step is the dashed rectangles with their respective stepping frequency. The plain colored rectangles are the resulting *feasibility region*. In cases (a) and (b), if the DCM position in the x-axis is outside the feasibility region (say because of a push in $+x$ direction), the planned step is not feasible, a step forward must be done (as in case (c)). Moreover, in case (b), if the DCM in the y-axis is outside the pink region because of a push in the $-y$ direction, changing the footstep location can be tricky as it might lead to a collision between the feet. However, reducing the step frequency (case (a)) allows us to have a feasible solution. Therefore, being able to update the footstep plan in terms of footstep position or/and footstep timings helps improve the humanoid balance capabilities.

A. Problem formulation

We discretize (40) and define the k^{th} 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 (40) becomes:

$$\bar{N}\mathbf{p}_u^0 \leq \sum_{k=0}^{\infty} O(t = t_c^k) \left[\mathrm{e}^{-\omega(t_c^k - t^0)} - \mathrm{e}^{-\omega(t_c^{k+1} - t^0)}\right] \tag{41}$$

We set the ZMP constraint as a rectangle of dimension (dx, dy) inside the support foot during the single support phase and slide it (in a straight line) to the other foot during the double support phase. We discretize this path into a fixed number of positions $N_{\text{ds}} + 1$ for each step (Fig. 8 shows in green the case where $N_{\text{ds}} = 3$). We rewrite t_c^k as a function of the i^{th} step and the j^{th} ZMP constraint, that is $t^{i,j}$, as follows:

$$\begin{aligned} t^{i,j} &= t_c^{i(N_{\text{ds}}+1)+j} \\ t^{i,0} &= t_s^{i-1} \text{ for } i \geq 1 \\ t^{i,N_{\text{ds}}} &= t_{\text{ds}}^i \text{ for } i \geq 0 \end{aligned} \tag{42}$$

Figure 6 illustrates how the indexes are set w.r.t to the walking phases. We split the sum in (41) between three intervals:

- Single supports phases (intervals $[t^{i,N_{ds}}; t^{i+1,0}]$);
- Double support phases (intervals $[t^{i,j}; t^{i,j+1}]$);
- Intervals beyond the footsteps plan horizon.

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

$$\mathrm{e}^{-\omega t^0}\bar{N}\mathbf{p}_u^0 \leq \sum_{i=0}^{N_s-1} \left[O^{i,N_{\text{ds}}}(\mu^{i,N_{\text{ds}}} - \mu^{i+1,0}) + \sum_{j=0}^{N_{\text{ds}}-1} O^{i,j}(\mu^{i,j} - \mu^{i,j+1}) \right] + \tilde{O} \tag{43}$$

$$O^{i,j} = \frac{1}{2}\begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + \bar{N}\left[\alpha_j \mathbf{p}_f^i + (1 - \alpha_j)\mathbf{p}_f^{i-1}\right] \tag{44}$$

where $\alpha_j = j/N_{\text{ds}}$.

If the current walking phase is in a single support, then:

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

. We can identify in the inequality (43) three parts:

- The first sum: as $\mu^{i,N_{\text{ds}}} = \mathrm{e}^{-\omega t_{ds}^i}$ and $\mu^{i+1,0} = \mathrm{e}^{-\omega t_s^i}$, this part denotes the single support phases and $O^{i,N_{\text{ds}}} = O^{i+1,0}$. It is the ZMP constraint under the foot $\mathbf{p}_f^i$;
- The inner sum: representing the double support phase, where the ZMP constraint slides from $\mathbf{p}_f^{i-1}$ to $\mathbf{p}_f^i$.
- $\tilde{O}$ representing the tailing: that is to say, how the ZMP constraints are defined after the footstep plan horizon.

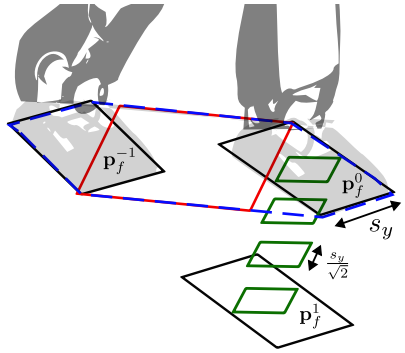


Fig. 8: View of the foot and footprints with the computation of the ZMP constraint rectangle shape. The red rectangle, once computed, defines $\bar{N}$ and the green squares are the future ZMP constraints oriented accordingly.

Except for $i=0$, the $O^{i,j}$ in (44) are similar to those in (26); yet they differ in terms of numbers during the double support phases. Indeed, here their number is constant N_{ds}; whereas in Sec. III their number is varying, it is the double support phase duration divided by the sampling rate Δt.

For the case $i=0$, the location of the footsteps $\mathbf{p}_f^{-1}$ and $\mathbf{p}_f^0$ are fixed. We set $O^{0,j}$ constant for all j so that the ZMP constraint, in the current walking phase, best approximates the current support polygon. Figure 8 shows an example and Sec. IV-B gives more details.

As in (35), the tailing $\tilde{O}$ accounts for the planning outside the horizon. Only a truncated or periodic tailing can be used because there are no more steps planned beyond this horizon. Our plan being defined with a long enough horizon (more than three steps), we set a truncated tail having the ZMP constraint to reach the middle of the two last contacts as follows:

$$\begin{aligned}\tilde{O} &= \sum_{j=0}^{N_{\text{ds}}-1} O^{N_s,j}(\mu^{N_s,j}-\mu^{N_s,j+1}) + O^{N_s,N_{\text{ds}}}\mu^{N_s,N_{\text{ds}}} \\ O^{N_s,j} &= \frac{1}{2}\begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + \bar{N}\left[\frac{\alpha_j}{2}\mathbf{p}_f^{N_s} + (1-\frac{\alpha_j}{2})\mathbf{p}_f^{N_s-1}\right]\end{aligned} \tag{45}$$

From (41), we get a constraint between the position of the footsteps and their timing to guarantee the stability condition. Therefore, we can formulate an optimization problem with,

- the steps timing $\mu^{i,j}$ and
- the footsteps position $\mathbf{p}_f^i$,

as decision variables. The steps timings can be retrieved as,

$$\begin{aligned} t_{\text{ds}}^i &= -\frac{1}{\omega}\ln(\mu^{i,N_{\text{ds}}}) \\ t_{\text{s}}^i &= -\frac{1}{\omega}\ln(\mu^{i+1,0}) \end{aligned} \tag{46}$$

which gives the timing plan illustrated in Fig. 6.

We add constraints to this optimization problem on: (i) the step locations, as defined in Sec. III-C2; and (ii) the steps phase duration such that:

$$\begin{aligned} \Delta t_{\text{ds,m}} \leq\ & t_{\text{ds}}^i - t_{\text{s}}^{i-1} \leq \Delta t_{\text{ds,M}} \\ \Delta t_{\text{ss,m}} \leq\ & t_{\text{s}}^i - t_{\text{ds}}^i \leq \Delta t_{\text{ss,M}} \\ \Delta t_{\text{s,m}} \leq\ & t_{\text{s}}^i - t_{\text{s}}^{i-1} \leq \Delta t_{\text{s,M}} \end{aligned} \tag{47}$$

with $\Delta t_{\text{ds,m}}$, $\Delta t_{\text{ss,m}}$, $\Delta t_{\text{s,m}}$ being the minimum: double support duration, single support duration, and whole step duration respectively. The super-script M represents the upper bound of these constraints. Noting $\mu_{\text{ds,m}} = \mathrm{e}^{-\omega\Delta t_{\text{ds,m}}}$ and similarly for the other duration bounds. The timing constraints are:

$$\begin{aligned} \mu_{\text{ds,m}}\mu^{i,0} \geq\ & \mu^{i,N_{\text{ds}}} \geq \mu_{\text{ds,M}}\mu^{i,0} \\ \mu_{\text{ss,m}}\mu^{i,N_{\text{ds}}} \geq\ & \mu^{i+1,0} \geq \mu_{\text{ss,M}}\mu^{i,N_{\text{ds}}} \\ \mu_{\text{s,m}}\mu^{i,0} \geq\ & \mu^{i+1,0} \geq \mu_{\text{s,M}}\mu^{i,0} \end{aligned} \tag{48}$$

The constraint defined by (43) is quadratic and nonconvex, making the problem complex to solve and more time-consuming w.r.t our MPC. Nevertheless, this constraint becomes linear if we take independently the positions or the timings as decision variables. The solution we adopted is to solve alternately the problem either with fixed step timings or with fixed step locations. This makes it possible to formulate the entire problem as two QP optimizations.

Finally, to have always a solution and avoid the QP solvers to fail, we add slack variables s to (43) to retrieve the parameters bound to the activated constraints.

B. ZMP constraint region choice

The normals $\bar{N}$ being constant, the size of the ZMP constraints can vary, however, their orientations are similar.

The 1st ZMP constraint, i.e., the red rectangle in Fig. 8, covers the largest area of the current support polygon [51]. Recall that the remaining constraints are set as illustrated by the green ones in Fig. 8.

The size of the rectangle is set to always fit inside the upcoming support polygon. To meet the latter, a conservative way is to set the rectangle as a square of size $\frac{s_y}{\sqrt{2}}$, where s_y is the width of the robot foot.

By doing so, the area allowed for ZMP is reduced in the horizon but the ZMP constraint covers the largest area in the current walking phase. Since the contribution of the ZMP trajectory to the stability condition is exponentially decaying as we get further into the horizon (because of the time-decreasing exponential term in (4)), this solution keeps a suitable approximation of the feasibility region. This is all the more true as optimization is done at the frequency of the MPC in the control loop.

C. Steps location problem

Having the decision variables $P_f = \begin{pmatrix}\mathbf{p}_f^{1\,T} & \dots & \mathbf{p}_f^{N_s\,T}\end{pmatrix}^T$ and the planned steps $\hat{P}_f = \begin{pmatrix}\hat{\mathbf{p}}_f^{1\,T} & \dots & \hat{\mathbf{p}}_f^{N_s\,T}\end{pmatrix}^T$. We solve (with $w_s \gg 1$ and given steps timing):

$$\begin{aligned} P_f, s = \underset{P_f,s}{\arg\min}\ & \|\hat{P}_f - P_f\|_2 + w_s\|s\|_2 \\ \text{s.t.}\quad & \text{Stability, (43)} \\ & \text{Kinematics, (28)} \end{aligned} \tag{49}$$

D. Steps timing problem

Starting from the desired step timings obtained from a planner $\hat{T}_s = \left[(\hat{t}^0_{\text{ds}}, \hat{t}^0_{\text{s}}), \ldots, (\hat{t}^{N_s-1}_{\text{ds}}, \hat{t}^{N_s-1}_{\text{s}})\right]$. The objectives $\hat{\mu}^{i,j}$ are computed such that:

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

If (49) produces steps that differ from the reference, we favor relatively (to the difference) faster steps. Therefore, we create a weight function $w(\hat{\mathbf{p}}^i_f, \mathbf{p}^i_f)$ such that:

$$w(\hat{\mathbf{p}}^i_f, \mathbf{p}^i_f) = K_s \|\hat{\mathbf{p}}^i_f - \mathbf{p}^i_f\|_2 \tag{51}$$

The resulting QP problem is:

$$\begin{aligned} \mu, s = \underset{\mu^{i,j}, s}{\arg\min} \; & \| \sum_{i=0}^{N_s-1} \sum_{j=0}^{N_{ds}} \hat{\mu}^{i,j} - \mu^{i,j} \|_2 + w_s \|s\|_2 \\ & + \sum_{i=1}^{N_s-1} w(\hat{\mathbf{p}}^i_f, \mathbf{p}^i_f) \|\mu^{i,0} - \mu^{i-1,0}\|_2 \\ \text{s.t. } & \text{Stability, (43)} \\ & \text{Timing, (48)} \\ & \mu^{i,j} \leq \mathrm{e}^{-\omega t^0} \end{aligned} \tag{52}$$

with $w_s \gg 1$ and $\mu = \left(\mu_{i,j \; i \in [\![0;N_s-1]\!], j \in [\![0;N_{\text{ds}}]\!]}\right)^T$.

The output timings are the reference steps duration T_s for the IS-MPC in Sec. III.

E. Sequential linear problem solving

As our problem is formulated as two sequential and inter-dependent QPs, we alternately solve the step location problem (49) and then the step duration one (52), in this order. The reason behind the latter order is that without having slack variables, kinematics and step duration constraints, QP (49) always has a solution whereas QP (52) might not. Therefore, the step location problem is more likely to provide a feasible initial solution. Finally, as the IS-MPC updates the footstep locations once again, it is more advantageous to finish the alternate problem-solving by the QP (52).

This pattern of alternating two optimization problems is also found in other research, e.g., [52] for another problem. It can be repeated at will until a satisfactory solution is found. Each of the two QPs' constraint (43) is created using the previously computed μ and P_f respectively. We end up performing this alternate pattern twice (i.e., solving 4 QP only) to obtain in a short time a compromise between step duration and step location optimization. The footsteps plan is optimized before each iteration of the IS-MPC.

Note that if the initial walking plan is feasible, the output is unchanged by construction. Otherwise, the main goal is not to guarantee a global convergence but to have a new feasible walking plan. Therefore, our method only contributes to improving the balance of the humanoid.

F. Stepping recovery while standing

When the humanoid is standing in place, the defined region in (43) is a geometric condition on where the DCM can be without having to make a step. Because we are using a rectangular shape constraint on the ZMP, the DCM stability region is also a rectangle. Knowing from which vertices the DCM violates the stability condition allows planning which supports foot to use to recover. This is done depending on how are the feet positioned. This strategy appears to be effective if the robot is pushed forward or backward, see Sec. VII. For lateral pushes, we choose the support foot to be the furthest one from the current DCM pose. Nevertheless, for a better balance, especially in the case of lateral pushes, it remains necessary to develop more tools to recover such as in [53]), using the angular momentum, etc.

V. Robot force control

The gait generated in Sec. III is forwarded to a whole-body task-space controller in which we regulate the CoM trajectory and the contact forces to achieve the 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 delayed 1st order system w.r.t the reference ZMP $\mathbf{p}_{z,r} = \begin{pmatrix} x_{z,r}(t) & y_{z,r}(t) \end{pmatrix}^T$ computed in the IS-MPC in Sec. III. The admittance control shall apply a reference net wrench leading to the ZMP behavior expressed in (22). During single-support phases, the admittance control applies a wrench to track the contact's CoP reference at $\mathbf{p}_{z,r}$. The delay in the ZMP dynamics allows this reference to be temporarily outside the contact polygon resulting in a faster ZMP. During the double support phase, the reference wrench is distributed among the contacts and a reference CoP is computed for each of them. We assume each contacts to have their own 1st order dynamic w.r.t the real CoP with parameters λ_c; ideally $\lambda_c \simeq \lambda$. We note $\mathbf{c}_{L,r}, \mathbf{c}_{R,r}$ and $\mathbf{c}_L, \mathbf{c}_R$ respectively the reference and the modeled CoP for the left and right foot. This model, with a piece-wise constant reference, has similar dynamics as in (18):

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

which can be integrated over $[t^0 + \delta_d; t^0 + \delta_d + \Delta t]$ leading to:

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

Recall that super-script $\square^*$ refers to the current references applied to the robot. Additionally, having the same mode for f_L and f_R, the vertical forces on the left and right foot gives:

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

And similarly for the right component. Then, we set $\mathbf{c}_{L,r}, \mathbf{c}_{R,r}$ to have the same overall ZMP, that is:

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

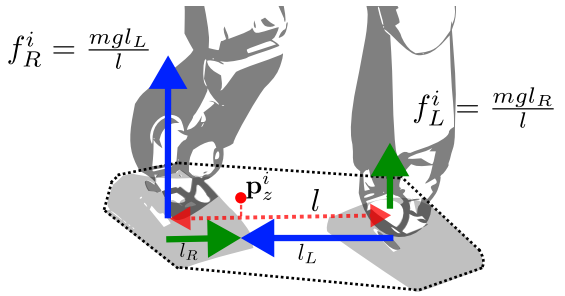


Fig. 9: Vertical force distribution depending on the desired ZMP position at time t^i. We have $\gamma = l_L/l$.

This equation gives the relation between the CoPs and the ZMP. If we assume that the vertical forces are defined in advance, this relation is linear.

There is not a single way to distribute the vertical forces and the CoPs to get the desired ZMP; we chose to minimize the moments at the ankles.

We note $\mathbf{p}_L$ and $\mathbf{p}_R$ the location of the left and right ankles and m the mass of the humanoid. To get the values of $f_L(t)$ and $f_R(t)$, we define $\gamma(t) \in [0;1]$ as the normalized projection of the vector $\mathbf{p}_z(t) - \mathbf{p}_L$ over the vector $\mathbf{p}_R - \mathbf{p}_L$ and set:

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

Figure 9 illustrates how $f_L(t)$ and $f_R(t)$ are computed. The reference vertical forces are set by (55) with parameters λ_f. To obtain $\mathbf{c}_{L,r}, \mathbf{c}_{R,r}$, one can formulate an optimization problem to approach quadratically (56). Also, as we have a ZMP horizon, we extend the force distribution computation on the entire double support phase of length $t_{\text{ds}}^0 - t^0$.

Let $\mathbf{c}_L^i, \mathbf{c}_{L,r}^i$ (and similarly for the right side) be the CoP location at time $t^i = t^0 + i\Delta t$ and the piece-wise constant reference in the interval $[t^i; t^{i+1}]$, respectively. The same notation is used for f_L as $f_L^i = f_L(t^i)$. Using (54) and (53), we have recursively:

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

Having $\mathbf{p}_z^i = \mathbf{p}_z(t^i)$ the modeled ZMP at $t^i = t^0 + i\Delta t$, we compute the future vertical force (f_L^i, f_R^i) using (57), see Fig. 9.

We then compute $(\mathbf{c}_{L,r}^i, \mathbf{c}_{R,r}^i)$ for $i \in \left[0; \left\lfloor \frac{t_{\text{ds}}^0 - t^0}{\Delta t} \right\rfloor\right]$ as follows:

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

with $w_d \ll w_z$; R_R and R_L are the rotation matrices to the right and left contact frame. The first term of the cost function is the ZMP tracking error minimization (from (56)), and the second term minimizes the moment at the ankles.

B. Contact transition

When feet contacts is released, deactivating force control often leads to residual forces on the released feet. This residual force is not considered in the pendulum state and could disturb the pendulum dynamics, see Fig. 10. One solution to overcome this is to consider it explicitly in the dynamics of the pendulum. We can update the IS-MPC stability constraint using (39) to account for a disturbance of duration δ_p set empirically to 0.1 s from experimental data.

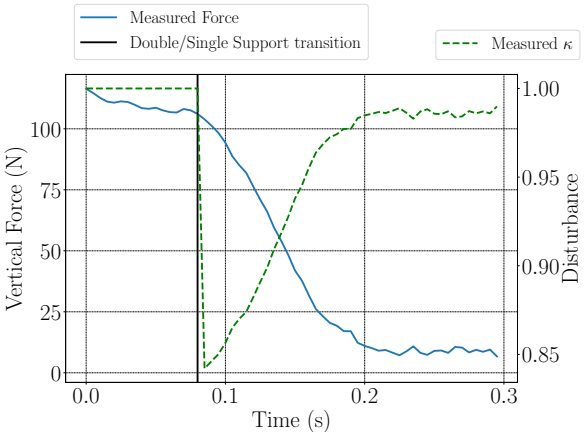


Fig. 10: Measured disturbances during contact release. Around 100 N are still observed once the swing foot force control is deactivated, resulting in a change of the pendulum dynamics.

To have a measured estimation of the mass, we rely on the force sensors located at each foot and a low-pass filter with a very high cut-off period (≥ 10 s). More accurate estimation of the robot mass and the vertical acceleration $\ddot{z}_c$ can be obtained using the non-filtered values of the force sensors. This can be useful in case of model updates, e.g., holding objects.

VI. Simulations

Real experiments would not be possible prior to intensive simulations of all the humanoid robots described later in Sec. VII. In this section, we introduce our simulation environment and relate in particular a comparative study between our closed-loop version of the IS-MPC and its original implementation without. A more thorough data analysis is left to real experiments in Sec. VII.

A. Simulation environment

The rigid body simulation of our humanoids and their environment is using the `Choreonoid` framework combined with the OpenHRP3 physical engine [54]. The computed joint commands from the whole-body control (same as the one in Sec. VII) are directly sent to the simulator which includes the low-level joint control dynamics. Hence, one can simply switch to direct control once satisfied with the simulation runs. `Choreonoid` also includes simulation of IMUs, force/-torque sensors and possibly other ones. Robots-embedded state observers and estimators are implemented similarly in the simulation. The simulated environments include flat terrains, uneven terrain and slopes (see Fig 11). Alas, it is not yet possible to simulate soft terrains reliably.

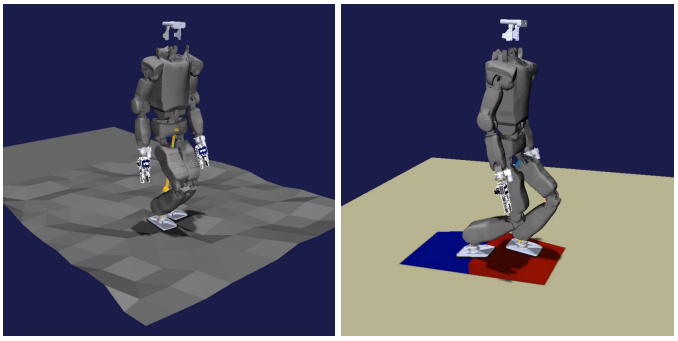
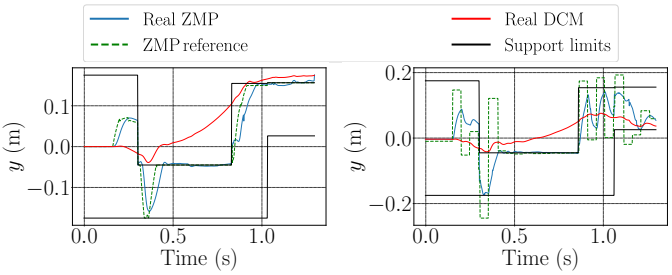


Fig. 11: Choreonoid simulation environment for uneven terrain and slope with RHPS1 robot



(a) DCM closed-loop IS-MPC. (b) Closed-loop IS-MPC.

Fig. 12: Simulation comparative between DCM closed-loop IS-MPC and closed-loop IS-MPC.

B. Stabilizer and closed-loop comparison

Besides the theoretical advantages of our method, we performed a set of comparative studies in simulation without noise and on flat terrain (ideal cases). As an example, we instructed our robots to start walking with high-frequency steps without footsteps adaptation. The reason is that by imposing a short double support duration, the robot ZMP must lie quickly under the support foot highlighting the weakness of the original *stabilizer* policy. We considered three walking control schemes:

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

We report results obtained with walking parameters set to arbitrary step duration of 0.8 s and double support durations of 0.2 s and 0.15 s. The step length is set to 25 cm (usual). The open-loop scheme failed for the double support duration of 0.2 s and the DCM closed-loop one failed for the double support duration of 0.15 s. Our proposed scheme never failed in these conditions. Pushing further the limits, there are conditions where the three methods fail. Yet, in all the simulations we made, our approach never failed where one of the two others succeeded. Figure 12 illustrates the measured DCM and ZMP and the reference ZMP for the two closed-loop cases. Notice that the ZMP reference computed by the *stabilizer* does not consider the ZMP dynamics; thus it cannot provide references outside the support polygon. Moreover, notice in Fig 12b that the delay δ_d is nil, and in Fig 12 that the measured ZMP can leave the constraint. This is because the defined support polygon is slightly smaller than the real one to have additional safety margins during the experiments.

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 we possess), see Fig. 13 and on different environments:

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

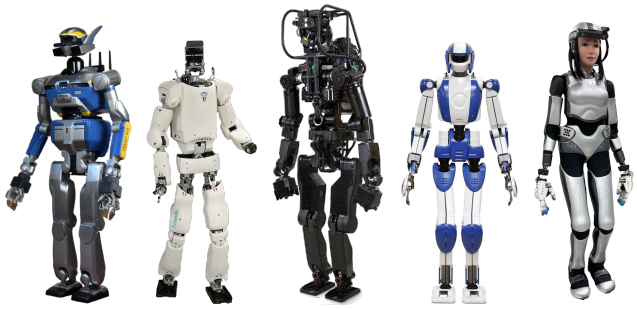


Fig. 13: All the humanoids used in the experiments. From left to right, the HRP-2KAI: a modified version of the HRP-2, used in the DRC; RHPS1: the new humanoid robot from Kawasaki Robotics co-designed with AIST and CNRS; HRP-5P: the last prototype of the HRP family designed to be torque controlled and to be used in building and harsh applications [55]; the HRP-4: the sample present in France and used for the Airbus joint project [56]; and finally, the HRP-4CR used in the ANA Avatar Xprize contest [57].

Human pushes (i.e., disturbances) are introduced either during walking or in standing phases.

All the humanoids are controlled in kinematics (high-gains position). The whole-body control is achieved by `mc_rtc` task-space QP control framework[2]. It computes the floating base and joints acceleration, which, once integrated twice are forwarded as joints command. The reason for such extensive experiments is our aim to achieve a 'plug-and-play' walking controller, or in other words, a *plug-and-walk* software that can be further improved and tried on several humanoid platforms. It is a well-recognized fact that most of the existing walking algorithms are highly tuned for a given humanoid or bipedal platform. Such a tuning never appears clearly specified in almost all existing academic papers in the field.

For each robot, we can adjust experimentally the model parameters, which are:

- The MPC's ZMP model first-order parameters λ and δ_d;
- The force distribution first-order parameters λ_c and λ_f;
- The admittances gain K_a for the CoP tasks and the foot force difference control (FFDC) gain K_z.

These parameters are interlinked and set by fitting the modeled CoP, vertical forces or ZMP to the measured ones. Indeed, λ

[2]https://jrl-umi3218.github.io/mc_rtc/

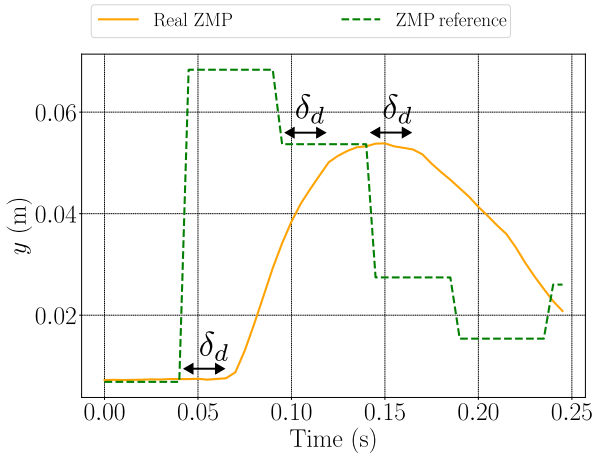


Fig. 14: Delay between the real ZMP and the sent reference on HRP-2KAI.

represents the model's 1st order response, the higher K_a is, the faster the wrench is applied and therefore the higher λ is. This is similar for the control of the contacts vertical forces. We set K_a as low as possible to estimate λ.

For the time being the tuning process for these parameters remains empirical; yet, for the sake of transparency, we provide *ad-hoc* guideline for the tuning procedure we did to conduct our experiments.

1) Set $K_a = 0.01$, $K_z = 10^{-4}$, $\lambda = \lambda_f = \lambda_c = 40$ and make the robot step in place.
2) δ_d can be identified from the transition into the walking phase, see Fig. 14.
3) Adjust λ such that the model-ZMP fits well the measured one in single supports.
4) The feet in single support should not vibrate, otherwise K_a should be increased (and so is λ).
5) Adjust λ_z such that the model vertical force fits the measured one in double support.
6) Vibration in double support is often due to vertical forces control; K_z and λ_f should be set up to a smooth and a complete transition of the ZMP on the future support foot.
7) λ_c should be at λ and adjusted to improve the ZMP trajectory w.r.t the model once in double support.
8) Finer tuning can be done by making the robot step forward and backward.

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

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

The IS-MPC weights that handle the DCM trajectory, β_u, and DCM velocity, $\beta_{\dot{u}}$, are set to a non-zero 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 reduce the compliance of the robot during disturbances and make it less likely to trigger steps. The computation performances on one robot are shown in Table III, they may slightly fluctuate depending on the onboard hardware.

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

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

In addition to those parameters, the whole-body QP cost function is made of a set of weighted tasks having as decision variables the acceleration of the joint angles and of the floating base [58]. We use the following tasks as QP-control objectives:

- The position, velocity and acceleration of the CoM;
- The contact forces (CoP Task);
- The chest orientation w.r.t the orientations of the feet;
- The swing foot position speed and acceleration during single support phases;
- A low-gains posture task (defined joints) to mitigate redundancy and singularities.

These parameters (common to all robots) are listed in Table IV. Transition from a swing foot task to a CoP Task is done once the measured force on the swing foot is over a defined threshold. The experiments show the robustness

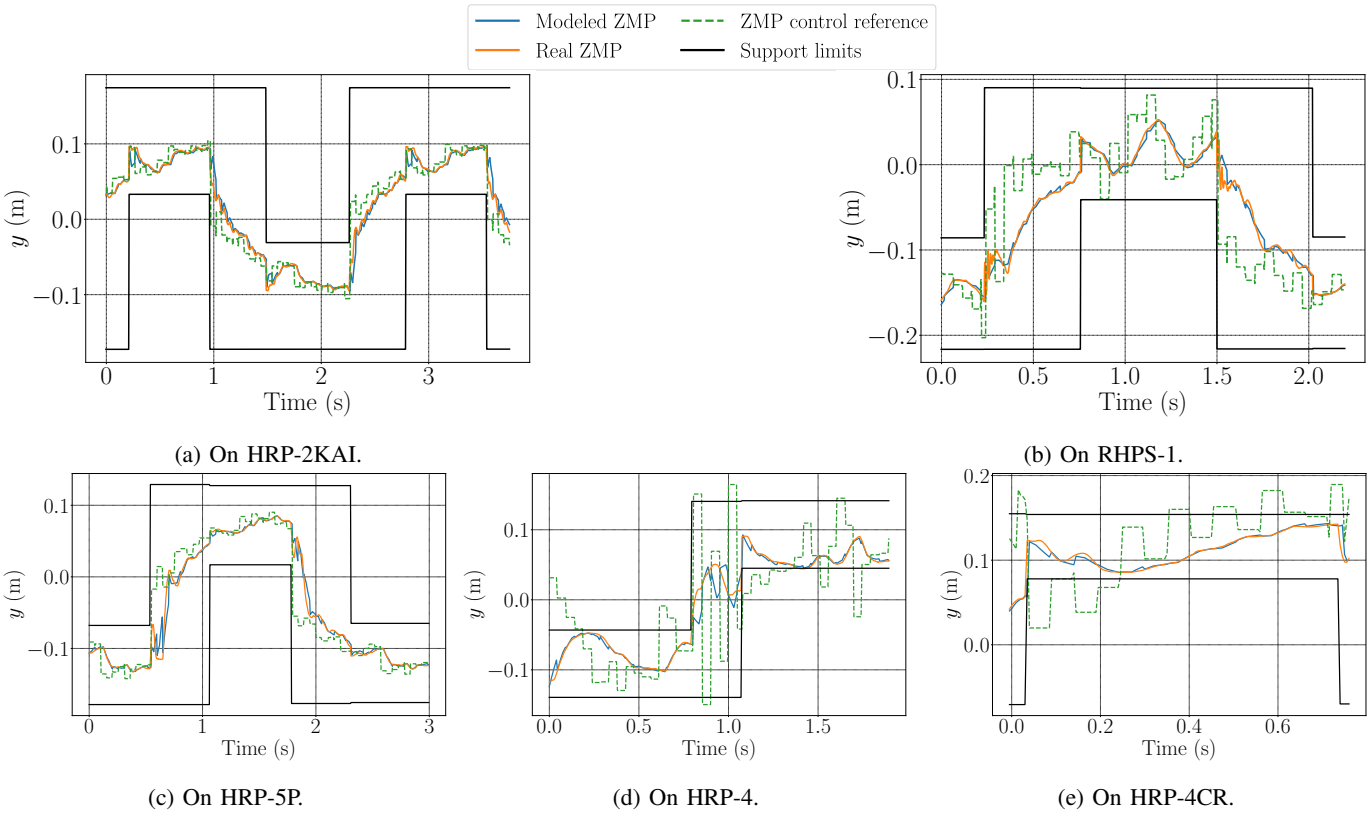
TABLE IV: Tasks parameters

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	$2\sqrt{K_p}$	$2\sqrt{K_p}$	$2\sqrt{K_p}$

of the proposed method by making the robot walk on long distances, uneven terrains, and under external disturbances provided by operator pushes. We finally tested perturbations during a standing phase to trigger stepping recovery.

All the plots are displayed in a frame linked to the robot floating base such as: $\mathbf{z}$ is the vertical axis, the plane $(\mathbf{z},\mathbf{y})$ is the coronal plane and the plane $(\mathbf{z},\mathbf{x})$ is the sagittal plane

We tried to achieve the same walking conditions for each humanoid robot; yet, due to some practical aspects, this was not always possible. Most of the robots we have are a bit old and we took care to not push them too harshly. Also, the



(a) On HRP-2KAI.

(b) On RHPS-1.

(c) On HRP-5P.

(d) On HRP-4.

(e) On HRP-4CR.

Fig. 15: ZMP model compared with robots' measured ZMP when walking on rigid flat floors; model state is updated at a sampling rate Δt.

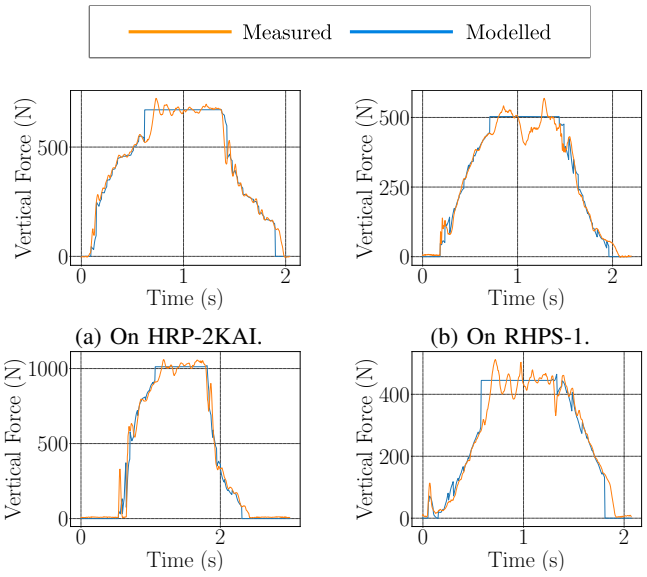
HRP-4 is located in France whereas the remaining ones are in Japan, where experiments in the campus are more strictly regulated. For example, experiments on HRP-4CR were indoor and on rigid flat floor due to mechanical weaknesses, they are reported in [57].The multimedia material accompanying the paper is thorough enough to complement the data.

A. *Force control and model evaluation*

Force control model evaluation is assessed on all the listed humanoids. Figure 15 shows that the first-order dynamic of the ZMP with a delay is an appropriate estimation of the real ZMP dynamics. To emphasize the fact that we aim at modeling the ZMP behavior, to make the robot HRP-2KAI, HRP-5P and RHPS1 walk on a compliant mattress, we adjusted λ. Even if the model response timing is very slow, the control is still stable and the robots well balanced to execute dynamic walks (see Sec. VII-D). Figure 10 illustrates how the measured disturbance from the swing foot is used to improve the pendulum trajectory planning. The figure shows only the measured κ (defined in Sec V-B) as it captures the vertical (core) component of the disturbance.

Finally, we show in Fig. 16 that the vertical forces during double support also match the expected model; when the model value is constant, the robot is in single support (the vertical force is no longer controlled).

The measure on the HRP-4 force sensor is imprecise due to a cross-talk between measured torque and vertical forces impacting the ZMP measure in double support. This coupling



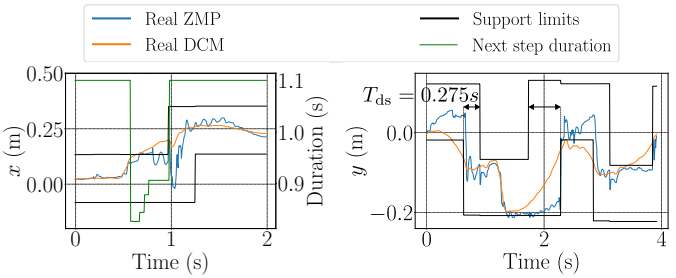
(a) On HRP-2KAI.

(b) On RHPS-1.

(c) On HRP-5P.

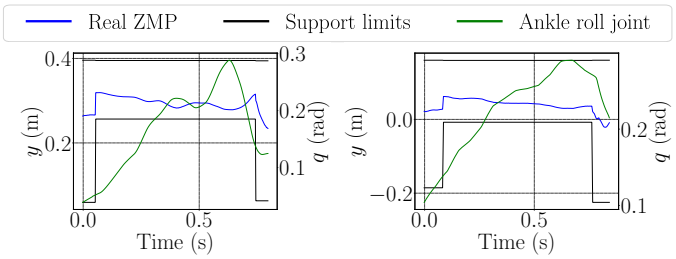
(d) On HRP-4CR.

Fig. 16: Vertical force model compared with the real robots measured one (left foot) while walking on a rigid flat ground. Model state is updated at sampling rate Δt.



(a) Forward push. (b) Lateral push.

Fig. 17: Depending on the disturbance direction, the step duration plan is updated differently (these behaviors extend to all grounds).



(a) On RHPS1. (b) On HRP-2KAI.

Fig. 18: On similar compliant ground with similar walking, feet spring damper reduces oscillation in single support.

has been properly identified in single support only and is countered by a corrective plug-in.

B. Footsteps and steps timing adaptation

The Footsteps and steps timing adaptation is also assessed on all the humanoid robots. Figure 17a illustrates a recovery after a front push and how the planning is updated. In most cases, increasing the step frequency and/or stepping in the direction of the disturbance helps to recover. However, in some configurations, the best solution is to increase the current or the next step duration as there are no proper steps to help the balance. This is seen in Fig. 17b where a lateral disturbance induces a slower double support duration to allow the DCM to reach the next support foot.

C. Static push recovery

We pushed each humanoid in a standing phase to infer the condition defined in Sec. IV-F and trigger a recovery step. The size of the ZMP region is set smaller than the allowed one such that a recovery step is triggered even if the robot is well-balanced. The supporting foot is set depending on the position of the DCM and the contact configuration at the instant of the pushes. Depending on which side of the balance region is broken, if the disturbance is frontal, the support foot is the one closest to the DCM; otherwise, the support foot is the one furthest from DCM. If a stepping is triggered, we do not provide any specific recovery footstep plan. The recovery steps are completely decided w.r.t the feasibility region. The limitation of this push recovery scheme depends mostly on the intensity of the perturbation. If the required recovery steps are unfeasible the robot will certainly fall. In such cases, it becomes necessary to introduce a variation of the angular momentum in addition to the previous strategy or change the height of the CoM (i.e, add vertical CoM acceleration). In both cases, LIPM hypotheses are not valid anymore.

D. Walking on soft ground

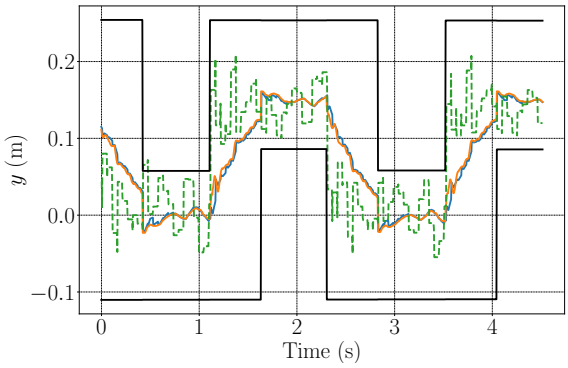
This case is conducted on HRP-2KAI, HRP-5P and RHPS1. Walking on compliant terrains for bipedal robots is a challenging task that showcases the need for knowledge of how the forces are behaving w.r.t the control inputs.

As an example, without our control scheme being active, HRP-2KAI could not stand on the mattress by itself with the CoM initially over the support area. For instance, a successful locomotion in such conditions has been already demonstrated using a torque-controlled humanoid robot [59] on which the contact compliance and feedforward force control is a considerable plus in this kind of environment. Since our control method is modeling the contact forces responses, one must adapt the parameters that model the forces' behavior to make our robots walk on compliant terrains. We made sure that the deformation of the mattress we chose never reaches its limit once in double or single support, this is to avoid a support meeting rigid ground conditions at full compression.

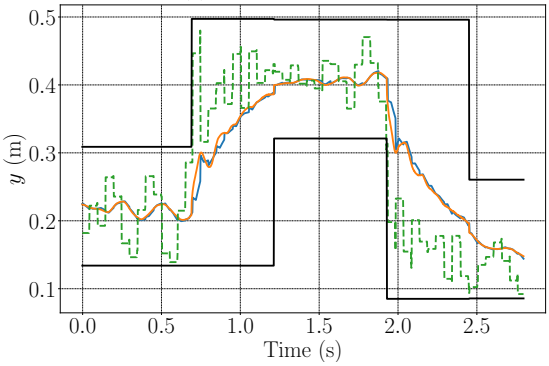
The ZMP model is evaluated in Fig 19. As mentioned in Sec. VII-A, walking on a compliant floor required to adapt the ZMP model parameter λ. It might be necessary to adapt also the FFDC K_z gain to have a more reactive vertical force control. This update of K_z was done only for the HRP-2KAI robot (it was changed from 0.0001 to 0.0002). All the humanoids we experimented over a compliant ground are position-controlled. However, the HRP-2KAI is equipped with a shock-absorbing mechanism made of a rubber bush between the foot and the ankle actuation. This difference in mechanical design affects the foot motion to control the desired force. HRP-2KAI feet motion is much smoother in single support than the one on RHPS1 (that has a rigid link between its ankles and feet). This can be seen at the joint level in Fig. 18 where, in similar conditions (environment, stepping frequency, step length, ZMP trajectory profile), RHPS1 feet oscillate whereas HRP-2KAI ones is relatively smoother.

E. Walking outdoors

Outdoor experiments highlight the benefit of compliance with force control; they are made with HRP-4 and HRP-2KAI, see Fig. 1. Indeed, even outdoor terrains that are nearly flat have non-negligible local roughness. Additionally, some terrains on which the robots walked have pseudo-compliant behavior due to the grass or ground. This allows us to assess the capability to adapt the force behavior model to various conditions. The ZMP model evaluation on such terrains is shown in Fig 20. The accompanying multimedia material shows the compliance of the contact feet on the terrain. Moreover, such terrains are uneven with various local slopes. When this slope is upward, the walking is not much affected

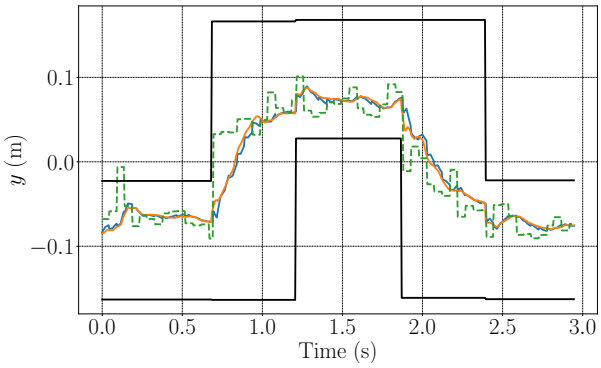


(a) On HRP-2KAI, $\lambda = 4$.

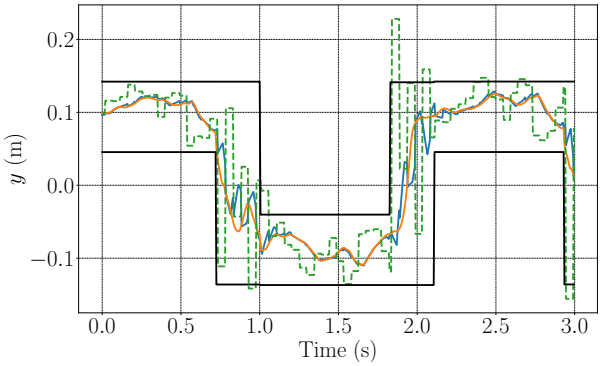


(b) On RHPS1, $\lambda = 5$.

Fig. 19: ZMP model compared with robots' measured ZMP when walking on soft grounds, model state is updated at sampling rate Δt.



(a) On HRP-2KAI, $\lambda = 12$.



(b) On HRP-4, $\lambda = 11$.

Fig. 20: ZMP model compared to the real robots ZMP walking outdoors, the model state is updated at a sampling rate Δt.

as early contacts can be detected with forces' measurement. In downward slopes, the lack of knowledge of the contact location dictates keeping on lowering the leg until contact is detected, inducing potentially a disturbance. Our force control scheme can accommodate this situation, as it keeps moving the feet down until force is measured. One can also directly use the FFDC if the real contact remains close to the feet. In that objective, the FFDC K_z gain is increased on the HRP-4 for this experiment from 0.0001 to 0.00015. In practice, poor contact detection is the main reason for few failures we experienced.

In the experiments with HRP-2KAI, the wind was very strong and added non-negligible perturbations. For the HRP-4, strong perturbations occur when steeping on thick roots.

VIII. Conclusion

In this paper, we presented a new walking control scheme for bipedal robots that can adjust step location and duration. This novel aspect of this control scheme is in the use of a LIPM's model MPC in a closed-loop on the real robot pendulum state (CoM, CoM velocity and ZMP) and does not necessitate a side stabilizer module. Finally, we presented a method to distribute the contact wrench resulting from the MPC computation. This scheme has been tested on five different humanoid robots on various grounds and scenarios to showcase its versatility and robustness.

We aimed to have as few parameters as possible and to provide a guideline to tune them. However, even if these parameters are mainly robot-dependent, they also depend on the *mechanics* of the ground. In fact, our experiments revealed that it could be possible to enhance further the *robustness* and the *plug-and-walk* aspect of such a control scheme to estimate some of the parameter online or at least autonomously during a calibration process, this is part of our shortcoming future work. This is likely possible as some of the parameters are observable. Moreover, the 1st order model for ZMP dynamics can be retrieved by modeling the floor with spring damping dynamics under the force control scheme we detailed in (12), however, the general dynamics reflect rather a 2nd order behavior which requires an appropriate estimation of the ZMP velocity. We have also tried to make the HRP-4 robot walk on river pebbles outdoors, but the controller failed in all trials. One of the causes is the flatness and rigidity of the soles; the other is clearly the lack of proper handling of the momentum. As future work, we aim to extend our control policy with full centroidal dynamics (variable CoM height and Angular momentum) [44], [36]. Moreover, we are also working on redesigning the soles, as extensions of [60] with compliant material to better cast the grounds at contacts. All these aspects are somewhat interconnected, understanding such dependencies is paramount to reaching the *plug-and-walk* goal we are aiming for. As stated in Sec. II, we are

also working toward a pure machine-learning approach for walking, convinced that robust walking would emerge from a neat hybridization that takes full benefit of both approaches.

APPENDIX
PROOF OF THE THEOREM

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 a N_v vertices convex polygon (with possibly multiple vertices at the same position) $\forall t \in [t^0;\infty]$. Let's 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 1st and 2nd element of the i^{th} row of $\bar{N}$ respectively, and $o_i(t)$ the value of the i^{th} 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 $[t^0;\infty]$:

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

using the definition of the stability condition (4):

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

This result holding for each row i, the last equation can be turned as in (40) which proves the necessity.

Let $\mathbf{v}_z^j(t)$ be the 2D coordinate of the j^{th} 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 2 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 (40)):

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

and let E be the convex set of initial DCM positions $\mathbf{p}_u^0$ such that (40) 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 $\gamma \in [0;1]$.

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

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.

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

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

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

To prove sufficiency, let $\mathbf{p}_u^0 \in E$. E being 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 particular 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):

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

Using now (68) and (69):

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

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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009, pp. 1594–1600.

[5] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, and A. Orita, "Real time motion generation and control for biped robot– 4th report: Integrated balance control-," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009, pp. 1601–1608.

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

[7] N. Scianca, D. De Simone, L. Lanari, and G. Oriolo, "MPC for humanoid gait generation: Stability and feasibility," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1171–1188, 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 Robotics and Automation Letters*, vol. 6, no. 2, pp. 1582–1589, 2021.

[9] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the HRP-4 humanoid robot using whole-body admittance control," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 277–283.

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

[11] J. Carpentier and P.-B. Wieber, "Recent progress in legged robots locomotion control," *Current Robotics 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," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 12, 10 2022.

[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: CRC Press, 31 October 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 Transactions on Robotics*, vol. 39, no. 2, pp. 905–922, 2023.

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

[18] K. Yin, K. Loken, and M. van de Panne, "SIMBICON: Simple biped locomotion control," *ACM Trans. Graph.*, vol. 26, no. 3, p. 105es, 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.

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

[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," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.

[23] E. Dantec, M. Naveau, P. Fernbach, N. Villa, G. Saurel, O. Stasse, M. Taix, and N. Mansard, "Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots*, 2022, pp. 638–644.

[24] J. Grizzle, G. Abba, and F. Plestan, "Asymptotically stable walking for biped robots: analysis via systems with impulse effects," *IEEE Transactions on Automatic Control*, vol. 46, no. 1, pp. 51–64, 2001.

[25] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[26] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3D bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.

[27] A. D. Ames, "Human-inspired control of bipedal walking robots," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, 2014.

[28] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 370–387, 2018.

[29] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 2811–2817.

[30] H. Duan, A. Malik, J. Dao, A. Saxena, K. Green, J. Siekmann, A. Fern, and J. Hurst, "Sim-to-Real learning of footstep-constrained bipedal dynamic walking," in *International Conference on Robotics and Automation*, 2022, pp. 10 428–10 434.

[31] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[32] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation*, vol. 2, Taipei, Taiwan, 2003, pp. 1620–1626.

[33] J. Englsberger, C. Ott, and A. Albu-Schffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[34] S. Caron, "Biped stabilization by linear feedback of the variable-height inverted pendulum model," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 9782–9788.

[35] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3d-slip model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5134–5140.

[36] K. Guan, K. Yamamoto, and Y. Nakamura, "Virtual-mass-ellipsoid inverted pendulum model and its applications to 3d bipedal locomotion on uneven terrains," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 1401–1406.

[37] 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 *International Conference on Robotics and Automation*, 2022, pp. 10 412–10 419.

[38] M. Murooka, M. Morisawa, and F. Kanehiro, "Centroidal trajectory generation and stabilization based on preview control for humanoid multi-contact motion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8225–8232, 2022.

[39] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 137–142.

[40] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[41] R. J. Griffin and A. Leonessa, "Model predictive control for dynamic footstep adjustment using the divergent component of motion," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1763–1768.

[42] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Walking control based on step timing adaptation," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 629–643, 2020.

[43] L. Lanari, S. Hutchinson, and L. Marchionni, "Boundedness issues in planning of locomotion trajectories for biped robots," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 951–958.

[44] F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "From walking to running: 3d humanoid gait generation via mpc," *Frontiers in Robotics and AI*, vol. 9, 2022.

[45] 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 *IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 734–740.

[46] M. Murooka, K. Chappellet, A. Tanguy, M. Benallegue, I. Kumagai, M. Morisawa, F. Kanehiro, and A. Kheddar, "Humanoid loco-manipulations pattern generation and stabilization control," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5597–5604, 2021.

[47] K. Bouyarmane, K. Chappellet, J. Vaillant, and A. Kheddar, "Quadratic programming for multirobot and task-space force control," *IEEE Transactions on Robotics*, vol. 35, no. 1, p. 6477, 2019.

[48] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4489–4496.

[49] S. Kajita, F. Asano, M. Moriswa, K. Miura, K. Kaneko, F. Kanehiro, and K. Yokoi, "Vertical vibration suppression for a position controlled biped robot," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 1637–1642.

[50] M. Benallegue and F. Lamiraux, "Estimation and stabilization of humanoid flexibility deformation using only inertial measurement units and contact information," *International Journal of Humanoid Robotics*, vol. 12, no. 03, p. 1550025, 2015.

[51] C. Knauer, L. Schlipf, J. M. Schmidt, and H. R. Tiwary, "Largest inscribed rectangles in convex polygons," *Journal of Discrete Algorithms*, vol. 13, pp. 78–85, 2012.

[52] S. Brossette and P.-B. Wieber, "Collision avoidance based on separating planes for feet trajectory generation," in *IEEE-RAS International Conference on Humanoid Robotics*, 2017, pp. 509–514.

[53] A. S. Habib, F. M. Smaldone, N. Scianca, L. Lanari, and G. Oriolo, "Handling non-convex constraints in mpc-based humanoid gait generation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 167–13 173.

[54] S. Nakaoka, "Choreonoid: Extensible virtual robot environment built on an integrated gui framework," in *IEEE/SICE International Symposium on System Integration*, 2012, pp. 79–85.

[55] I. Kumagai, M. Morisawa, T. Sakaguchi, S. Nakaoka, K. Kaneko, H. Kaminaga, S. Kajita, M. Benallegue, R. Cisneros, and F. Kanehiro, "Toward industrialization of humanoid robots: Autonomous plasterboard installation to improve safety and efficiency," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 20–29, 2019.

[56] A. Kheddar, S. Caron, P. Gergondet, A. Comport, A. Tanguy, C. Ott, B. Henze, G. Mesesan, J. Englsberger, M. Roa, P.-B. Wieber,

F. Chaumette, F. Spindler, G. Oriolo, L. Lanari, A. Escande, K. Chappellet, F. Kanehiro, and P. Rabate, "Humanoid robots in aircraft manufacturing: the airbus use-cases," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 30–45, October 2019.

[57] R. Cisneros-Limón, A. Dallard, M. Benallegue, K. Kaneko, H. Kaminaga, P. Gergondet, A. Tanguy, R. P. Singh, L. Sun, Y. Chen, C. Fournier, G. Lorthioir, M. Tsuru, S. Chefchaouni-Moussaoui, Y. Osawa, G. Caron, K. Chappellet, M. Morisawa, A. Escande, K. Ayusawa, Y. Houhou, I. Kumagai, M. Ono, K. Shirasaka, S. Wada, H. Wada, F. Kanehiro, and A. Kheddar, "A cybernetic avatar system to embody human telepresence for connectivity, exploration, and skill transfer," *International Journal of Social Robotics*, Jan 2024.

[58] K. Bouyarmane and A. Kheddar, "On weight-prioritized multi-task control of humanoid robots," *IEEE Transactions on Automatic Control*, vol. 63, no. 6, pp. 1632–1647, 2018.

[59] 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 *IEEE-RAS International Conference on Humanoid Robots*, 2019, pp. 25–32.

[60] 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 *IEEE-RAS 17th International Conference on Humanoid Robotics*, 2017, pp. 432–437.

Antonin Dallard received an Engineering Master degree in mechanical and Industrial from Arts Et Métiers Institute of Technology. In 2020, he started a Ph.D on the topic of Humanoid robot teleoperation and locomotion at the CNRS-University of Montpellier, LIRMM in France and at the CNRS-AIST Joint Robotics Laboratory, Tsukuba in Japan.

Mehdi Benallegue holds an engineering degree from the National Institute of Computer Science (INI) in Algeria, obtained in 2007. He earned a masters degree from the University of Paris 7, France, in 2008, and a Ph.D. from the University of Montpellier, France, in 2011. His research took him to the Franco-Japanese Robotics Laboratory in Tsukuba, Japan, and to INRIA Grenoble, France. He also worked as a postdoctoral researcher at the Collège de France and at LAAS CNRS in Toulouse, France. Currently, he is a Research Associate with CNRS AIST Joint robotics Laboratory in Tsukuba, Japan. His research interests include robot estimation and control, legged locomotion, biomechanics, neuroscience, and computational geometry.

Nicola Scianca received his Ph.D. in control engineering from Sapienza University of Rome, Italy, in 2020. He is currently a Researcher at the Department of Computer, Control and Management Engineering (DIAG). In 2019, he was a Visiting Student with the Model Predictive Control Laboratory, University of California at Berkeley, CA, USA. His research interests include model predictive control for humanoid robots.

Fumio Kanehiro received the BE, ME, and PhD in engineering from The University of Tokyo, Japan, in 1994, 1996, and 1999, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science in 1998-1999. In 2000, he joined the Electrotechnical Laboratory, Agency of Industrial Science and Technology, Ministry of Industrial Science and Technology (AIST-MITI), later reorganized as National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. From April 2007, he was a visiting researcher at the LAAS-CNRS for one year and three months. He is currently Director of CNRS-AIST JRL (Joint Robotics Laboratory), IRL, AIST. His research interests include the software frameworks and whole body motion planning of humanoid robots.

Abderrahmane Kheddar (F'22) received the BS in Computer Science degree from the Institut National d'Informatique (ESI), Algiers, the MSc and PhD degree in robotics, both from Pierre et Marie Curie, Sorbonne University, Paris. He is presently Directeur de Recherche at CNRS. His research interests include haptics and humanoids at large. He is Editor of the IEEE Robotics and Automation Letter, Editor of the IEEE Transactions on Robotics (2013-2018) and a founding member of the IEEE Transactions on Haptics. He is an IEEE Fellow, AAIA Fellow, and member of the National Academy of Technology of France.