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 extended dynamics of the inverted pendulum. Our scheme includes re-planning of the steps locations and steps 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 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 have a great amount of variety, e.g., [1], [2] (see also Sec. II). Indeed, the complexity of this problem lies in the interlink between different bricks consisting of (i) reactive footstep planning, (ii) whole-body switched control under tight balance constraints, (iii) stability of the control itself (as understood in control theory), and finally (iv) perception to understand the surrounding environment and the bipedal robot state within it.

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 [3] and popularized by the first Honda humanoids family (see knowledge reports in [4], [5], [6], [7]). It is still successfully implemented in many humanoids use-cases, yet it has evolved in numerous shades and sophistications (see Sec. II).

The common implementation of such centroidal models is to reason on a simplified dynamic model, i.e., the dynamics around the center of mass (CoM), to draw fast dynamic planning and control that usually implies the exploitation of the zero moment point (ZMP) location for dynamic balancing.

Manuscript received June 30, 2023; revised Month XX, 202X; 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 X. Xxxxxxx 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. 1: HRP-2KAI and HRP-4 robots walking outdoors. Experiments with the HRP-4 are done without safety ropes.

Indeed, the ZMP gives a criterion for maintaining balance on the surface in contact with the ground; namely, it must be maintained at all times 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 [8]) or the input (the LIP mode) for a linear dynamic system linking it to the CoM. However, the instability of the model entails that even if the ZMP is inside the support polygon, the associated CoM trajectory might still be divergent.

The Intrinsically Stable MPC (IS-MPC) [9] is a walking gait generator, derived from the LIPM, which incorporate stability constraint explicitly: the presence of this constraint allows to prove recursive feasibility. This means that it is always possible to find a solution satisfying the constraints, and internal stability, i.e., the CoM trajectory is always bounded with respect to the ZMP. This stability constraint is also used for footsteps adaptation (directly in the MPC) or steps timings adjustments (externally) [10].

However the LIPM being too simplistic, it fails in grasping the complexity of the whole multi-body dynamics. Therefore, one may expect to deal with important discrepancies between the centroidal and the whole-body dynamics. As the inverted pendulum is derived from the simplified equation of the centroidal dynamics, controlling the pendulum means controlling the contact forces applied on the robot. Different grounds and control methods obviously result in different behaviors (see

Sec. II). To take these discrepancies into account, it is common to use a correction policy that mitigate the modeling gaps in using the real robot state and a model of the behavior of the contact forces, see e.g., [11]. Such a mitigation is commonly called the *stabilizer* in the humanoid research jargon. The stabilizer usually has several parameters and tuning it is usually not trivial, especially when combined with the controller's parameters.

Using such a stabilizer, one disconnects the gait generator from the real robot centroidal state as the contact forces (i.e., the ZMP) are updated, which changes the CoM dynamics. Moreover, this correction policy is short-sighted compared to the MPC policy which is based on state prediction, and might invalidate some of the benefits of having such prediction, this is well detailed in Sec. III.

One might think that MPC should take care of this issue thanks to its inherent robustness. However, the robustness of MPC comes from the fact that it is possible to close the loop, i.e., the predicted trajectory is recomputed at each time starting from the measured state. Yet, if the contact force model is not taken into account, the generated walking gait would still be not feasible; and hence, a stabilizer would still be necessary.

This paper proposes a solution to this issue and present a humanoid walking control scheme with the following newies:

1) A reformulation of the IS-MPC that allows to completely and effectively close the loop on the robot state and does not need external correctional term (the so-called *stabilizer*) to enforce and warrant balance, Sec. IV;
2) An horizon-based force distribution scheme to regulate the contact forces during double-support phases, Sec. V;
3) A pendulum feasibility solver based on a demonstrated extension of the stability constraint that allows to adapts the footsteps location and timing using the real state of the robot, Sec. VI;
4) An assessment of our method in extensive and challenging scenarios using five different humanoid robots, all of them controlled with the same software to illustrate the robustness, the versatility and the portability of our algorithms, Sec. VII

The control scheme is organized according to Fig 2 on which for each of the contribution aspect: the order of execution, the sections and the equation that formulate the optimization problem is displayed.

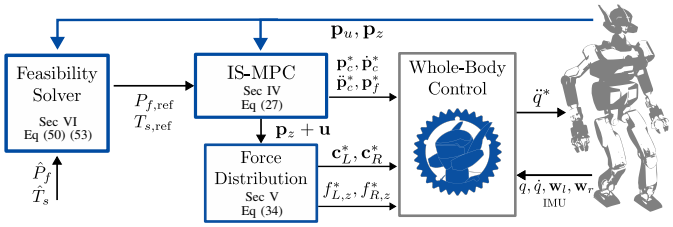


Fig. 2: Global control scheme, the $*$ super script represent the control references for the whole body-control or for the robot. The swing foot trajectory planning or the force control scheme is included in the whole-body control box as they are outside from the article contribution. The blue elements highlight our adding w.r.t existing approaches.

II. Background

Bipedal and humanoid locomotion is a rich field of research in the robotics domain. It would be pretentious to claim covering all its facets in a single section. We invite interested readers to refer to the excellent reviews in [1], [2], [12], [13], some of which are recent, and the monographs [14], [8].

We put aside all the work done in passive bipedal locomotion [15] (i.e., non actuated) as we are interested in motorized humanoid robots that do not content in predefined cycled locomotions[1]. Three schools of methodologies emerge in the bipedal locomotion domain: (i) the school of walking using hybrid-systems control approach; (ii) the school of data-driven and machine learning walking control, and (iii) the school of pendulums-based walking control so called abusively ZMP-based school. Our work lies in the last category that will be given a more thorough synthesized update.

Hybrid zero dynamics (HZD) based bipedal walking consists in projecting the whole-body humanoid dynamics on its zero submanifold and designing feedback control based on virtual constraints, see e.g., [16], [17], [14]. A comparative analysis between centroidal pendulum-based approaches (we abusively termed ZMP school) and feedback control design based on virtual constraints is reported in [12]. Impressive walking skills have been demonstrated on the biped MABEL having compliant transmission [18], on the humanoid DURUS in [19], and on Cassie and the humanoid Digit[2] using HZD and its variants in, e.g., [12]. Compared to the so-called ZMP school, HZD formalism provides stability proof (with exponential convergence rate) of the walking controller scaling from the reduced model up to the whole body dynamics. Moreover, HZD integrates impacts explicitly in the models [20] and shows possible walks using nearly point contacts.

Data-driven and machine learning bipedal walking originates in the computer graphics and animation community. Since the early 2000th ACM SIGGRAPH published noticeable papers showing high dexterity skills in bipedal walking of animes from human-captured data, e.g., [21], [22], [23], [24], to cite a few. Years ago, we have tried to port some of the promising approaches on our humanoid robots, yet we failed because computer animation methods often do not consider hard constraints found in the robotics domain. The major one is sensors noise and model uncertainties of all kinds. Others are, for some of the computer graphics and animation approaches, the lack of torque bounds, tolerance to auto-collisions and collisions, lack of non-smoothness considerations (such as plausible contact models and impacts), etc. Nowadays however, the emergence of SimToReal research (that allow generating millions of reliable simulations including dynamics and noise close to real physics), e.g., [25], new robust techniques in machine learning, deep learning and learning techniques from few experiences... have allowed

[1]This research is however useful to have insights in motorized gait energy optimization.

[2]Jonathan W. Hurst, *Building robots that can go where we go*, IEEE Spectrum, 26 February 2019.

promising transfer of data-driven approaches to real robotic bipedal implementation. Yet very successful ones are few: that are the amazing results obtained with Cassie, e.g., [26], [27] and on quadrupeds, e.g., [28]. It should be noted that the successful teams, are those that also have *a priori* very good grasp in model-based control approaches implemented on real robotic platforms demonstrating rich dynamic walking behaviors[3]. In our laboratory, recent results in robust locomotion by means of reinforcement-learning on highly reduced robots without torque feedback [29].

Yet, in our team, we have been mainly focusing on LIPM based walking initiated by [30]. The main idea consists in computing the bounded dynamics of the CoM that is feasible, i.e., the one that enforces the ZMP to lie in the support polygon of the current and future contacts in order to guarantee dynamic balance of the walk. In what follows, we synthesize the background on which our method is built.

Our work relies on the contribution of [9], therefore we kept the notations as close as possible to the original one, that is:

- Super-scripts $x^{\square}$ refer to indexes in a sequence, when it designates the power of x there will be no possible confusion;
- 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 and y variables refer to components of a position in the x and y axis respectively. The axis component can be specified in some cases on the lower-script without possible confusion.

A. Inverted Pendulum Mode

The LIP model is derived using the Newton Euler equations under the assumption that the angular momentum around the CoM is constant, and that the CoM is at a constant height. It leads to a relationship between the CoM position $\mathbf{p}_c = (x_c, y_c)^T$ and the ZMP pose $\mathbf{p}_z = (x_z, y_z)^T$ such as in the (x,y) plane (corresponding to the ground):

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

where $\omega^2 = g/\bar{z}_c$ with g the gravitational acceleration magnitude and $\bar{z}_c$ the constant CoM height.

This model was initially introduced for humanoid locomotion by [30] using a LQ preview control to generate a CoM trajectory under a pre-generated ZMP objective trajectory. The reformulation to a Model Preview Control initially done in [31] allowed to extend the decision variables to the CoM and the ZMP trajectories including constraints: on the ZMP to maintain contact stability but also on the steps to keep their kinematic feasibility [9], [32], [33].

Using the following change of variable:

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

Equation (1) can be rewritten as

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

[3]This remark was pointed by Dr Steve Heim in a private discussion.

where x_u is the Divergent Component of Motion (DCM), introduced in [4]; its dynamic is such that, while the CoM is converging exponentially towards the DCM, this latter is diverging exponentially from the ZMP. Controlling this 1st order dynamics between the ZMP and the DCM has been used extensively in walking gait generation, e.g., [33], [34], [9], [10], as it simplifies the dynamics and extends the stability criterion for the gait generation.

Using the DCM, having a non diverging LIP dynamics, results in finding the ZMP trajectory such that the DCM remains bounded.

This trajectory exists and must fulfill the following condition w.r.t the current DCM, see details in [35].

$$x_u^0 = x_u^\star(\tilde{t}, x_z) \triangleq \omega \int_{\tilde{t}}^{\infty} x_z(\tau) \mathrm{e}^{-\omega(\tau-\tilde{t})}\,\mathrm{d}\tau \tag{4}$$

with e being the exponential function, i.e., $\mathrm{e}^{(\cdot)} \triangleq \exp(\cdot)$, then $\mathrm{e}^1 = e$; and $x_z(\tau) = 0$ for $\tau < \tilde{t}$.

Equation (4) is referred to as the *stability condition*. Instead of dealing with the instability of the CoM/ZMP dynamics by penalizing divergent trajectories inside the cost function (commonly minimizing the CoM jerk) which only gives a heuristic solution, and can be heavily dependent on the duration of the prediction window of the MPC, using Eq. (4) guarantees the boundness of the CoM (if the ZMP or its velocity is bounded) at all time.

Moreover, one can see in this condition more interesting points:

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 steps location and the steps timings parameters (i.e., for both double and single support durations), the stability condition also captures the feasibility of a walking plan.

This simplified and low-order dynamical system (DCM/ZMP) makes the LIPM an appropriate candidate for walking gait generation using MPC either linear or not (i.e., NMPC), e.g., [36], [37].

However, the LIPM remains a constrained model of the CoM dynamics. Extensive work has therefore been done to get rid of those constraints by controlling the full centroidal model, e.g. [38], or by controlling a variable height pendulum, e.g., [39], [40], and its angular momentum, e.g. [41].

Once it comes to control a real robot under the LIPM assumptions, one must guarantee that the robot dynamics will follow the generated gait. We therefore need to mitigate modeling uncertainties, that are prone to errors, and disturbances. To do so, a complementary policy based on the DCM tracking feedback can be added to the control of the ZMP in order to enforce the DCM to converge exponentially to its planned reference, e.g., [42], [11]. This DCM feedback control loop is often named *stabilizer* as its purpose is to *stabilize* the reference walking gait. The addition of such DCM feedback control concedes the open-loop aspect of the gait generators we previously mentioned as we 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 [9], [31], only part of the state is provided to

the planning as the ZMP dynamics from the force control is not considered and the need for such complementary policy is still needed.

B. Force control

With a position-controlled robot, the forces at the feet are regulated using an admittance control that tracks a reference wrench $\mathbf{w}_f^r$ by displacing the feet in the opposite direction of the desired force. Robot links are controlled using an acceleration based tracking law [43],

$$\ddot{\mathbf{p}}_f = -K_d(\dot{\mathbf{p}}_f - \dot{\mathbf{p}}_f^r) - K_p(\mathbf{p}_f - \mathbf{p}_f^r) + \ddot{\mathbf{p}}_f^r \tag{5}$$

where $\mathbf{p}_f$ is the 6D pose of the robot foot link, K_p and K_d are respectively the stiffness and damping coefficient. In the case of force control, we set $\mathbf{p}_f^r$, $\dot{\mathbf{p}}_f^r$ and $\ddot{\mathbf{p}}_f^r$ such that [43]:

$$\dot{\mathbf{p}}_f^r = K_a(\mathbf{w}_f^m - \mathbf{w}_f^r) \tag{6}$$

where $\mathbf{w}_f^r$, $\mathbf{w}_f^m$ are the reference and measured wrench in the link frame respectively; K_a is the admittance gain.

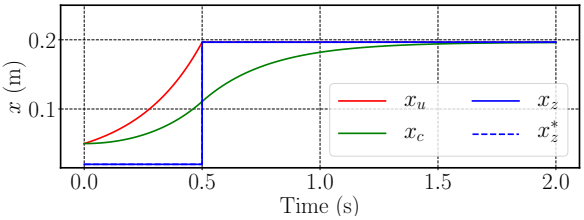
In order to control the robot ZMP, we must control the Center of Pressure (CoP) of the currents contacts. This is done by controlling the moment in the x and y direction in the frame of the contact and the vertical force. To control the moments, we use Eq. (6) in two dimensions. However, the control of the vertical forces during a double support phase is done using a foot force difference control (FFDC) [44], [45]. Noting $(\dot{p}_{Lz}, \dot{p}_{Rz})$ the vertical velocity of the left and right foot respectively, FFDC with gain K_z updates the reference velocity $(\dot{p}_{Lz}^r, \dot{p}_{Rz}^r)$ [11] as follows:

$$\dot{p}_{Lz}^r \leftarrow \dot{p}_{Lz}^r - 0.5\dot{p}_{\delta f_z} \tag{7}$$

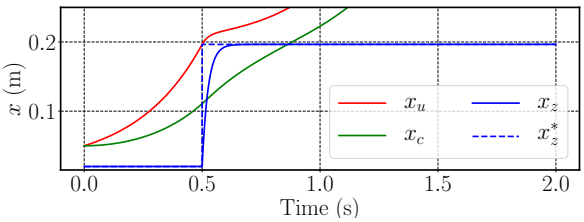
$$\dot{p}_{Rz}^r \leftarrow \dot{p}_{Rz}^r + 0.5\dot{p}_{\delta f_z} \tag{8}$$

$$\dot{p}_{\delta f_z} \triangleq K_z\left[(f_{Lz}^r - f_{Rz}^r) - (f_{Lz}^m - f_{Rz}^m)\right] \tag{9}$$

When it comes to evaluate 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 [46]. Moreover, even in a configuration without a shock absorbing mechanism, a 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 reflect the behavior between the desired CoP and the references forwarded into the force control scheme. A simpler model empirically evaluate the behavior between the CoP and its reference (using a force control defined in Eq. (6)) by means of a 1st order dynamics [44].



(a) LIPM without ZMP bounds: the DCM can be captured by placing the ZMP on it.



(b) When the ZMP dynamics is introduced, the DCM cannot be captured.

Fig. 3: LIPM behaviors for the toy example.

III. 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 dynamics of a perfect pendulum model in one dimension.

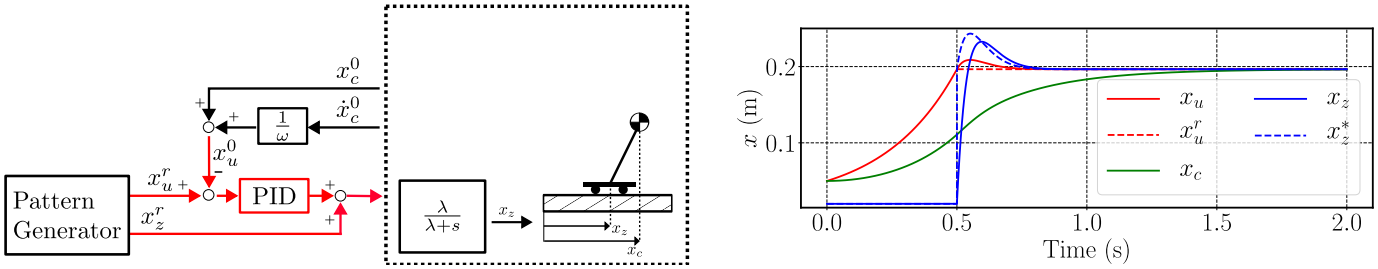
We initialize the pendulum with an arbitrary and distinct DCM and ZMP position at initial time $t = 0$ and must choose the ZMP pose at $t = \Delta t$ (here $\Delta t = 0.5s$) to bring the pendulum to a stop. Since it is a toy example we assume that the final position of the pendulum does not matter, but the same reasoning extends to the case where it does.

Using Eq. (4), having $x_z(\Delta t) = x_u(\Delta t)$ stabilizes the pendulum, see Fig. 3a; here, the ZMP jumps instantly to the DCM pose. This is however unrealistic, because the floor is never perfectly stiff and the control induces delays, especially for position-controlled robots. Previous work tends to consider the ZMP behavior $x_z(t)$ to track its reference x_z^* with at least a linear first order model [44], that is, with a parameter λ:

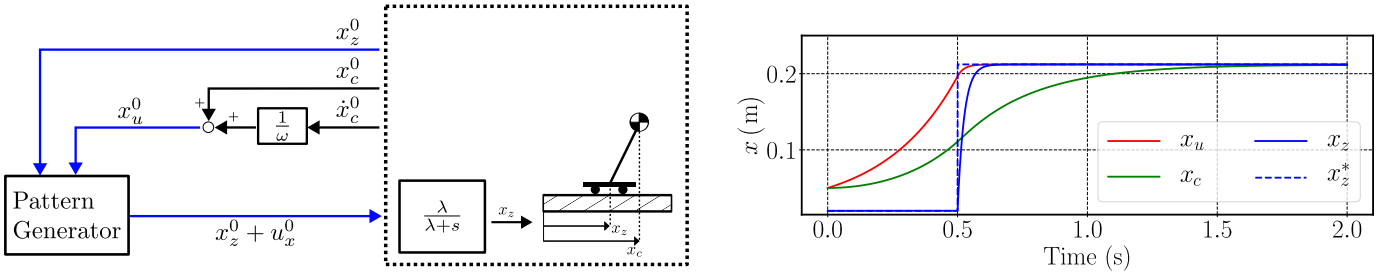
$$\dot{x}_z(t) = -\lambda(x_z(t) - x_z^*) \tag{10}$$

Introducing these ZMP dynamics inside our LIP model, by setting $x_z^* = x_u(\Delta t)$, leads to instability (see Fig. 3b). This is because the delays in ZMP results in a deviation of the DCM; subsequently the pendulum's dynamics is not captured at its initial position.

To account for this dynamics, one could correct the ZMP reference using a DCM feedback policy; which could be a simple proportional-integral-derivative (PID) control policy (or at least proportional) 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) [11]. This policy has for downside 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



(a) Using a DCM feedback control $x_z^* = x_u^r + K(x_u - x_u^r)$, (with $K > 1$), we can correct the ZMP reference to be stable.



(b) If the dynamic of the ZMP is known, the DCM can be captured using Eq. ((4)).

Fig. 4: Linear inverted pendulum models control schemes.

the real robot which could lead to breaches in the planning constraints.

> Our solution relies on using the dynamics of the ZMP directly and explicitly inside the walking pattern generation, see Fig. 4b. In our example, this amounts at solving Eq. (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 the one we use is illustrated in Fig. 4. It can be noted that in real-case applications, the 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 the need of a DCM feedback policy to guarantee a proper tracking of the pendulum state.

IV. Walking Gait Generation by Closed Loop IS-MPC

The IS-MPC generates a feasible CoM gait. This gait is computed to follow a *footstep plan*. This footstep plan is made of the footsteps print (which includes the location and the orientation) and the footsteps duration over a preview horizon longer than the MPC control horizon. We note T_p the horizon duration of the footstep plan.

The initial description of the IS-MPC uses the LIP mode by taking $\dot{x}_z$ (or equivalently $\dot{y}_z$) as an input term, we can then define the following dynamics from Eq (1) with the state $(x_c, \dot{x}_c, x_z)$

$$\begin{pmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{x}_z \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_c \\ \dot{x}_c \\ x_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{x}_z \tag{11}$$

However, in order to use this MPC in closed-loop on the robot pendulum state, in terms of ZMP and DCM, we must model the relation between the ZMP reference forwarded to the force control (in our case the admittance control) and the measured ZMP value. We model this behavior as a first order system similarly to Eq. (10) with a parameter λ and with a delay δ_d such that:

$$\dot{x}_z = -\lambda(x_z(t) - x_z^*(t - \delta_d)) \tag{12}$$

The reason behind adding the delay 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 actually applied. It includes software latency and joint torque tracking response time. The dynamics in Eq. (11) can then be rewritten as:

$$\begin{pmatrix} \dot{x}_c \\ \ddot{x}_c \\ \dot{x}_z \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & -\lambda \end{pmatrix} \begin{pmatrix} x_c \\ \dot{x}_c \\ x_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \lambda \end{pmatrix} x_z^*(t - \delta_d) \tag{13}$$

These dynamics are of course the same for the y-axis.

Let's consider this system at an instant t^0 and analyze its future dynamics. The reference $x_z^*(t)$ is the input to the force control loop, so let's consider it piece-wise constant on a duration Δt such as $\Delta t > \delta_d$. We define inputs u_x^i, on an interval $[t^i; t^{i+1}]$ where $t^i = t^0 + i\Delta t$, as increments added to the ZMP reference x_z^* such that on $[t^i; t^{i+1}]$:

$$\begin{aligned} x_z^*(t) &= x_z^{0d} + \sum_{k=0}^{i} u_x^k \\ x_z^{0d} &= x_z^0 + (x_z^{0*} - x_z^0)(1 - e^{-\lambda \delta_d}), \end{aligned} \tag{14}$$

with x_z^0 being the current estimated robot's ZMP and x_z^{0*} being the current commanded ZMP applied to the robot. We note $t_d^0 = t^0 + \delta_d$, $t_d^i = t^0 + \delta_d + i\Delta t$. The ZMP dynamics in Eq (12)

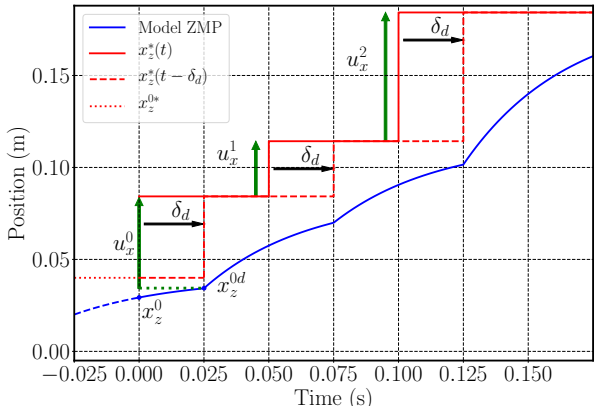


Fig. 5: Chosen 1st order ZMP model with $\lambda = 25$, $\delta_d = 0.025$, $\Delta t = 0.05$. The green dashed line indicates the starting point of the first arrow.

can then be integrated with Eq (14) to obtain the following ZMP trajectory for $t \in [t_d^i; t_d^{i+1}]$:

$$x_z(t) = x_z^{0d} + \sum_{k=0}^{i} u_x^k \left[1 - \mathrm{e}^{-\lambda(t-t_d^k)}\right] \tag{15}$$

One can see each $u_\square^k$ 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 $\mathbf{u}^k = (u_x^k, u_y^k)^T$ and the future contact location as $\mathbf{p}_f^i = (x_f^i, y_f^i)^T$ where i is the step number. At each iteration of the proposed closed loop IS-MPC considers a control horizon of C samples of Δt duration each such that this horizon contains N_s steps, and generates the following values:

1) The reference inputs $U = (\mathbf{u}^0, \ldots, \mathbf{u}^{C-1})^T$;
2) The footsteps locations $P_f = (\mathbf{p}_f^1, \ldots, \mathbf{p}_f^{N_s})^T$ under their imposed time parameters $T_{s,\mathrm{ref}} = \left[(t_{\mathrm{ds,ref}}^0, t_{\mathrm{s,ref}}^0), \ldots, (t_{\mathrm{ds,ref}}^{N_s-1}, t_{\mathrm{s,ref}}^{N_s-1})\right]$ which are the time of the start and end of the single support duration. Figure 6 shows the organization of the time variables. $P_{f,\mathrm{ref}}$ are the reference footsteps location. We note $\mathbf{p}_f^0 = (x_f^0, y_f^0)^T$ the current support foot position and $\mathbf{p}_f^{-1}$ the current swing foot location in double support.

The control horizon time is then $T_c = C\Delta t$. Once computed, the LIP dynamics defined in Eq. (13) is integrated under the computed input (u_x^0, u_y^0) at a sampling rate $\delta t \leq \Delta t$; δt represents the sampling period of the robot whole body control loop.

The MPC solves a quadratic optimization problem (described Eq. (27)) under different linear constraints to enforce dynamic balance.

A. Constraints

1) ZMP location constraints: At every sample k, i.e., $t = t^0 + k\Delta t$, $\mathbf{p}_z^k = (x_z^k, y_z^k)^T$ must be enforced to be within a constraint polygon –being itself a sub-part of the robot support polygon, located either under the support foot if t is during a single support phase or in-between both feet if t is during

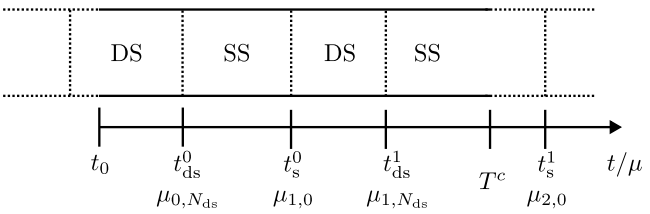


Fig. 6: Disposition of the planning time variables.

a double support phase. Therefore, for each sample k we can define $N_k = N(t)$ and $O_k = O(t)$, which are the $n_k \times 2$ normal matrix and the $n_k \times 1$ offset vector respectively. This gives the following constraint:

$$N_k \mathbf{p}_z^k \leq O_k, \tag{16}$$

which constrains the ZMP to be within a n_k vertices convex polygon.

Note that since N_k and O_k define an admissible ZMP position over the horizon, they depend on the position of the feet and thus on the decision variable P_f. However having N_k and O_k define the whole support area, makes the constraint (16) nonlinear with respect to P_f.

Nevertheless, one important exception is the support phase at the current time t^0. Indeed, at the current double support and first single support phases, the ZMP constraint polygon is completely defined by the current robot foot poses and does not depend on control decision variables. Therefore, the first double support constraint can be as large as possible, and for instance be a polygonal approximation of the convex hull of the full footprint.

To keep the linearity of the the next ZMP constraints in the horizon, the ZMP allowed regions are simplified as a rectangle of dimension (dx, dy) sliding from one foot to the other. Between footstep $i-1$ and i, $O(t)$ is discretized in time to define $O_{i,j} = O(t_{\mathrm{s}}^{i-1} + j\Delta t)$ which can be expressed in the interval $[t_{\mathrm{s}}^{i-1}, t_{\mathrm{s}}^i]$ as:

$$O_{i,j} = \frac{1}{2}\begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + N_{i_j}\left(\alpha_{i_j}\mathbf{p}_f^i + (1-\alpha_{i_j})\mathbf{p}_f^{i-1}\right) \tag{17}$$

$$\alpha_{i,j} = \frac{j\Delta t}{t_{\mathrm{ds}}^i - t_{\mathrm{s}}^{i-1}}, \tag{18}$$

with $0 \leq \alpha_{i,j} \leq 1$ is the weighting variable allowing to the rectangle to slide from a contact to the other.

The ZMP constraints being set as a rectangle, the normal matrix of the ZMP constraints are then defined as follows:

$$N_{i,j} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \\ -1 & 0 \\ 0 & 1 \end{pmatrix} R_{i,j} \tag{19}$$

with $R_{i,j} = R(t_{\mathrm{s}}^{i-1} + j\Delta t)$ a 2×2 rotation matrix of a chosen angle around the vertical axis that sets the orientation of the

rectangle. This orientation (in double support) is set by a linear interpolation of the orientation between the planned steps $i-1$ and i such as $R_{i,0}$ is the orientation of the step $i-1$ and $R(t^i_{\text{ds}})$ is the orientation of the step i. In single support, the orientation of the rectangle is the one of support foot i.

2) Footsteps constraints: The footsteps locations should be kinematically reachable by the robot, therefore the difference between two consecutive footsteps locations must remain bounded.

This can be defined by using N_{kin} and O_{kin} the 4×2 normal matrix and the 4×1 offset vector of the step kinematic constraints such that

$$N_{\text{kin}}(\mathbf{p}^i_f - \mathbf{p}^{i-1}_f) \leq O_{\text{kin}} \tag{20}$$

$$O_{\text{kin}} = \frac{1}{2}\begin{pmatrix} dx_f \\ dy_f \\ dx_f \\ dy_f \end{pmatrix} \tag{21}$$

with N_{kin} the normal matrix of a rectangle oriented according to $\mathbf{p}^{i-1}_f$ and therefore is defined as in Eq. (19)

3) Stability constraint: As we compute $x_z(t)$ until T_c, one must make an assumption over the ZMP values beyond the control horizon. They will be referred to as $\tilde{x}_z(t)$ which are defined for $t > T_c$ and represent the *tail* of the trajectory over the horizon [9]. Then, the stability constraint defined in Eq. (4) becomes:

$$x^0_u - \tilde{c}_x - c_{x,d} = \omega \int_{t^0_d}^{t^0_d+T_c} x_z(\tau)\mathrm{e}^{-\omega(\tau-t^0)}\,\mathrm{d}\tau \tag{22}$$

with

$$\begin{aligned} \tilde{c}_x &= \omega \int_{t^0_d+T_c}^{\infty} \tilde{x}_z(\tau)\mathrm{e}^{-\omega(\tau-t^0)}\,\mathrm{d}\tau \\ c_{x,d} &= \omega \int_{t^0}^{t^0_d} x^0_z + (x^{0*}_z - x^0_z)\left[1-\mathrm{e}^{-\lambda(\tau-t^0)}\right]\mathrm{e}^{-\omega(\tau-t^0)} d\tau \end{aligned} \tag{23}$$

We can also write equivalently using Eq. (4):

$$x^\star_u(t^0, x_z) = c_{x,d} + \omega \int_{t^0_d}^{t^0_d+T_c} x_z(\tau)\mathrm{e}^{-\omega(\tau-t^0)}\,\mathrm{d}\tau + \tilde{c}_x \tag{24}$$

In our case, substituting the behavior of $x_z(t)$ inside Eq. (15) in Eq. (22), this condition becomes:

$$x^0_u - \tilde{c}_x - c_{d,x} = \mathrm{e}^{-\omega\delta_d}\left[x^{0d}_z + \frac{\lambda}{\lambda+\omega}\sum_{k=0}^{C-1} u^k_x \mathrm{e}^{-\omega k\Delta t}\right] \tag{25}$$

4) Tailing: Refers to the behavior of $\tilde{x}_z(t)$ as this latter can follow different behaviors depending on the expected future motion, see [9] for more details.

- *Truncated* tailing is setting a ZMP to a stop after the control horizon, therefore:

$$\forall i \geq C,\ \ u^i_x = 0,\ \ \tilde{c}_x = 0.$$

- *Periodic* tailing is expecting that the ZMP trajectory will periodically repeat every T_c, therefore, $\tilde{x}_z(t)$ follows inputs $\tilde{u}^j$, that is $\tilde{u}^j_x = u^i_x$, where j is congruent to i modulo C.

We can then rewrite the stability condition as:

$$x^0_u - c_{d,x} = \mathrm{e}^{-\omega\delta_d}\left[x^{0d}_z + \frac{\lambda}{(\lambda+\omega)(1-\mathrm{e}^{-\omega C\Delta t})}\sum_{k=0}^{C-1} u^k_x \mathrm{e}^{-\omega k\Delta t}\right] \tag{26}$$

- *Anticipative* tailing uses the information about the part of the footstep plan which is defined over T_c to generate a reference ZMP path. It allows $\tilde{x}_z(t)$ to follow the desired behavior from the walking plan and thus extends the preview horizon without increasing the number of decision variables. Beyond that extended horizon the truncated or periodic approaches remain necessary to complete the tail. More details are in [9].

In all cases, Eq. (25) defines a linear equality constraint in terms of the control decision variables.

B. Cost Function

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

1) ZMP trajectory objective: The ZMP objective $x_{z,\text{obj}}$ in the cost function is located on the current support foot for each step. This means for the future steps location in the horizon, $x_{z,\text{obj}}$ is defined by the steps location decision variables of the MPC. An offset can be added to the value of $x_{z,\text{obj}}$ to handle cases where the modeled ZMP target if not perfectly centered within the support foot. For example, having a ZMP target placed closer to the interior of the foot will limit the lateral variation of the CoM yet will make the balance less robust in case of disturbances. We also have a component of the ZMP velocity magnitude with an objective velocity set as zero.

2) DCM trajectory objective: To generate a DCM objective trajectory, we compute the stable DCM $x^\star_u(t^0, x_{z,\text{obj}})$ corresponding to the objective ZMP trajectory and integrate it under the LIPM using Eq. 11 in the control horizon with the ZMP objective trajectory $x_{z,\text{obj}}$ as the input. This also allows us to obtain a DCM velocity objective using the DCM dynamics in Eq. (3).

3) Steps location objective: This one simply aims to have the steps location decision variable as close as possible to the references ones.

To summarize, the decision variable in the horizon being:

$$U = (u^0_x, u^0_y, \ldots, u^{C-1}_x, u^{C-1}_y),\ P_f = (x^1_f, y^1_f, \ldots, x^{N_s}_f, y^{N_s}_f)$$

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

$$\begin{aligned} U, P_f = \arg\min\ & \beta_{\dot{z}}\|\dot{P}_z\|_2 + \beta_z\|P_z - P_{z,\text{obj}}\|_2 + \\ & \beta_f\|P_f - P_{f,\text{ref}}\|_2 + \beta_u\|P_u - P_{u,\text{obj}}\|_2 + \\ & \beta_{\dot{u}}\|\dot{P}_u - \dot{P}_{u,\text{obj}}\|_2 \end{aligned} \tag{27}$$

under the constraints:

- ZMP position constraints in Eq. (16);
- Footsteps kinematics constraints in Eq. (20);
- Stability constraints in Eq. (25).

V. Robot force control

The pattern generated by the scheme described in Sec. IV is forwarded to a whole-body task-space controller in which we regulate the CoM trajectory and the contact forces to achieve the desired walking dynamics.

A. CoP regulations

As the robot ZMP dynamic $\mathbf{p}_z(t)$ is modeled as a delayed 1st order system w.r.t the computed ZMP reference $\mathbf{u}^0$ (computed in the IS-MPC in Sec. IV), the admittance control shall apply a reference net wrench to obtain the following ZMP behavior in the interval $[t_d^0; t_d^0 + \Delta t]$ according to Eq. (15).

$$\mathbf{p}_z(t) = \mathbf{p}_z^{0d} + \left[1 - \mathrm{e}^{-\lambda(t-t^{0d})}\right]\mathbf{u}^0 \tag{28}$$

During the single support phase, the admittance control can directly apply a wrench to track a reference of the contact's CoP at $\mathbf{p}_z^{0d} + \mathbf{u}^0$. Thanks to the delays and the ZMP dynamics, this reference can temporarily be set to be outside the contact polygon, in order to increase real ZMP velocity.

During the double support phase, one must distribute the reference wrench along the contacts. We therefore need to compute for each contact a reference CoP. We assume that each contact's reference CoP has its own 1st order dynamic w.r.t the real contact CoP with parameters λ_c; ideally $\lambda_c \simeq \lambda$. We note $\mathbf{c}_{L,u}, \mathbf{c}_{R,u}$ and $\mathbf{c}_L, \mathbf{c}_R$ respectively the reference CoP and the modeled CoP for the left and right contact. This model, with a constant reference over $[t_d^0; t_d^0 + \Delta t]$, can be developed similarly to (15) to obtain:

$$\begin{aligned} \mathbf{c}_L(t) &= \mathbf{c}_{L,u} + (\mathbf{c}_L^{0d} - \mathbf{c}_{L,u})\mathrm{e}^{-\lambda_c(t-(t^0+\delta_d))} \\ \mathbf{c}_L^{0d} &= \mathbf{c}_L^* + (\mathbf{c}_L^0 - \mathbf{c}_L^*)\mathrm{e}^{-\lambda_c\delta_d} \end{aligned} \tag{29}$$

Additionally, having the same mode for $f_{L,z}$ and $f_{R,z}$, the vertical forces on the left and right foot, gives:

$$\begin{aligned} f_{L,z}(t) &= f_{L,u,z} + (f_{L,z}^{0d} - f_{L,u,z})\mathrm{e}^{-\lambda_f(t-(t^0+\delta_d))} \\ f_{L,z}^{0d} &= f_{L,z}^* + (f_{L,z}^0 - f_{L,z}^*)\mathrm{e}^{-\lambda_f\delta_d} \end{aligned} \tag{30}$$

And similarly for the right component. We then must set $\mathbf{c}_L^u, \mathbf{c}_R^u$ such that it produces the same overall ZMP, that is:

$$\frac{\mathbf{c}_L(t^0+\Delta t)f_{L,z} + \mathbf{c}_R(t^0+\Delta t)f_{R,z}}{f_{L,z}+f_{R,z}} = \mathbf{p}_z(t^0+\Delta t). \tag{31}$$

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 are usually infinite ways to distribute the vertical forces and the CoPs in order to get a desired ZMP. To solve this 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 robot mass. To get the values of $f_{L,z}$ and $f_{R,z}$. We define $r \in [0;1]$ as the normalized projection of the vector $\mathbf{p}_z(t^0+\Delta t) - \mathbf{p}_L$ over the vector $\mathbf{p}_R - \mathbf{p}_L$ and set:

$$\begin{aligned} f_{R,z} &= rmg \\ f_{L,z} &= (1-r)mg \end{aligned} \tag{32}$$

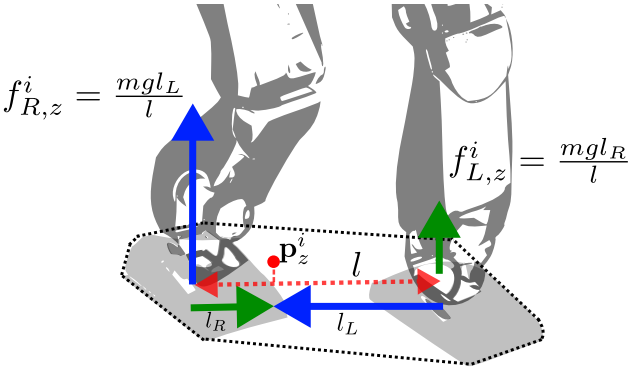


Fig. 7: Vertical force distribution depending on the desired ZMP position at time t_i.

Figure 7 shows a visual explanation of how the desired vertical forces are computed. The reference vertical forces for each legs are set using Eq. (30) with parameters λ_f.

To obtain $\mathbf{c}_{L,u}, \mathbf{c}_{R,u}$, one can formulate an optimization problem to approach quadratically Eq. (31) under the constraints that the modeled CoP shall remain inside the contact polygon made of the normals matrix N_{Lc}, N_{Rc} and the offset vectors O_{Lc}, O_{Rc} for the left and right contact respectively. Moreover, as we have an horizon over the future ZMP location, this problem can be extended to compute the force distribution in the entire horizon of the double support phase of length $t_{\mathrm{ds}}^0 - t^0$.

We note $\mathbf{c}_L^i$ and $\mathbf{c}_{L,u}^i$ (and respectively $\mathbf{c}_R^i$,$\mathbf{c}_{R,u}^i$) the CoP at time $t^i = t^0 + i\Delta t$ and the piece-wise reference in the interval $[t^i; t^{i+1}]$.

Using recursively the dynamics described in Eq. (29), for $i > 0$, we have:

$$\begin{aligned} \mathbf{c}_L^i = \mathbf{c}_L^{0d}\mathrm{e}^{-\lambda_c(i\Delta t-\delta_d)} &+ \sum_{k=2}^{i}(1-\mathrm{e}^{-\lambda_c(k-1)\Delta t})\mathrm{e}^{-\lambda_c\delta_d}\mathbf{c}_{L,u}^{i-k} \\ &+ (1-\mathrm{e}^{-\lambda_c(\Delta t-\delta_d)})\mathbf{c}_{L,u}^{i-1}. \end{aligned} \tag{33}$$

Having $\mathbf{p}_z^i$, the modeled ZMP at time $t^i = t^0 + i\Delta t$, we can compute the future vertical force $(f_{L,z}^i, f_{R,z}^i)$ using the presented method in Fig. 7. We then solve the following optimization problem formulated as a QP to compute $(\mathbf{c}_{L,u}^i, \mathbf{c}_{R,u}^i)$ for $i \in [0; (t_{\mathrm{ds}}^0 - t^0)/\Delta t]$

$$\begin{aligned} \underset{\mathbf{c}_{L,u}^i, \mathbf{c}_{Ru}^i}{\arg\min} \quad & w_z \| \textstyle\sum_i \frac{\mathbf{c}_L^i f_{L,z}^i + \mathbf{c}_R^i f_{R,z}^i}{f_{L,z}^i + f_{R,z}^i} - \mathbf{p}_z^i \|_2 + \\ & w_d \| \textstyle\sum_i R_L(\mathbf{c}_L^i - \mathbf{p}_L) - R_R(\mathbf{c}_R^i - \mathbf{p}_R) \|_2 \\ \text{s.t.} \quad & N_{Lc}\mathbf{c}_L^i \leq O_{Lc} \\ & N_{Rc}\mathbf{c}_R^i \leq O_{Rc} \end{aligned} \tag{34}$$

with $w_d \ll w_z$; R_R and R_L are the rotation matrices that rotate from the world frame to the contact frame (Left or Right). The first term of the cost function is the ZMP tracking error minimization (from (31)), and the second term

$\|\sum_i R_L(\mathbf{c}_L^i - \mathbf{p}_L) - R_R(\mathbf{c}_R^i - \mathbf{p}_L)\|_2$ is added to the cost function to minimize the moment at the ankles.

B. Robot mass estimation

The previous description of the vertical forces distribution stands if the mass m of the robot is well known. To have a measured estimation of the mass, we rely on the force sensors located at each feet. Using then a low-pass filter with a very high cut-off period (≥ 10 s), we can have a more accurate estimation of the robot mass and additionally an estimation of the vertical acceleration $\ddot{z}_c$ using the non-filtered values of the force sensors. This can be useful in case of modeling errors and when the robot is performing tasks such as carrying objects.

C. External disturbances

During the transition between the double support phase to the single support phase, the force applied on the swing foot is no longer controlled and then not considered in the pendulum state. However, it is still possible that the swing foot applies some forces before it leaves the floor and then disturb the pendulum dynamics. For example, this can be seen in Fig. 13. One solution to overcome this perturbation is to consider it explicitly in the dynamics of the pendulum.

Writing the centroidal dynamics and adding the external disturbance wrench from the swing foot, $(\mathbf{f}_{\text{swg}}, \mathbf{n}_{\text{swg}})$ expressed at the swing foot, the pendulum dynamics becomes as follows [47]:

$$\ddot{x}_c = \omega^2 \left(x_c - \kappa x_z + \Delta x_c'\right) \tag{35}$$

$$\omega^2 = \frac{g + \ddot{z}_c}{c_z - z_z} \tag{36}$$

$$\kappa = 1 - \frac{f_{\text{swg,z}}}{m(g + \ddot{z}_c)} \tag{37}$$

$$\Delta x_c' = \frac{1}{m(g+\ddot{z}_c)}[(p_{\text{swg},z} - z_z) f_{\text{swg},x} - p_{\text{swg},x} f_{\text{swg},z} + n_{\text{swg},y} - \dot{L}_{c,y}] \tag{38}$$

where $\dot{\mathbf{L}}_c = (\dot{L}_{c,x}, \dot{L}_{c,y})^T$ is the derivative of the angular momentum at the center of mass. The dynamics of the CoM in the y-axis are similar with an expression of $\Delta y_c'$ such that:

$$\Delta x_c' = \frac{1}{m(g+\ddot{z}_c)}[(p_{\text{swg},z} - z_z) f_{\text{swg},y} - p_{\text{swg},y} f_{\text{swg},z} - n_{\text{swg},x} + \dot{L}_{c,x}] \tag{39}$$

The DCM Eq. (3) can therefore be rewritten as follows,

$$\dot{x}_u = \omega(x_u - (\kappa x_z - \Delta x_c')). \tag{40}$$

An external disturbance is then seen in the pendulum dynamics as an offset on the ZMP with a proportional coefficient on its trajectory. We then control the dynamics of the pendulum under the new ZMP state $\kappa x_z - \Delta x_c'$. If we know the duration of this perturbation δ_p, we can rewrite the stability condition as follows:

$$\begin{aligned} x_u^0 = {} & \omega \int_{t^0}^{\infty} (\kappa x_z(\tau) - \Delta x_c') \mathrm{e}^{-\omega'(\tau - t^0)} d\tau \\ & - \omega \int_{t^0+\delta_p}^{\infty} (\kappa x_z(\tau) - \Delta x_c') \mathrm{e}^{-\omega'(\tau - (t^0+\delta_p) + \delta_p)} d\tau \\ & + \omega \int_{t^0+\delta_p}^{\infty} x_z(\tau) \mathrm{e}^{-\omega(\tau - (t^0+\delta_p) + \delta_p)} d\tau \end{aligned} \tag{41}$$

Using the definition in Eq. (4) we have

$$\begin{aligned} x_u^0 = {} & x_u^\star(t^0, \kappa x_z - \Delta x_c') \\ & - \mathrm{e}^{-\omega' \delta_p} x_u^\star(t^0 + \delta_p, \kappa x_z - \Delta x_c') \\ & + \mathrm{e}^{-\omega \delta_p} x_u^\star(t^0 + \delta_p, x_z). \end{aligned} \tag{42}$$

It is possible to adapt Eq. 23 if we suppose the modeled ZMP delay δ_d to be inferior to δ_p; then we can update Eq. 24 accordingly and use the disturbance inside the IS-MPC.

The MPC stability constraint during single support is therefore updated to be aware of measured external perturbations. The duration of the perturbation δ_p is set empirically with experimental results and has a value of 0.1 s.

It is important to note that this method can be used for any other external perturbation if the perturbation force is known or planned, similarly to what is done in [47].

VI. Steps Timing and position pre-optimization

In this section, we describe how it is possible to update the step planning if this latter is not feasible; now N_s refers to the number of steps inside the step plan.

Knowing the current constraint (ZMP constraints, kinematics constraints and stability condition in Eq. (4)), one could compute the region in which the current DCM lies to have a solution to the IS-MPC problem. Drawing on this idea we aim at rewriting the stability condition by combining the equality condition of Eq. (4) with the ZMP constraint in Eq. (16). To do so, we assume that the normals N defined in Eq. (19) are constant over time and we note it $\bar{N}$. This gives the following stability criterion:

Theorem 1. *If the ZMP constraint is defined as $\bar{N}\mathbf{p}_z(t) < O(t)$ for $t \in [t^0; \infty]$, then the stability condition can be satisfied if and only if:*

$$\bar{N}\mathbf{p}_u^0 \leq \omega \int_{t^0}^{\infty} O(\tau) e^{-\omega(\tau - t^0)} \,\mathrm{d}\tau \tag{43}$$

The proof of this theorem is given in the Appendix.

This inequality constraint requires the DCM to remain inside a polygon to ensure that there is a ZMP trajectory that will keep the DCM trajectory bounded. We call it the *feasibility region*. The equation of this region captures, as a whole, the planned motion of the pendulum, as $O(t)$ represents the location and the size the ZMP constraint, being itself dependent on the step location. Finally, as the ZMP constraint locations are defined with a specific timing, the stability constraint also captures the steps duration.

Figure 8a shows a comparison between two robots executing the same walking plan using the IS-MPC gait generation. In the right case, the robot is being pushed forward. We see that once the DCM is outside the feasibility region for the green plan, new steps must be acheived to keep the balance. Moreover, we show as an example in Fig. 8b how the steps duration influence the required step length to recover from a disturbance. If (for a specified step duration) the required step position to stay balanced is out of a kinematics constraint (i.e, a step too long), then lowering the step duration allows to reduce the length of the required step. Hence, the feasibility region becomes wider.

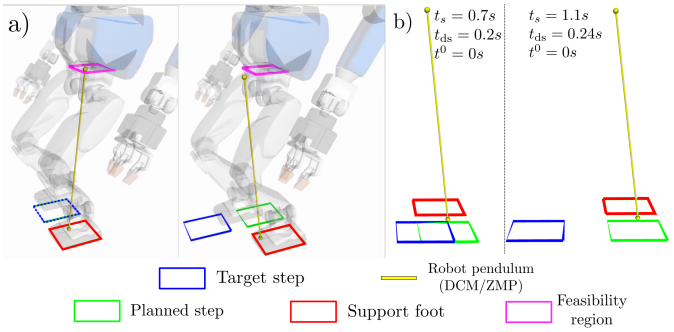


Fig. 8: Feasibility region and how it influence the step plan.

We can discretize Eq. (43) using $O(t)$ as O_i, which is the offset of the ZMP constraint in the interval $[t^i;t^{i+1}]$. The stability condition becomes:

$$\bar{N}\mathbf{p}_u^0 \leq \sum_{i=0}^{\infty} O_i \left[e^{-\omega(t^i - t^0)} - e^{-\omega(t^{i+1} - t^0)} \right] \tag{44}$$

We reduce the ZMP constraint into a rectangle of dimension (dx, dy) located under the support foot during support phase and sliding from one foot to the other during the double support phase in a straight line. Therefore, the path of this rectangle over the plan connects the support position in a piece-wise linear fashion. We perform a space-discretization of this path into $N_{\text{ds}}+1$ positions for each step (for example, Fig. 9 shows in green the case where $N_{\text{ds}} = 3$). Thus, we rewrite the stability condition, for the case of a plan of N_s steps, in the following way:

$$\begin{aligned} \bar{N}\mathbf{p}_u^0 \leq e^{\omega t^0} \Bigg(\tilde{O} + \sum_{i=0}^{N_s-1} \Bigg[O_{i,N_{\text{ds}}}(\mu_{i,N_{\text{ds}}} - \mu_{i+1,0}) \\ + \sum_{j=0}^{N_{\text{ds}}-1} O_{i,j}(\mu_{i,j} - \mu_{i,j+1}) \Bigg] \Bigg) \\ O_{i,j} = \frac{1}{2} \begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + \bar{N} \left[\alpha_j \mathbf{p}_f^i + (1-\alpha_j)\mathbf{p}_f^{i-1} \right] \end{aligned} \tag{45}$$

where $\mu_{i,j} = e^{-\omega t^{i,j}}$ and $\alpha_j = j/N_{\text{ds}}$ with N_{ds} a fixed parameters. $t^{i,j}$ represents either the start of the double/single support phase or intermediate phases. Figure 6 illustrates how are organized the indexes w.r.t to the walking phases.

We make one exception on the definition of $O_{i,j}$ when $i = 0$. As the location of the footsteps $\mathbf{p}_f^{-1}$ and $\mathbf{p}_f^0$ are not decision variables; we set $O_{i,j}$ as O_0 in order to have a ZMP constraint region that approximates as much as possible the current support polygon. Figure 9 shows an example of these definitions and Sec. VI-A gives more detail.

Moreover, there is still the necessity to incorporate a tailing $\tilde{O}$, similarly to Eq. (23), in order to account for the planning outside the decision variables. However, only a truncated or periodic tailing can be used since we are using the whole footstep plan and we don't possess more precise information on the future plan. As our plan is defined long enough in the horizon (more than three steps), we simply set a truncated

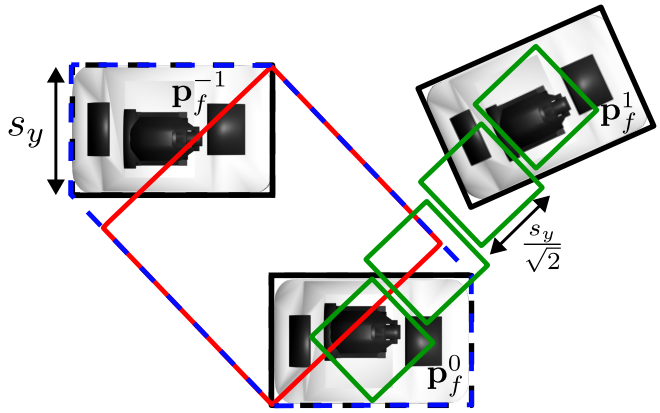


Fig. 9: Computation of the ZMP constraint rectangle shape. The red rectangle once computed define $\bar{N}$ and the green squares are the future ZMP constraints oriented accordingly.

tail having the ZMP constraint to reach the middle of the two contacts, therefore:

$$\begin{aligned} \tilde{O} = \sum_{j=0}^{N_{\text{ds}}-1} O_{N_s,j}(\mu_{N_s,j} - \mu_{N_s,j+1}) + O_{N_s,N_{\text{ds}}}\mu_{N_s,N_{\text{ds}}} \\ O_{N_s,j} = \frac{1}{2} \begin{pmatrix} dx \\ dy \\ dx \\ dy \end{pmatrix} + \bar{N} \left[\frac{\alpha_j}{2} \mathbf{p}_f^{N_s} + (1 - \frac{\alpha_j}{2}) \mathbf{p}_f^{N_s-1} \right] \end{aligned} \tag{46}$$

From Eq. (44), we get a constraint that defines a relationship between the position of the footsteps and their timing in the horizon in order to guarantee stability condition. Therefore, we can formulate an optimization problem that has as decision variables:

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

The actual steps timings can be retrieved as follows:

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

which gives the timing plan illustrated in Fig. 6.

We add bounds to this optimization problem on: (i) the steps location, as defined in Sec. IV-A2; and (ii) the steps phase duration such that:

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

with Δt_{ds}^m, Δt_{ss}^m, Δt_{s}^m the minimum: double support duration, single support duration, and whole step duration. The superscript M represent the upper bound of those constraints. Noting $\mu_{\text{ds}}^m = e^{-\omega \Delta t_{\text{ds}}}$ and similarly for the other duration bounds. The timings constraints can be rewritten as:

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

The constraint defined by Eq. (45) is unfortunately nonconvex, making the problem complex to solve, and thus cannot run at the same frequency as our MPC. Nevertheless, we note that the constraint becomes linear if we take independently the positions or the timings as decision variables. The solution we resort to is to solve alternately the problem either with fixed steps timing or with fixed step locations. This makes it possible to formulate the entire problem as QP optimization.

Finally, to ensure the feasibility of the problem we add slack variables s such that if the problem is not feasible, we can still retrieve the parameters bounded to the activated constraints. The details of this optimization is described hereafter.

A. ZMP constraint region choice

We set the ZMP constraint as a rectangle of dimension (dx, dy), however, as $\bar{N}$ must be constant in the whole horizon, we must choose an orientation of the rectangle beforehand, the size of the ZMP constraint can however change. In double support phase, we set the orientation of the constraint region to approximate as much as possible the current support polygon region with a rectangle. Figure 9 showcase the decision of $\bar{N}$ where the red rectangle covers the largest area in the support polygon [48]. If the robot is in single support phase, the size and orientation of the rectangle can be set to fit those of the support foot.

In the horizon, the size of the rectangle must be set so that it always fits inside the future support polygon. If the orientation of the rectangle differs from the planned step one, a conservative way is then to set the size of the rectangle to a square of size $s_y/\sqrt{2}$, where s_y is the width of the robot foot.

This method reduces the area allowed for ZMP; but this occurs only for supports that happen later in the footstep plan. Since the contribution on the stability is exponentially decaying in the horizon, this solution remains suitable. This is especially true since this optimization is done at the frequency of the MPC in the control loop.

B. Steps location problem

The problem is defined such as we minimize the error between the footstep decision variables $P_f = (\mathbf{p}_f^1, \dots, \mathbf{p}_f^{N_s})^T$ and the planned ones $\hat{P}_f = (\hat{\mathbf{p}}_f^1, \dots, \hat{\mathbf{p}}_f^{N_s})^T$. We therefore solve the following problem having as decision variables P_f and the slack variables s.

$$\begin{aligned}
&\underset{P_f,s}{\arg\min} \|\hat{P}_f - P_f\|_2 + w_s\|s\|_2 \\
&\text{s.t.} \quad \bar{N}\mathbf{p}_u^0 + s \leq \mathrm{e}^{\omega t^0}\left(\tilde{O} + \sum_{i=0}^{N_s-1}\left[O_{i,N_{\mathrm{ds}}}(\mu_{i,N_{\mathrm{ds}}} - \mu_{i+1,0}) \right.\right. \\
&\qquad\qquad \left.\left. + \sum_{j=0}^{N_{\mathrm{ds}}-1} O_{i,j}(\mu_{i,j} - \mu_{i,j+1})\right]\right) \\
&\qquad \text{Kinematic constraint in Eq. (20)}
\end{aligned} \tag{50}$$

With $w_s \gg 1$. The output of this problem provides the references steps $P_{f,\mathrm{ref}}$ for the IS-MPC on Sec. IV

C. Steps timings problem

Here, we aim to remain as close as possible to the desired planned steps timings $\hat{T}_s = \left[(\hat{t}_{\mathrm{ds}}^0, \hat{t}_{\mathrm{s}}^0), \dots, (\hat{t}_{\mathrm{ds}}^{N_s-1}, \hat{t}_{\mathrm{s}}^{N_s-1})\right]$. The objectives $\hat{\mu}_{i_j}$ are then computed such that:

$$\hat{\mu}_{i,j} = e^{-\omega\left[\hat{t}_s^{i-1} + \frac{j}{N_{\mathrm{ds}}}(\hat{t}_{\mathrm{ds}}^i - \hat{t}_s^{i-1})\right]} \tag{51}$$

However if the solution from the steps location optimization produces steps that differ from the reference, we want to plan shorter steps duration than planned in order to come back quickly to the actual plan. To achieve this, we add to the cost function a weight function based on the initial step plan $\hat{\mathbf{p}}_f^i$ and the corrected one $\mathbf{p}_{f,\mathrm{ref}}^i$ $w(\hat{\mathbf{p}}_f^i, \mathbf{p}_{f,\mathrm{ref}}^i)$ such as:

$$w(\hat{\mathbf{p}}_f^i, \mathbf{p}_{f,\mathrm{ref}}^i) = K_s\|\hat{\mathbf{p}}_f^i - \mathbf{p}_{f,\mathrm{ref}}^i\|_2 \tag{52}$$

This weight is used in a step-timing minimization term.

The resulting QP problem is as follows:

$$\begin{aligned}
&\underset{\mu_{i,j},s}{\arg\min} \| \sum_{i=0}^{N_s-1}\sum_{j=0}^{N_{ds}}(\hat{\mu}_{i,j} - \mu_{i,j})\|_2 + w_s\|s\|_2 \\
&\qquad + \sum_{i=1}^{N_s-1} w(\hat{\mathbf{p}}_f^i, \mathbf{p}_{f,\mathrm{ref}}^i)\|\mu_{i,0} - \mu_{i-1,0}\|_2 \\
&\qquad \text{s.t. } \bar{N}\mathbf{p}_u^0 + s \leq \mathrm{e}^{\omega t^0}\left(\tilde{O} + \sum_{i=0}^{N_s-1}\left[O_{i,N_{\mathrm{ds}}}(\mu_{i,N_{\mathrm{ds}}} - \mu_{i+1,0}) \right.\right. \\
&\qquad\qquad \left.\left. + \sum_{j=0}^{N_{\mathrm{ds}}-1} O_{i,j}(\mu_{i,j} - \mu_{i,j+1})\right]\right) \\
&\qquad\qquad \text{Timing constraint in Eq. (49)}
\end{aligned} \tag{53}$$

With $w_s \gg 1$. The output timings of this problem will be considered as the reference steps duration $T_{s,\mathrm{ref}}$ for the IS-MPC in Sec. IV using Eq. (47).

D. Sequential linear problem solving

As our problem is formulated as two different QP problems, we alternately solve the steps location problem and then the steps duration one. The problems' constraints are created using the previously computed μ or O. The reference values $\hat{\mu}$ and $\hat{P}_f$ are always the ones coming from an external footstep plan.

This pattern of alternating QPs can be repeated several times, so we decided to perform it twice to obtain in a short computation time a compromise between step duration and step location optimization. It is important to note that after this sequence is done, the steps location are once more optimized by the IS-MPC.

E. Stepping recovery while standing

The defined region in Eq. (45) represents a geometric condition on the DCM which defines the region where the DCM can be without having to make a step. We can then use it to trigger a walking plan if necessary and increase the stability of the robot even if it is not stepping (i.e., standing in place).

Moreover, because we are using a rectangular shape constraint on the ZMP. This stability region on the DCM is also a rectangle. Therefore, once the DCM under a disturbance is out of the region, knowing from which vertices of the rectangle the DCM violates the stability condition allows to plan which support foot to use to recover. This is done depending on which side of the rectangle the DCM is outside of the feasibility region and how are the feet positioned.

This strategy appears to be effective if the robot is pushed forward or backward as we show in Sec. VII. In case of lateral pushing, we choose the support foot to be the furthest one from the current DCM pose. However, for a better balance, especially in case of lateral pushing, it remains necessary to develop more tools to recover such as having non-convex kinematic constraints (this would allow the left foot to be placed on the right side of the right foot), anticipating the disturbance (to increase the support polygon surface) or using the angular momentum.

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. 10 and on different environments:

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

Disturbance from human pushes are introduced either during walking or in standing phases.

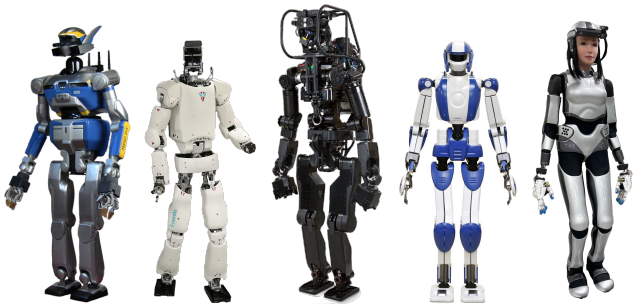


Fig. 10: All the humanoids used in the experiments. From left to right, the HRP-2KAI: a modified version of the HRP-2, used in the DRC; RHPS1: the new humanoid robot from Kawasaki Heavy Industries co-designed with AIST and CNRS; HRP-5P: the last prototype of the HRP family designed to be torque controlled and to be used in building and harsh applications [49]; the HRP-4: the sample present in France and used for the Airbus joint project [50]; and finally, the HRP-4CR used in the ANA Avatar Xprize contest [51].

All the humanoids are controlled in high-gains position, i.e., kinematics control. The whole-body control for all our humanoids is made using the `mc_rtc` task-space optimization control framework[4]. The reason behind this effort is our aim to offer the community 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 appear clearly specified in almost all existing papers in the field. Most of existing algorithms require parameters to tune for each robot and often this knowledge is not highlighted in academic papers. The so-called 'magic numbers' is what prohibits robustness in porting the same algorithm to other humanoid platforms. This is the reason why we decided to implement our approach in all the platforms we possess (we hope that the openness of the code would encourage the readers to do so on their own humanoid and provide any criticisms and feedback on how it can be further improved).

For each robots, 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 gains K_a for the CoP tasks and the foot force difference control (FFDC) gain K_z.

These parameters are set by fitting the modeled CoP, vertical forces or ZMP to the measured ones. It is important to notice the correlation between the parameters. As λ represents the model's 1^{st} order response, the higher the admittance gains are, the faster the wrench is applied and therefore the higher λ is. This is similar for the control of the contacts vertical forces. As an admittance task is more compliant with a lower gain, we set the λ parameters by trying to set the admittance gains as low as possible.

Alas, for the time being the tuning process for these parameters remains empirical; yet, for the sake of transparency in the choice of the control parameters, we provide *ad-hoc* guideline for the tuning procedure (or at least the key elements) we did to conduct our experiments.

1) Set $K_a = 0.01$, $K_z = 10^{-4}$, $\lambda = \lambda_f = \lambda_c = 40$.
2) Make the robot step in place.
3) δ_d can be identified from the transition into walking phase, see Fig. 11.
4) Adjust λ such that the model-ZMP fits well the measured one in single supports.
5) The feet in single support should not vibrate, if it is the case with an appropriate value of λ, K_a should be increased (and so is λ).
6) Adjust λ_z such that the model vertical force fits the measured one in double support.
7) Vibration in double support are often due to vertical forces control, K_z and λ_f should be set to have a smooth and complete transition of the ZMP on the future support foot.
8) λ_c should be at λ and adjusted to improve the ZMP trajectory w.r.t the model once in double support.
9) Finer tuning can be done by making the robot steps forward and backward.

The chosen parameters for each humanoid we used are listed in Table I whereas the common parameters to all robots are displayed on Table II. We also list the sampling period δ at

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

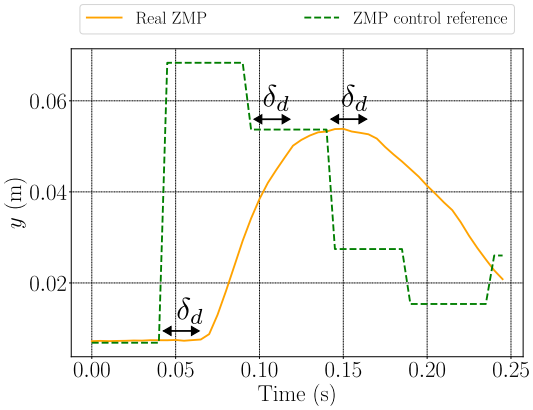


Fig. 11: Delay between the real ZMP and the sent reference on HRP-2KAI.

TABLE I: Chosen parameters for each robot

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

which the logged data and the computed joints angles are sent. The IS-MPC weights that handles 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 done as those weights reduces the compliance of the robot during disturbances and make it less likely to trigger steps.

In addition to those parameters, the whole-body control computes the reference joints angles by solving a quadratic optimization problem where the 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. We use the following tasks as QP-control objectives:

- Control the position, velocity and acceleration of the CoM;
- Control the contact forces (CoP Task);
- Control the chest to a defined orientation w.r.t the orientations of the feet;
- Control the swing foot position speed and acceleration during single support phases;
- Reach a defined joint configuration to account for redundancy and singularities (Posture Task), with low task-gains.

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

Each of these tasks is formulated to set the acceleration of a body, CoM or joints as in the form of the one defined in Eq. (5). Those parameters (common to all robots) are listed in Table III.

TABLE III: 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	$2\sqrt{K_p}$	$2\sqrt{K_p}$	$2\sqrt{K_p}$

The experiments show the robustness of the proposed method by making the robot walk on long distances, uneven terrain (without any specific knowledge of it) and under external disturbances provided by operator pushes. We finally tested perturbations during a standing phase to trigger stepping recovery.

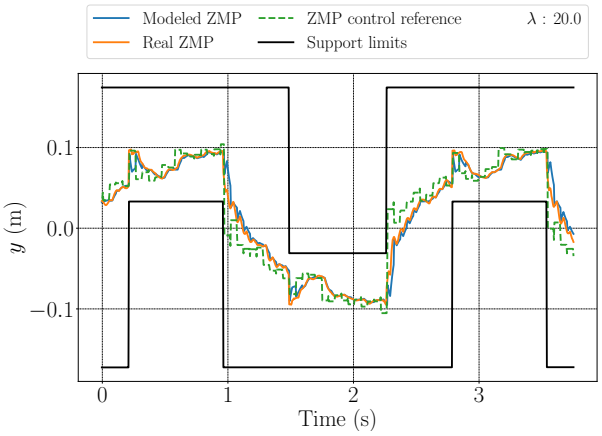
All the plots are displayed in a frame linked to the robot floating base such as: $\mathbf{z}$ is the vertical axis, the plane $(\mathbf{z},\mathbf{y})$ is the coronal plane and the plane $(\mathbf{z},\mathbf{x})$ is the sagittal plane

At first, we tried to achieve the same walking conditions for each humanoid robot. Nevertheless, 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 HRP-4 is located in France whereas the remaining are in Japan, where experiments in the campus are more strictly regulated. For example, experiments on HRP-4CR were only indoor and on rigid flat floor due to mechanical weaknesses, all data for this robot are reported in [51] as it served the ANA Avatar XPrize purpose exclusively. It was possible to run experiments on the HRP-4 outdoor without safety ropes, which is not yet technically possible in Japan. Not all data are reported for each robot, we made a choice that could have been different, yet the multimedia material accompanying the paper is thorough enough to complement the data.

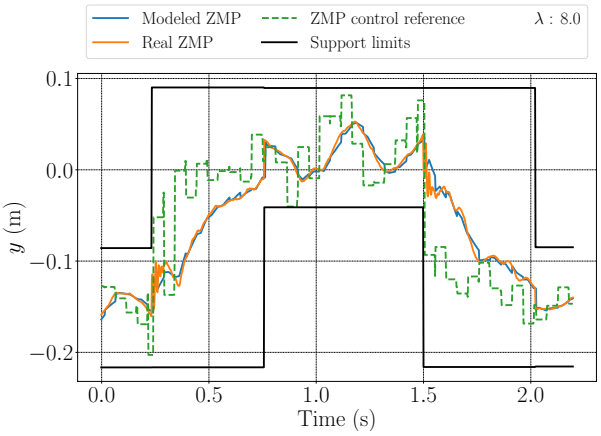
A. *Force control and model evaluation*

Force control model evaluation is assessed on all the listed humanoids. Figure 12 shows that the first order dynamic of the ZMP with a delay is an appropriate estimation of the real ZMP dynamics. To emphasize on the fact that we aim at modeling the ZMP behavior, in order to make the robot HRP-2KAI, HRP-5P and RHPS1 walk on a very compliant material (here a mattress), we adjusted the 1$^{\text{st}}$ order response λ. For instancce, even if the model response timing is very slow, we can keep the control stable and the robots well balanced to execute dynamic walks (see Sec. VII-D).

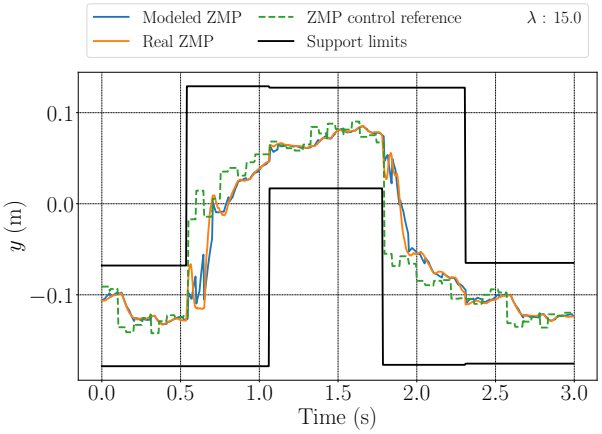
We also show in Fig. 13 how the measured disturbance from the swing foot is used to enhance the pendulum trajectory planning. The figure only shows the measured κ (defined in Sec V-C) as it captures the vertical component of the disturbance which is the core component in our case. Moreover, the value of $\Delta y'_c$ depends on the absolute position of the robot in the world frame.



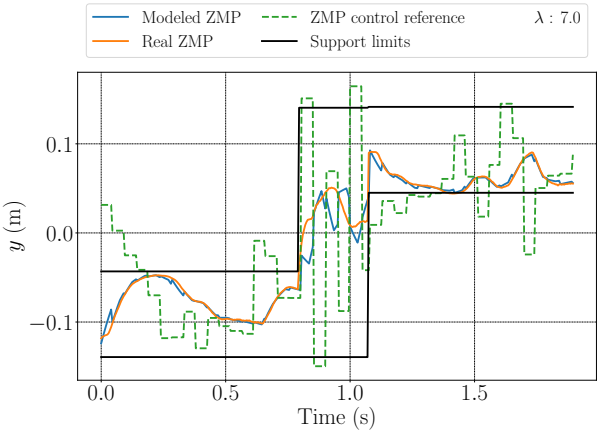
(a) On HRP-2KAI.



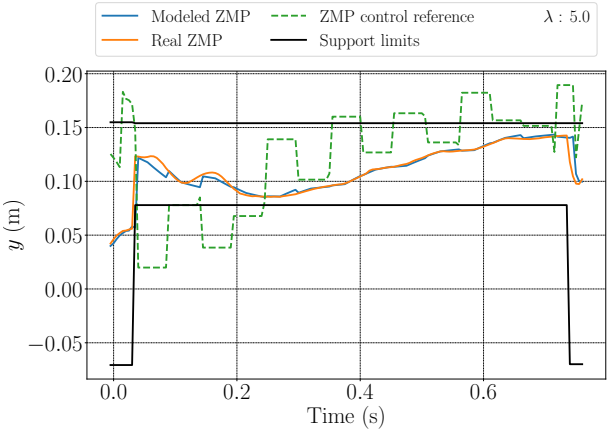
(b) On RHPS-1.



(c) On HRP-5P.



(d) On HRP-4.



(e) On HRP-4CR.

Fig. 12: ZMP model compared with robots' measured ZMP when walking on rigid flat floors; model state is updated at a sampling rate Δt.

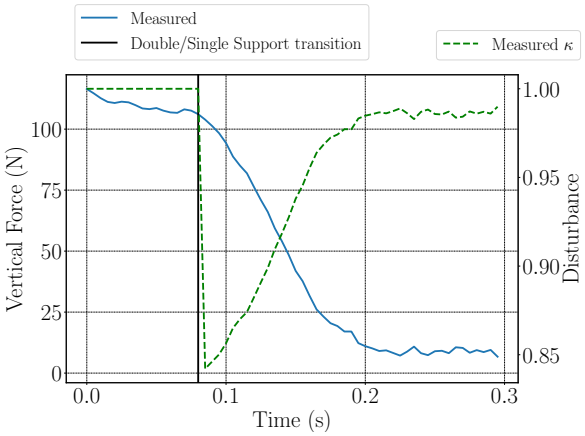


Fig. 13: Measured disturbances during contact release. Around 100 N are still observed once the swing foot force control is deactivated, resulting in a change of the pendulum dynamics.

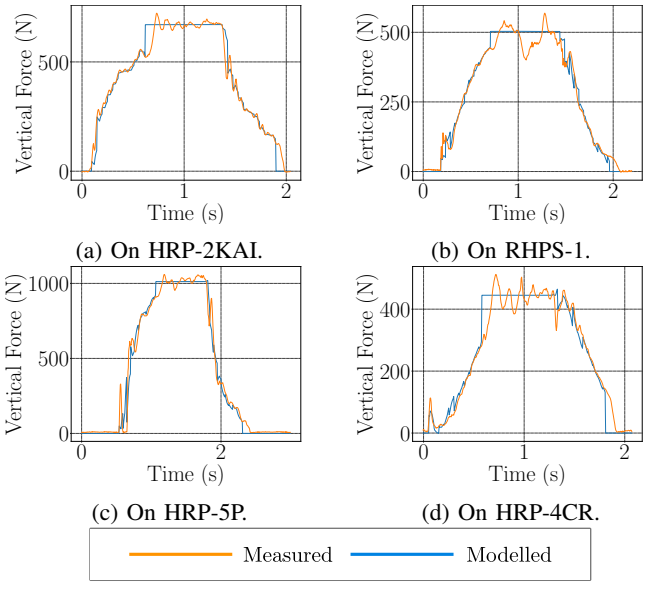


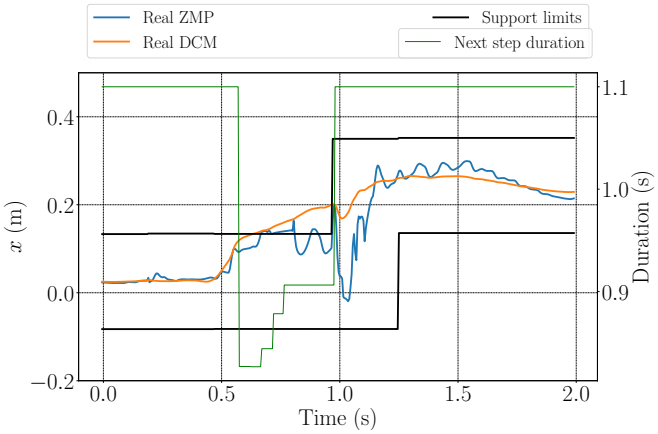
Fig. 14: Vertical force model compared with the real robots measured one (left foot) while walking on a rigid flat ground. Model state is updated at sampling rate Δt.

Finally, we show in Fig. 14 that the vertical forces during double support also matches the expected model, when the model on the Fig. 14 is a constant value, the robot is in single support so the vertical force is no longer controlled.

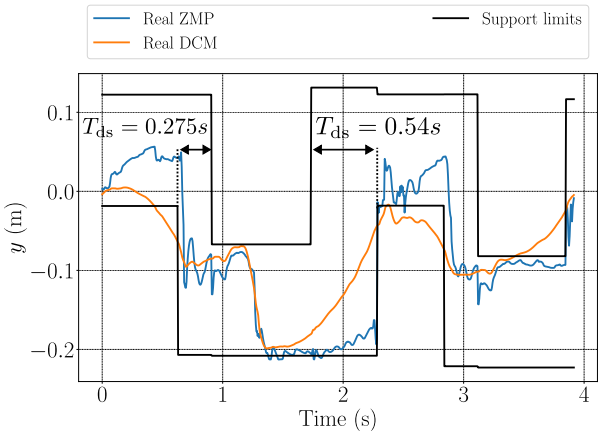
It must be noted that the measure on the HRP-4 force sensor were imprecise due to a cross-talk between measured torque and vertical forces impacting the ZMP measure in double support. This coupling has been properly identified in single support only and is countered by a corrective plug-in on the sensor data.

B. Footsteps and steps timing adaptation

The Footsteps and steps timing adaptation is also assessed on all the humanoid robots. We show in Fig. 15a, which



(a) Forward (similarly observed for backward) push.



(b) Lateral (similarly observed for right or left) push.

Fig. 15: Depending on the disturbance direction, the step duration plan is updated differently (these behaviors extend to all grounds).

illustrates a recovery after a front push, how the planning is updated in case of disturbances. 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 as it can be seen in Fig. 15b where a lateral disturbance induces a slower double support duration to provide time to the DCM to reach the next support foot.

C. Static push recovery

Among all listed humanoids robots, we created by hand disturbances in a standing phase and used the condition defined in Sec. VI-E to trigger a recovery step. To add safety margins, the size of the allowed ZMP region is set to be smaller than the real size of the robot foot, therefore a recovery step can be triggered even if the robot can remain balanced without stepping. The decision of the support foot is made manually depending on the position of the DCM and the contact configuration at the instant of the pushes. If the disturbance is frontal, we choose the support foot which is

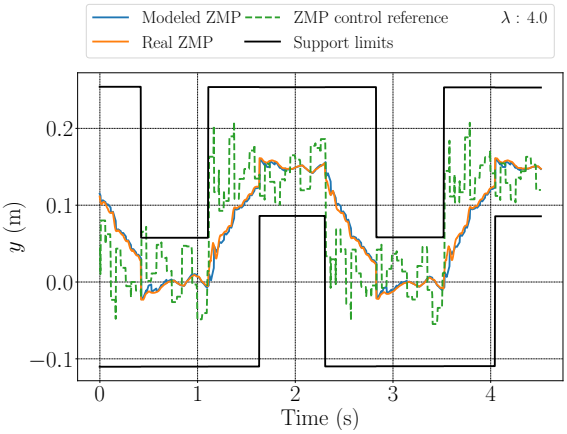
the closest from the DCM. However if the disturbance is lateral, the support foot is the one the furthest from DCM. We defined a perturbation to be lateral or frontal depending on which sides of the stability region has been broken. Recall that once the necessity of stepping has been detected, we do not forward 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. Once a disturbance occurs, a step is triggered, however, if the required steps to recover becomes unfeasible because it they are too long or if their duration is too short, the robot will certainly fall. In those 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). This comes down to remove the hypothesis on the LIP mode to make the robot dynamics closer to the complete centroidal model.

D. Walking on soft ground

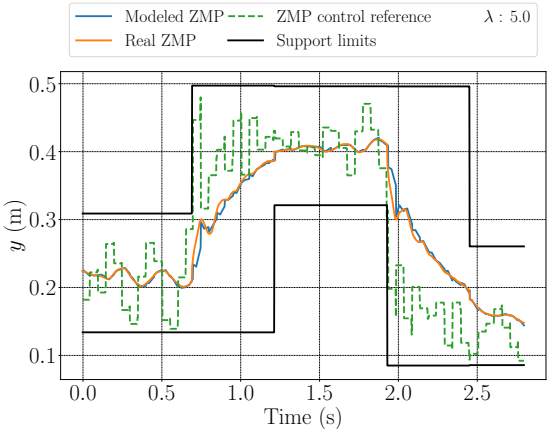
This test has been conducted on HRP-2KAI, HRP-5P and RHPS1. Walking on compliant terrain for bipedal robot is a challenging task that showcases the need for the knowledge of how the force are behaving w.r.t the controls inputs.

As an example, without our control scheme being active, HRP-2KAI could not stand on the mattress by itself, even when the robot was controlled to be rigid 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 [52] 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, in order to make our robots walk on compliant terrain, one must adapt the parameters that models the forces behavior. In our case, it came down to adapt the 1^{st} order parameter λ. The mattress we chose is compliant, yet we made sure its deformation never reaches a full compression 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 16. As mentioned in Sec. VII-A, walking on compliant floor mostly required to adapt the ZMP model parameter λ. However we also needed to adapt 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. Among the available robots we experimented over a compliant ground, all of them were position-controlled. However the HRP-2KAI is equipped with an absorbing mechanism made of a rubber bush between the foot and the ankle actuation. This difference in mechanical design seems to have an effect on 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. 17 where we see that in similar condition (environnement, stepping frequency, step length, ZMP trajectory profile), RHPS1 feet oscillate whereas HRP-2KAI ones is moving relatively smoothly.



(a) On HRP-2KAI.

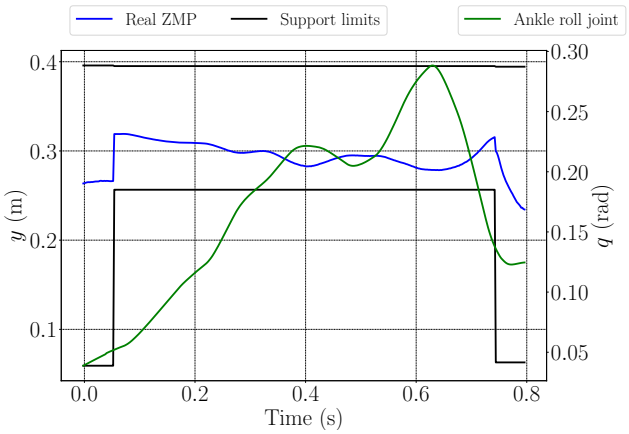


(b) On RHPS1.

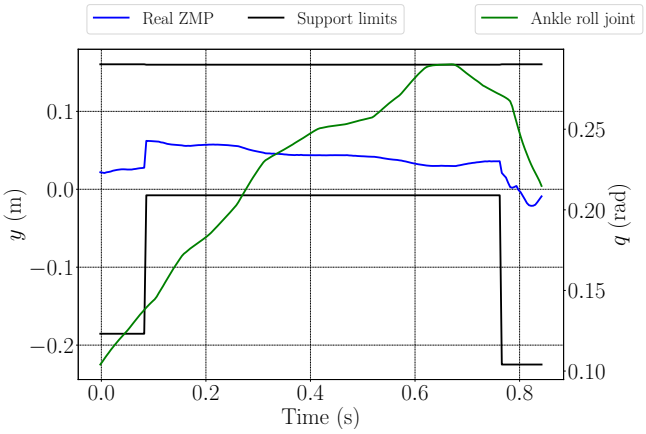
Fig. 16: ZMP model compared with robots' measured ZMP when walking on soft grounds, model state is updated at sampling rate Δt.

E. Walking outdoors

Outdoor experiments are made with HRP-4 and HRP-2KAI as it can be seen on Fig. 1. Walking on outdoors terrains showcases the necessity of compliance for the force control. This is because even for outdoor terrains that are nearly flat, they have non negligible local roughness. Additionally, some terrains on which the robots walked have a pseudo-compliant behavior due to the grass or the dirt. This allows us to display the capability to adapt the force behavior model to various conditions. The ZMP model evaluation on such terrain is displayed in Fig 18. The multimedia material attached to this paper shows the compliance of the contact feet on the terrain. Moreover, this kind of terrain are uneven with various local slopes. When this slope has a positive angle, the walking control is not much affected as the early contact can be detected with the forces measurement. However, in the case of a downward slope, the lack of knowledge of the contact location can lead to enable the force control of the landing foot in the air inducing potentially a disturbance. In our used force control scheme, FFDC can accommodate for this kind of false positive contact, as it will keep moving the feet down

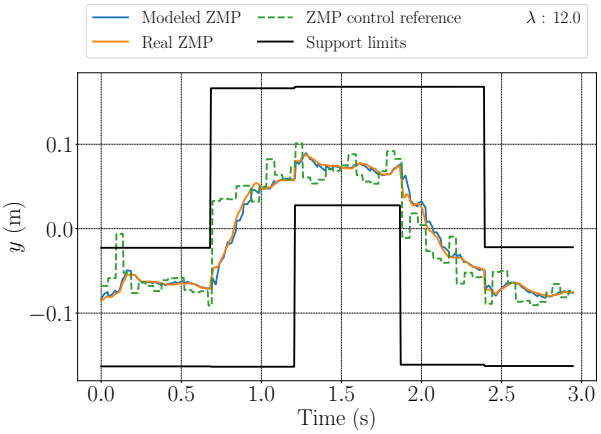


(a) On RHPS1.

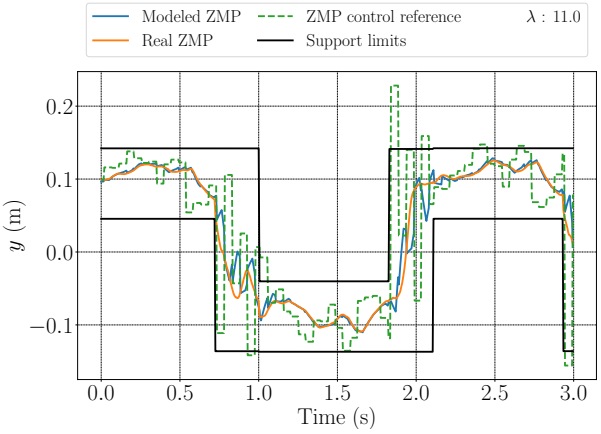


(b) On HRP-2KAI.

Fig. 17: On similar compliant ground with similar walking, feet spring damper reduces oscillation in single support.



(a) On HRP-2KAI.



(b) On HRP-4.

Fig. 18: ZMP model compared with the real robots ZMP walking outdoors, model state is updated at a sampling rate Δt.

until force is measured. In that objective, the FFDC K_z gain is increased on the HRP-4 for this experiment from 0.0001 to 0.00015. This works if the real contact remains close from the feet. In practice, poor contact detection is the main reason of few outdoor experiments failure.

It should be noted that in both case, the wind was very strong and added non-negligible perturbations. This is noticeable from the multimedia for the HRP-2KAI. For the HRP-4, between the starting and ending experiments, the grass was cut and sometime strong perturbations occur when steeping on apparent thick roots.

VIII. Conclusion

In this paper, we presented a new walking control scheme for bipedal robots that can adjust steps location and duration. This novel aspect of this control scheme is in the use of a LIPM's model MPC in 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 control the feet contact forces accordingly to the expected 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 nature* of the ground the robot is walking on. 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 seems to be possible as some of the parameters represent a model we try to fit and we can observe. Moreover, the 1$^{\text{st}}$ order model for ZMP dynamics can be retrieved by modeling the floor with a spring damping dynamics under the force control scheme we detailed in Eq. (6), however the general dynamics reflects rather a 2$^{\text{nd}}$ order behavior which require an appropriate estimation of the ZMP velocity. We have also tried to make the HRP-4 robot to walk on river pebbles outdoors, but the controller failed in all trials. One of the cause is the flatness and rigidity of the soles; the other is clearly the lack of proper handling of the momentum. As future work, we are currently investigating enhancing our controller with full

centroidal dynamics and redesigning the soles, as extensions of [53] with compliant material to better cast the grounds at contacts. All these aspects are somewhat interconnected, understanding such dependencies is paramount to reach the *plug-and-walk* initiative we are aiming for. As stated in Sec. II, our team is also working toward a pure machine-learning approach for walking, we are convinced that robust walking would emerge from a neat hybridization that takes full benefit of both approaches.

APPENDIX
PROOF OF THE THEOREM

We suppose here that $\bar{N}$ and $O(t)$ is respectively a $N_v \times 2$ matrix and a $N_v \times 1$ row vector such that the ZMP constraints $\forall t \in [t^0;\infty]$ is a N_v vertices convex polygon (with possibly multiple vertices at the same position). Without loss of generality we can assume that the normals $\bar{N}$ are all distinct and that $\bar{N}$ and $O(t)$ are 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, we note $\bar{n}_{x,i}$ and $\bar{n}_{y,i}$ respectively the 1st and 2nd element of the i^{th} row of $\bar{N}$ and $o_i(t)$ the value of the i^{th} row of $O(t)$. Therefore, the ZMP constraint can be written for each row i and gives:

$$\bar{n}_{x,i}x_z(t) + \bar{n}_{y,i}y_z(t) \leq o_i(t) \tag{54}$$

multiplying each side by $e^{-\omega(t-t^0)}$ and integrating on $[t^0;\infty]$:

$$\bar{n}_{x,i}\int_{t^0}^{\infty} x_z(\tau)e^{-\omega(\tau-t^0)}d\tau + \bar{n}_{y,i}\int_{t^0}^{\infty} y_z(\tau)e^{-\omega(\tau-t^0)}d\tau \leq \int_{t^0}^{\infty} o_i(\tau)e^{-\omega(\tau-t^0)}d\tau \tag{55}$$

using the definition of the stability condition of Eq. (4):

$$\bar{n}_{x,i}x_u^0 + \bar{n}_{y,i}y_u^0 \leq \int_{t^0}^{\infty} o_i(\tau)e^{-\omega(\tau-t^0)}d\tau \tag{56}$$

This result holding for each row i, the last equation can be summarized as in Eq. (43) which proves the necessity.

We note $\mathbf{v}_{z,j}(t)$ the 2D coordinate of the j^{th} vertex of the ZMP constraint polygon $\forall t \in [t^0;\infty]$ and $j \in [1;N_v]$. We note $\bar{N}_j$ and $O_{z,j}(t)$ respectively the 2×2 square matrix and the 2×1 row vector that represent 2 consecutive constraints crossing $\mathbf{v}_{z,j}(t)$. $\bar{N}_j$ is then a matrix made out of two rows of $\bar{N}$ and $O_{z,j}(t)$ is made out of two terms of $O(t)$. We can then write:

$$\mathbf{v}_{z,j}(t) = \bar{N}_j^{-1}O_{z,j}(t) \tag{57}$$

Let O_u be (from Eq. (43)):

$$O_u = \omega\int_{t^0}^{\infty} O(\tau)\mathrm{e}^{-\omega(\tau-t^0)}\,\mathrm{d}\tau \tag{58}$$

and let E be the convex set of initial DCM positions $\mathbf{p}_u^0$ such that Eq. (43) 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 $\gamma \in [0;1]$.

$$\forall t \in [t^0;\infty],\ \bar{N}(\gamma\mathbf{p}_{z,1}(t) + (1-\gamma)\mathbf{p}_{z,2}(t)) \leq \gamma O(t) + (1-\gamma)O(t) \tag{59}$$

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

We extract $O_{u,j}$ from O_u similarly to the definition of $O_{z,j}$ and we compute $\mathbf{v}_{u,j}$ the vertices of the convex polygon representing E.

$$\mathbf{v}_{u,j} = \bar{N}_j^{-1}O_{u,j} \tag{60}$$

$$\mathbf{v}_{u,j} = \omega\int_{t^0}^{\infty} \bar{N}_j^{-1}O_{z,j}(\tau)\mathrm{e}^{-\omega(\tau-t^0)}\,\mathrm{d}\tau \tag{61}$$

$$\mathbf{v}_{u,j} = \omega\int_{t^0}^{\infty} \mathbf{v}_{z,j}(\tau)\mathrm{e}^{-\omega(\tau-t^0)}\,\mathrm{d}\tau \tag{62}$$

To prove the sufficiency, let $\mathbf{p}_u^0 \in E$. E being convex, we can write $\mathbf{p}_u^0$ as a convex combination of the feasibility region vertices such as:

$$\mathbf{p}_u^0 = \sum_{j=1}^{N_v} \gamma_j\mathbf{v}_{u,j} \tag{63}$$

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

We consider the particular following ZMP trajectory $\forall t \in [t^0;\infty]$

$$\mathbf{p}_z(t) = \sum_{j=1}^{N_v} \gamma_j\mathbf{v}_{z,j}(t) \tag{64}$$

This trajectory satisfies the ZMP constraints as it is a convex combination of the extreme trajectories made out 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 Eq. (4) with Eq. (64):

$$\omega\int_{t^0}^{\infty} e^{\omega(\tau-t^0)}\sum_{j=1}^{N_v}\gamma_j\mathbf{v}_{z,j}(\tau)d\tau = \omega\sum_{j=1}^{N_v}\gamma_j\int_{t^0}^{\infty} e^{\omega(\tau-t^0)}\mathbf{v}_{z,j}(\tau)d\tau \tag{65}$$

Using now Eq. (62) and Eq. (63):

$$\sum_{j=1}^{N_v}\gamma_j\omega\int_{t^0}^{\infty} e^{\omega(\tau-t^0)}\mathbf{v}_{z,j}(\tau)d\tau = \sum_{j=1}^{N_v}\gamma_j\mathbf{v}_{u,j} = \mathbf{p}_u^0 \tag{66}$$

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

REFERENCES

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

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[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] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] S. Coros, P. Beaudoin, and M. van de Panne, "Generalized biped walking control," *ACM Trans. Graph.*, vol. 29, no. 4, jul 2010.

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

[24] 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.

[25] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 7309–7315.

[26] 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.

[27] 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.

[28] 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.

[29] R. P. Singh, Z. Xie, P. Gergondet, and F. Kanehiro, "Learning bipedal walking for humanoids with current feedback," arXiv, 2303.03724, cs.RO, 2023.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] L. Lanari and S. Hutchinson, "Inversion-based gait generation for humanoid robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 1592–1598.

[36] S. Caron and A. Kheddar, "Dynamic walking over rough terrains by nonlinear predictive control of the floating-base inverted pendulum," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5017–5024, 2017.

[37] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, 2017.

[38] 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.

[39] T. Koolen, M. Posa, and R. Tedrake, "Balance control using center of mass height variation: Limitations imposed by unilateral contact," in *IEEE-RAS 16th International Conference on Humanoid Robots*, 2016, pp. 8–15.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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 manufac-

turing: the airbus use-cases," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 30–45, October 2019.

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

[52] 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.

[53] 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.