Walking Control Based on Step Timing Adaptation

Majid Khadiv[⋆], Alexander Herzog[∇], S. Ali. A. Moosavian[†], and Ludovic Righetti[⋆,∘]

***Abstract*—[1] Step adjustment can improve the gait robustness of biped robots, however the adaptation of step timing is often neglected as it gives rise to non-convex problems when optimized over several footsteps. In this paper, we argue that it is not necessary to optimize walking over several steps to ensure gait viability and show that it is sufficient to merely select the next step timing and location. Using this insight, we propose a novel walking pattern generator that optimally selects step location and timing at every control cycle. Our approach is computationally simple compared to standard approaches in the literature, yet guarantees that any viable state will remain viable in the future. We propose a swing foot adaptation strategy and integrate the pattern generator with an inverse dynamics controller that does not explicitly control the center of mass nor the foot center of pressure. This is particularly useful for biped robots with limited control authority over their foot center of pressure, such as robots with point feet or passive ankles. Extensive simulations on a humanoid robot with passive ankles demonstrate the capabilities of the approach in various walking situations, including external pushes and foot slippage, and emphasize the importance of step timing adaptation to stabilize walking.**

***Index Terms*—Bipedal locomotion, robust walking, timing adjustment, push recovery, slippage recovery.**

I. Introduction

A. Motivation

Due to the unilateral nature of feet-ground interaction, legged robots can easily fall down. The most important aspect of walking control is therefore preventing falls even in face of strong perturbations. In this regard, a controller can act on three different aspects of the gait: 1) where to step, 2) when to step, and 3) how to manipulate ground reaction forces. State of the art walking pattern generators mostly focus on the third aspect, i.e, the optimal modulation of the center of pressure (CoP), sometimes in conjunction with step placement or timing adaption. While such strategies afford great flexibility in walking pattern generation, they implicitly assume high control authority on the end-effectors' CoP. Furthermore, since adapting step timing often gives rise to non-convex optimization problems, most available walking pattern generators keep footstep timing fixed.

In this paper, we study the problem of optimal footstep location and timing adaptation without explicit control of the feet CoP or the robot center of mass (CoM). This enables the relaxation of control constraints on the CoP associated to more traditional receding horizon algorithms, which is particularly relevant for biped robots with no or limited control authority over their CoP: robots with passive ankles, small or point feet.

B. Realtime walking pattern generators

To date, the most successful approaches for real-time walking control mostly consider linear models of the CoM dynamics and especially the linear inverted pendulum model (LIPM) [27]. Indeed, more complex models often lead to high dimensional, non-convex and computationally complex algorithms, therefore limiting their applicability for real-time planning and control [37], [32], [7], [23]. Recent work for quadruped locomotion also consider more descriptive dynamic models than the LIPM while retaining the capability of running algorithms in real-time. For instance, assuming the legs are massless and the base rotation is small, [10] uses a linear dynamic model between the base states and contact forces for real-time motion planning. [17] proposes an efficient motion planner and controller based on differential dynamic programming (DDP) that uses an approximation of the robot base dynamics. Although these proposed model predictive control (MPC) approaches are real-time capable, they need another reactive planner level to generate contact sequences. There are also formulations based on more descriptive models with the capability to reason about contact sequence and timing, for example using a relaxed contact model [39] or adding implicitly complementary constraints [55]. However, the resulting optimization problems are not convex and can easily get stuck in undesired local minima. In bipedal locomotion, the LIPM is still often preferred because legs constitute a considerable amount of the robot's total mass and there is little need to vary so much the CoM height.

Leveraging analytical solutions of the LIPM, several approaches generate CoM trajectories consistent with a predefined ZMP trajectory [19], [38], [4]. In these approaches, both the position and velocity of the CoM are restricted which constrains both divergent and convergent parts of the LIPM dynamics. In contrast, [49] constrained only the divergent part of the CoM dynamics to generate a trajectory for the divergent component of motion (DCM) based on predefined footprints (ZMP trajectory). Prior to [49], the divergent part of the LIPM dynamics was also used to explain human walking characteristics under the name of extrapolated center of mass (XCoM) [24]. This concept is equivalent to the original capture point (CP) idea [44], i.e. the point on which the robot should step to come to a stop. In these methods, there is no feedback from the current state of the robot to adapt the CoM motion in the presence of disturbances. To circumvent this, [12] proposed

[⋆] Movement Generation and Control Group, Max-Planck Institute for Intelligent Systems, Germany. e-mail: firstname.lastname@tuebingen.mpg.de

[∇] Google X, email: alexanderherzog001@googlemail.com

[†] Department of Mechanical Engineering, K. N. Toosi University of Technology, Tehran, Iran. e-mail: moosavian@kntu.ac.ir

[∘] Department of Electrical and Computer Engineering and Department of Mechanical and Aerospace Engineering, New York University, USA. e-mail: ludovic.righetti@nyu.edu

[1] Part of the material presented in this paper has been presented at the 2016 IEEE/RAS International Conference on Humanoid Robots (Humanoids), [29]

a feedback law tracking a DCM trajectory. Although this controller can quickly react to disturbances, perfect DCM tracking assumes unconstrained CoP manipulation.

All of these walking pattern generators can be seen as variants of the same MPC scheme [54]. One of the pioneering work relating walking pattern generation to optimal control [26] proposed a preview control method to generate CoM trajectories based on predefined ZMP trajectories. Feedback from the current robot state was used to recompute and adapt the motion online. [52] improved the performance of this approach in the presence of relatively severe pushes, by constraining the motion of the ZMP inside the support polygon rather than predefining a desired ZMP trajectory. The resulting algorithm computes both ZMP and CoM trajectories respecting feasibility constraints at each control cycle. In all these approaches, the receding horizon is several foot steps long with fixed step locations and timing.

C. Step adjustment and timing adaptation

Explicit manipulation of the CoP to control the DCM or CoM imposes several restrictions on the leg design. Indeed, robots with point contact feet [25] or robots with passive ankles [30] have very limited control authority, if any, on the foot CoP. The modulation of the CoP is also very limited for robots with actuated ankles, due to the rather small foot support area and the limited amount of available ankle torque. This is in contrast to step adjustment, which allows to select the next step location in a relatively large area compared to the support polygon. It constitutes therefore a more significant tool for stabilizing biped walking. Step adaptation algorithms makes walking pattern generators more robust against disturbances. Successful approaches can either employ heuristics to adapt predefined step sequence based on DCM tracking error [12], [30], consider step location as a decision variable in an MPC framework [11], [20], [21], use constrained optimization to adapt step location [15], [28] or modify a preview control approach [50]. However, in all these methods step timing is never adapted as it would render the problem non-convex when considered over several footsteps.

One of the earliest work on step timing adaptation [38] used an analytical approach to generate CoM/CoP trajectories, and a step period adaptation algorithm to alleviate ZMP fluctuations in case of an immediate change of the step location. [40] proposed a three stage algorithm to adapt ZMP trajectory, step location and timing. First, preview control is used to adapt the ZMP and compensate for CoM tracking error. If the computed ZMP is outside the support polygon, a second stage extends or shortens the step period. Finally the third stage adapts the next step location. [14] showed that using a combination of step location and timing adaptation increases significantly the basin of attraction for bipedal locomotion. [36] proposed an analytical method for computing nominal gait variables for a desired walking velocity, and an algorithm for adapting both step location and timing based on heuristics. [8] proposed an analytical method for step timing and foot placement adaptation based on the CoM state feedback, with a priority given to the sagittal gait. [18] modified the analytical approach of [12] by adjusting both step location and timing using heuristics to compensate for the DCM tracking error. In addition to step timing, single support duration can also be adapted [41], [31] to deal with soon or late landing of the swing foot using contact detection.

Step duration was also used as an optimization variables in [1], [34]. However, these approaches result in non-convex optimization problems which are computationally expensive and do not guarantee convergence to a global minimum. [35] proposed an extension of the gait planning approach in [20] to adjust step duration. They related the problem through a mixed-integer quadratic program (MIQP) which has combinatorial complexity. In [3], a robust approach is proposed to deal with the nonlinearity introduced by adapting step timing in a receding horizon controller. Furthermore, [5], [6] used timing adaptation to limit the acceleration of the swing foot, during walking on uneven terrains.

D. Viability and capturability constraints

Viability theory [2] is an appealing framework to discuss the stability of walking. The viability kernel includes all the states from which it is possible to avoid falling [51]. Computing this kernel is generally not possible but it has been argued [53] that it is sufficient to limit an integral of the states of the system over several previewed steps to guarantee long-term walking stability [54]. In general, viability can be guaranteed by setting a terminal condition on the states when considering several steps [49], [12], or by minimizing any derivative of the CoM over a sufficiently large horizon [26], [52], [20]. As a result, the majority of walking pattern generators today consider several steps of preview to optimize walking.

Capturability is the ability to come to a stop after a certain number of steps and was extensively analyzed for the LIMP and its extension in [33]. Based on an N-step capturability analysis, [9] proposed a heuristic approach that tries to find the gait parameters that bring the robot back to a desired gait cycle by taking the least number of steps. In [33], in particular, a bound for the infinite-step capturability set is computed, which is almost equivalent to computing the viablity set of the LIPM. It is a very valuable result, as this can guarantee viability without the need to consider several walking steps and without over-constraining the system, therefore significantly simplifying the walking pattern generator problem. To the best of our knowledge, this bound has never been used to synthesize a walking pattern generator.

E. Contributions of the paper

In this paper, we use results characterizing the viability kernel of the LIPM [33] to argue that it is sufficient to only consider the next step location and timing to ensure walking stability from any viable state[2]. This means that adding a longer preview of steps, under the LIPM assumption, does not improve the ability to reject disturbances as long as both step timing and location are adapted for the next footstep. To

[2] Throughout the paper, we use the term *walking stability* in the sense of *viability* (i.e. weak invariance) [51]. We also define *robustness* as the ability to recover from external impulsive disturbances.

the best of our knowledge, this insight has not been exploited in control synthesis, as state of the art receding horizon approaches always include sufficient conditions for viability over a preview of several steps despite it being unnecessary and computationally more complex.

Consequently, we design a model predictive controller that optimizes the next step location and, most importantly, timing at every control cycle and guarantees walking stability from any currently viable state. The algorithm consists in solving a quadratic program which contains at least an order of magnitude less decision variables than standard MPC walking algorithms [21]. To the best of our knowledge, it is the first algorithm able to adapt both step timing and location while keeping a convex problem. As our approach does not rely on CoP modulations, it is applicable to any biped robots, including robots with passive ankles and point feet.

We also propose a strategy to adapt online Cartesian swing foot trajectories to follow adapted step timing and location. Extensive simulations on a full humanoid robot with passive ankles using hierarchical inverse dynamics [22] demonstrate the capabilities of the approach for robots with limited control authority over their CoP. In particular, our experiments emphasize the importance of timing adaptation during walking. Interestingly, the full body controller does not require an explicit control of the CoP nor of the CoM horizontal motion. As a result, our approach allows the robot to handle a significant amount of disturbances including external pushes and foot slippage.

This paper extends our preliminary work [29] in three main directions. First, we present a complete viability analysis of the LIPM under footstep constraints and show that we can find the viability kernel by taking into account only the next footstep. This enables the formulation of a MPC walking controller with viability guarantees. Second, we present several push recovery simulations on a full humanoid robot with passive ankle and show that we can stabilize walking in the presence of strong pushes by merely adjusting step location and timing without any explicit control of the CoP or CoM. Finally, we present several slippage recovery simulations demonstrating that our approach is robust to stance foot slippage. To the best of our knowledge, slippage recovery for humanoid walking is seldom studied in the literature.

II. Problem Formulation and Viability

A. Fundamentals

The LIPM constrains the motion of the CoM on a plane by using a telescopic massless link connecting the CoP to the CoM [27]. Its dynamics is written as

$$\ddot{x} = \omega_0^2(x - u_0) \tag{1}$$

where $x \in \mathbb{R}^2$ is the CoM horizontal position (CoM height has a fixed value z_0), and $u_0 \in \mathbb{R}^2$ is the CoP position on the floor. For point contact feet, u_0 is identical to the contact point. ω_0 is the natural frequency of the pendulum ($\omega_0 = \sqrt{g/z_0}$, where g is the gravity constant). Using the DCM, $\xi = x + \dot{x}/\omega_0$, the LIPM dynamics can also be written as [12]

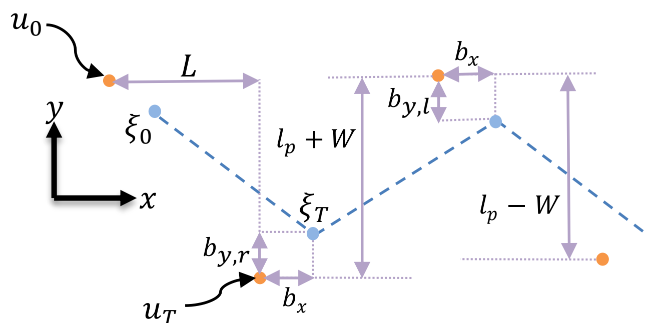

Fig. 1. Schematic view of walking with footprints, DCM, and DCM offset.

$$\dot{x} = \omega_0(\xi - x) \tag{2a}$$

$$\dot{\xi} = \omega_0(\xi - u_0) \tag{2b}$$

Equation (2) explicitly reveals the stable and unstable parts of the LIPM dynamics, where the CoM converges to the DCM (2a) and the DCM is pushed away by the CoP (2b). Solving (2b) as an initial value problem gives

$$\xi(t) = (\xi_0 - u_0)e^{\omega_0 t} + u_0 \tag{3}$$

and the DCM at the end of a step of duration T is

$$\xi_T = (\xi_0 - u_0)e^{\omega_0 T} + u_0 \tag{4}$$

This representation is very convenient, as it is sufficient to constrain the DCM motion without considering the stable part to ensure stable walking, for example by constraining the DCM location at the end of a specified time $\xi(T)$ as in [12]. In this case, a terminal condition (captured state) is set at the end of a predefined number of steps, and the desired DCM at the end of the current step is recursively computed.

This is not the only way to keep the DCM from diverging. In fact, a legged robot can instantaneously change its CoP location u_0 by taking a step [33] in order to limit the DCM motion. This is for example the approach taken in [44]. In [33], it was shown that the ∞-step capturability region is only a function of the maximum step length and minimum step time. Therefore, any viable state remains viable as long as the next step timing and location are decided such that the distance between the DCM and next step location does not exceed a certain bound. It is the approach we exploit in this paper.

B. The DCM offset

We now introduce the DCM offset, which is a convenient change of variable to synthesize controllers that enforce a desired CoM average velocity

$$b = \xi_T - u_T \tag{5}$$

where u_T is the next step location and ξ_T the DCM at the end of the step (Fig. 1). The LIPM solution can be written in terms of the next footprint location, the step duration and the DCM offset by solving (2b) as a final value problem

$$u_T = (\xi_{cur} - u_0)e^{\omega_0(T-t)} + u_0 - b \quad , \quad 0 \leq t \leq T \tag{6}$$

in which ξ_{cur} is the current DCM of the robot. Assume a desired CoM average velocity given by a desired step length L, width W and duration T, then the resulting desired DCM offset in the sagittal and lateral directions is simply[3]

$$b_x = \frac{L}{e^{\omega_0 T} - 1} \tag{7a}$$

$$b_y = (-1)^n \frac{l_p}{1 + e^{\omega_0 T}} - \frac{W}{1 - e^{\omega_0 T}} \tag{7b}$$

where l_p is the default step width and $n = 1$ when the right foot is stance, and $n = 2$ when the left foot is stance. W is the deviation of the step width with respect to the pelvis width. As this value shows how much the robot moves laterally, we will call it the *step width* (Fig. 1).

C. Viability bound on the DCM offset

We now express the viability region of the LIPM [2], [51], [53] in terms of the DCM offset. Computing the viability kernel is generally intractable but fortunately, it is possible to characterize these bounds for the LIPM as it was first shown in [33], where it was characterized in terms of the ∞-step capturability region (i. e. the set of DCM/ICP states from which the system can come to a captured state by taking an infinite (or less) number of steps). The ∞-step capturability bound d_∞ is written as [33]

$$d_\infty = L_{max} \frac{e^{-\omega_0 T_{min}}}{1 - e^{-\omega_0 T_{min}}} \tag{8}$$

with maximum step length L_{max} and minimum step duration T_{min}. (8) enables the analysis of the viability of the current state of the system but is not necessarily convenient to relate viability to the step location and duration. We now write the viability bounds in terms of the DCM offset. We limit our analysis to the sagittal plane dynamics for forward walking as the analysis of backward walking is similar and the lateral direction analysis is given in Appendix B. The maximum DCM offset $b_{x,max}$ is related to the maximum step length and minimum step duration as

$$b_{x,max} = \frac{L_{max}}{e^{\omega_0 T_{min}} - 1} \tag{9}$$

Interestingly, this maximum offset separates viable and non-viable states: I) if the DCM offset is larger than $b_{x,max}$, all possible choices of step timing and location will lead to its divergence and II) if the DCM offset is smaller than (or equal to) $b_{x,max}$, there exists at least one combination of step timing and location that keeps the DCM from diverging.

Case I: If $b_{x,0} > b_{x,max}$ at the start of a step then

$$\xi_{x,0} - u_{x,0} = \frac{L_{max}}{e^{\omega_0 T_{min}} - 1} + \epsilon \tag{10}$$

where $\epsilon > 0$. Using (4), (5) and (10), the DCM offset at the end of the step is

$$b_{x,T} = (\frac{L_{max}}{e^{\omega_0 T_{min}} - 1} + \epsilon) e^{\omega_0 T} - (u_{x,T} - u_{x,0}) \tag{11}$$

Substituting $u_{x,T} - u_{x,0} = L_{max}$ and $T = T_{min}$, the minimum realizable offset $b_{x,T}$ is therefore

$$b_{x,T} = \frac{L_{max}}{e^{\omega_0 T_{min}} - 1} + \epsilon \, e^{\omega_0 T_{min}} \tag{12}$$

Comparing (10) and (12), we see that the minimum realizable DCM offset at the end of the step increases by $e^{\omega_0 T_{min}}$. A sequence of steps will therefore result in a diverging geometric series with common ratio $e^{\omega_0 T_{min}}$ and all possible choices of step location and timing lead to divergence (i.e. a fall).

Case II: If $b_{x,0} \leq b_{x,max}$ at the start of a step then

$$\xi_{x,0} - u_{x,0} \leq \frac{L_{max}}{e^{\omega_0 T_{min}} - 1} \tag{13}$$

Using (4), (5) and (13), we find

$$b_{x,T} \leq (\frac{L_{max}}{e^{\omega_0 T_{min}} - 1}) e^{\omega_0 T} - (u_{x,T} - u_{x,0}) \tag{14}$$

Selecting the next step position and timing as $u_{x,T} - u_{x,0} = L_{max}$ and $T = T_{min}$, we find

$$b_{x,T} \leq \frac{L_{max}}{e^{\omega_0 T_{min}} - 1} \tag{15}$$

which shows that for any state satisfying (13), there exists at least one choice of step position and timing that keeps the next DCM offset bounded by $b_{x,max}$. The state is viable.

Remark The DCM offset bound (9) is equal to the ∞-step capturability bound (8) derived in [33]. However, it affords a different interpretation: the viability kernel defined by the DCM offset bound is the closure of the ∞-step capturability set. It contains states that never lead to a fall but that are not ∞-step capturable because they cannot lead to a stop even in an infinite number of steps.

III. Stepping controller

The stepping controller (Fig. 2) has three main stages: 1) we compute nominal step location and duration, and DCM offset for a desired walking velocity, 2) using these nominal values, we compute optimal desired step duration and location at each control cycle and 3) we adapt the swing foot trajectories used in a whole body controller to generate walking (Sec. IV).

A. First stage: nominal values for stepping

We find desired set of step length, width and duration that ensure a desired average walking velocity while satisfying the robot and environment constraints. The problem can be formulated in terms of nominal step length L_{nom}, width W_{nom}, and duration T_{nom} as

$$\begin{aligned} &v_x = \frac{L_{nom}}{T_{nom}} \quad , \quad v_y = \frac{W_{nom}}{T_{nom}} \\ &L_{min} \leq L_{nom} \leq L_{max} \\ &W_{min} \leq W_{nom} \leq W_{max} \\ &T_{min} \leq T_{nom} \leq T_{max} \end{aligned} \tag{16}$$

where v_x and v_y are the desired average walking velocities in sagittal and lateral directions. Step location bounds, L_{min}, L_{max}, W_{min} and W_{max}, are set according to the robot (limited

[3] cf. Appendix A for the derivation details

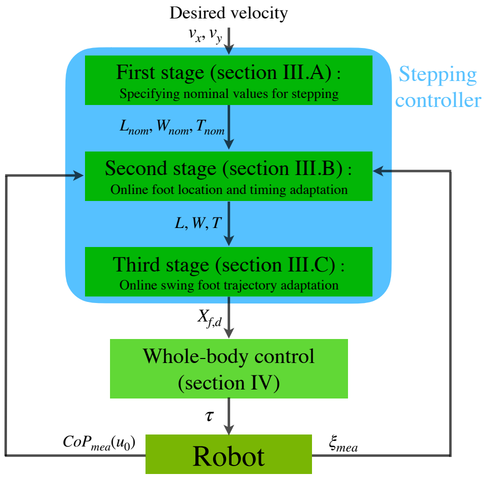
Desired velocity
vx, vy
First stage (section III.A) :
Specifying nominal values for stepping
Stepping controller
Lnom, Wnom, Tnom
Second stage (section III.B) :
Online foot location and timing adaptation
L, W, T
Third stage (section III.C) :
Online swing foot trajectory adaptation
Xf,d
Whole-body control
(section IV)
τ
CoPmea(u0)
Robot
ξmea

Fig. 2. Block diagram of the walking algorithm.

step length and width) and environment (limited area for stepping) limitations. Minimum step timing limits swing foot acceleration and maximum step timing limits slow stepping.

We compute these nominal values to keep them as far as possible from their boundaries

$$T_{nom} = \frac{B_l + B_u}{2} \tag{17}$$
$$L_{nom} = v_x(\frac{B_l + B_u}{2}) \quad , \quad W_{nom} = v_y(\frac{B_l + B_u}{2})$$

where we chose

$$B_l = \max\{\frac{L_{min}}{|v_x|}, \frac{W_{min}}{|v_y|}, T_{min}\} \quad , \quad v_x, v_y \neq 0$$
$$B_u = \min\{\frac{L_{max}}{|v_x|}, \frac{W_{max}}{|v_y|}, T_{max}\} \quad , \quad v_x, v_y \neq 0$$

and we remove the term including v_x or v_y from these equations ,when they are equal to zero. The resulting desired DCM offsets are then computed using (7).

B. Second stage: online foot location and timing adaptation

This stage adapts foot step timing and location based on the DCM measurement at each control cycle (typically 1 KHz on a torque controlled robot) to enforce viability constraints. Introducing the following change of variable for step timing

$$\tau = e^{\omega_0 T} \quad \text{or} \quad T = \frac{1}{\omega_0}\log(\tau) \tag{18}$$

Equation (6) becomes linear in the decision variables τ, b, u_T

$$u_T - (\xi_{mea} - u_0)e^{-\omega_0 t}\tau + b = u_0 \quad , \quad 0 \leq t \leq T \tag{19}$$

where t is the time elapsed since the beginning of the actual swing phase. The step adaptation problem is then the solution to the quadratic program

$$\begin{aligned}
\underset{u_T,\tau,b}{\text{argmin}} \quad & \alpha_1 \| u_T - u_0 - \begin{bmatrix} L_{nom} \\ W_{nom} \end{bmatrix} \|^2 + \alpha_2 |\tau - \tau_{nom}|^2 \\
& + \alpha_3 \| b - \begin{bmatrix} b_{x,nom} \\ b_{y,nom} \end{bmatrix} \|^2 \\
\text{s.t.} \quad & \begin{bmatrix} L_{min} \\ W_{min} \end{bmatrix} \leq u_T - u_0 \leq \begin{bmatrix} L_{max} \\ W_{max} \end{bmatrix} \\
& e^{\omega_0 T_{min}} \leq \tau \leq e^{\omega_0 T_{max}} \\
& u_T + b = (\xi_{mea} - u_0)e^{-\omega_0 t}\tau + u_0 \\
& \begin{bmatrix} b_{x,min} \\ b_{y,max,out} \end{bmatrix} \leq b \leq \begin{bmatrix} b_{x,max} \\ b_{y,max,in} \end{bmatrix}
\end{aligned} \tag{20}$$

where $b_{x,nom}$ and $b_{y,nom}$ are the nominal DCM offsets in sagital and lateral direction computed in Sec. III-A. The last inequality in (20) is the viability condition implemented as a hard constraint (cf. Appendix B for $b_{y,max,in}$ and $b_{y,max,out}$ definitions). It guarantees that the system remains viable provided that the current state is viable. In practice, we implement this constraint as a soft constraint with a very high weight with respect to the other cost terms to guarantee that the program will always return a solution even if viability cannot be maintained. This ensures that measurement noise or differences between the LIPM and the real robot do not lead to a controller failure and enables to subsequently activate a fall management strategy is necessary. Lexicographic optimization [13] could also be employed in this case but this makes little numerical difference.

C. Third stage: Swing foot trajectory adaptation

We adapt foot trajectories to follow changing step location and timing. Foot trajectories horizontal to the ground are simply represented with fifth order polynomials so trajectories are continuous in acceleration, which is important for inverse dynamics control. In the vertical direction, the swing foot height increases until the middle of the step and then decreases to land on the ground. Simply using two fifth order polynomials for each part of this trajectory would lead to two important problems: 1) a change in step timing close to mid-time can cause a jump from the first spline to the second one and 2) a change of step timing in the second part generates unavoidable fluctuations in the vertical direction which may cause ground-foot collisions. We use instead a 9th order polynomial for the whole step where we keep the swing foot height at the mid-time of the step as close as possible to the desired step height and enforce a stricly positive foot height that is lower than a maximum height. This results in the following QP

$$\begin{aligned}
\underset{c_i}{\text{argmin}} \quad & \| z_{f,d}(T/2) - z_{des} \|^2 \\
\text{s.t.} \quad & 0 \leq z(t) \leq z_{max} \\
& z_{f,d}(0) = 0 \quad z_{f,d}(t_{k-1}) = z_{k-1} \quad z_{f,d}(T) = 0 \\
& \dot{z}_{f,d}(0) = 0 \quad \dot{z}_{f,d}(t_{k-1}) = \dot{z}_{k-1} \quad \dot{z}_{f,d}(T) = 0 \\
& \ddot{z}_{f,d}(0) = 0 \quad \ddot{z}_{f,d}(t_{k-1}) = \ddot{z}_{k-1} \quad \ddot{z}_{f,d}(T) = 0
\end{aligned} \tag{21}$$

where $z_{f,d}$ is the vertical component of the swing foot, $k-1$ is the previous sample time and T is the adapted step timing.

The coefficients c_i of the polynomial are computed at each control cycle and we evaluate the polynomial at the current time to obtain the current desired foot position.

IV. Whole body control

We use hierarchical inverse dynamics to control the desired swing foot motion, while enforcing robot constraints, non-slipping contacts and constant CoM height. We specify a hierarchy of desired task space behaviors and constraints expressed as linear equalities and inequalities and the controller computes the resulting optimal joint accelerations, contact forces and actuation torques. We describe in the following our control tasks and constraints, all details of the controller can be found in [22].

A. Foot trajectory tracking

The swing foot tracking task ensures tracking of the desired motions computed in Section III-C. The task is written as

$$J_{sw}\ddot{q}+\dot{J}_{sw}\dot{q} = \ddot{X}_{f,d}+K_d(\dot{X}_{f,d}-\dot{X}_f)+K_p(X_{f,d}-X_f) \quad (22)$$

where $\dot{q}$ is the configuration of the robot, X_f and $X_{f,d}$ are the actual and reference swing foot positions, and J_{sw} is the swing foot Jacobian. K_p and K_d are diagonal gain matrices. We do not control swing foot orientation because we will apply approach to a robot with passive ankles without enough DOFs to control the orientation. For the stance foot, the task is to remain on the ground

$$J_{st}\ddot{q} + \dot{J}_{st}\dot{q} = 0 \quad (23)$$

where J_{st} is the stance foot Jacobian. This task keeps the stance foot in a stationary contact with the ground surface.

B. Center of Mass height tracking

While it is not necessary to control the horizontal motion of the center of mass, we want to keep a desired CoM height with the following task

$$J_{CoM}\ddot{q} + \dot{J}_{CoM}\dot{q} = K_p(z_d - z) - K_d\dot{z} \quad (24)$$

where J_{CoM} is the CoM height Jacobian, z is the CoM height and z_d is set to the CoM height used in the LIPM ($z_d = z_0$).

C. Posture control

We ensure a straight robot posture with the task

$$\ddot{q} = K_p(q_d - q) - K_d\dot{q} \quad (25)$$

where K_p and K_d are non zero diagonal gain matrices for the actuated joints.

D. Force regularization

During the double support phase, we exploit contact force redundancy to find forces that distribute the contact forces among the end-effectors in contact by defining the task objective $\lambda = F_{des}$, where F_{des} is the desired contact force. We put this task in the lowest priority (see Table I) such that acts only as a regularizer when relevant.

E. Task hierarchy

The hierarchy used for all our experiments is shown in Table I. We give the highest priority to physical consistency and actuation limits. The second rank enforces stance foot contact constraints and CoM height. The swing foot control, which essentially generates the walking motions is in the third rank. The last two ranks are used for the posture control and force regularization.

TABLE I
Hierarchy of tasks used in the humanoid simulation.

Rank	Nr. of eq/ineq constraints	Constraint/Task
1	6 eq	Centroidal momentum dynamics
	2×4 ineq	Torque limits
2	6 eq	Stance foot constraint
	1 eq	CoM height control
3	3 eq	Swing foot control
4	2×4 eq	Posture control
5	2×6 eq	Force regularization

V. LIPM simulation

In this section, we present simulation results using the LIPM simulation, where (1) is integrated and u_0 is reset by u_T at time T computed from (20). In our implementation of (20), we use a large α_3 compared to α_1 and α_2 to favor solutions close to the desired walking speed. For all reported experiments, we set $\alpha_1 = 1$, $\alpha_2 = 5$, $\alpha_3 = 1000$. Since we optimize the time of the current step at each control cycle, we need to prevent instantaneous stepping (and prevent step timing that are shorter than the current step). Therefore, we apply the adapted gait variables if $T \geq t + T_{gap}$, where t is the current time and T_{gap} is a user defined time gap. Otherwise we use the previous step values. Additionally, towards the end of a step, when the time gap is smaller than the remaining step time, we stop foot location or timing adaptation until the end of the step. We noticed in our experiments that this was sufficient to avoid time adaptation jitter.
The main goal in this section is to show the performance of our gait controller independent of the whole body controller. We study push recovery capabilities and show the importance of step timing adaptation. We then compare our controller with a standard MPC-based walking controller [20].

A. Simulation results using the LIPM

We simulate the LIPM with our controller using a set desired velocity. Footsteps and swing foot trajectories are computed as described in Section III. During each step, the stance foot is set as the point of contact of the LIPM. At the end of a step where the point of contact of the LIPM is changed, the foot index n is changed. We apply pushes on the robot and compare the recovery capabilities of the controller with and without step timing adjustment. We set the mass to 60 Kg and the CoM height to 80 cm, Table II defines the other variables (the step location limitations are specified with respect to the stance foot) which correspond to the Sarcos humanoid robot used in the next section.

We set a forward desired velocity ($v_x = 1\,\mathrm{ms}^{-1}$) and compute the nominal step length and step duration using (17),

TABLE II
PHYSICAL PROPERTIES OF THE ABSTRACT MODEL.

Value	Description	min	max
L	Step length	$-50\,\mathrm{cm}$	$50\,\mathrm{cm}$
W_{right}	Step width (right)	$-10\,\mathrm{cm}$	$20\,\mathrm{cm}$
W_{left}	Step width (left)	$-20\,\mathrm{cm}$	$10\,\mathrm{cm}$
T	Step duration	$0.2\,\mathrm{s}$	$0.6\,\mathrm{s}$

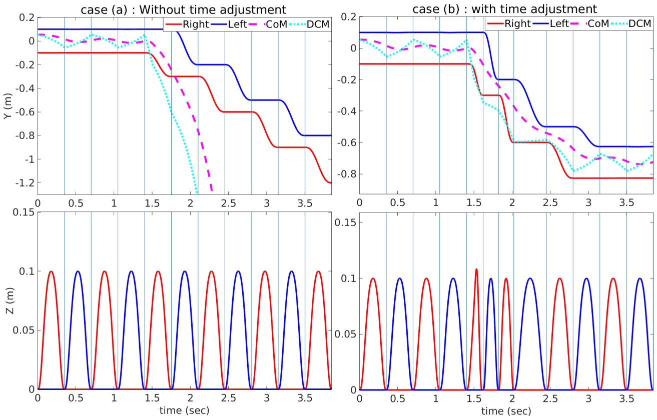

Fig. 3. Comparison of trajectories with and without time adjustment. Top: left and right feet, CoM and DCM horizontal positions. Bottom: left and right feet vertical trajectories. The vertical lines show the step duration.

as well as the nominal DCM offset using (7). After four steps, the robot is pushed at $t = 1.4\,\mathrm{s}$ to the right direction with a force $F = 325\,\mathrm{N}$, during $\Delta t = 0.1\,\mathrm{s}$. We conduct two simulations to compare the results of fixed and optimized step duration. In one case, we use (6) for the step adjustment, using the current DCM measurement. In a second case, we solve the optimization problem in (20) at each control cycle to generate the desired time and location of the next step.

Figure 3 illustrates the resulting trajectories for each case. We observe that without timing adjustment, the robot steps on the borders of the feasible area to recover from the push but because step timing is fixed, the DCM ends up diverging. In the case where timing adjustment is enabled, the algorithm constantly adapts the next footstep and landing time. As a result, the robot steps on the borders of the feasible area very fast to recover from the push. We notice that our approach results in smooth swing foot trajectories (without any discontinuity in position or velocity), despite the constant step location and timing adjustment.

Figure 4 shows the adapted step time at each control cycle. The nominal stepping time for this scenario is $T_{nom} = 0.35\,\mathrm{s}$. When the push is exerted at $t = 1.4\,\mathrm{s}$, the optimizer quickly decreases step timing. Interestingly, the optimizer increases step timing after push rejection above the nominal step time ($T = 0.446\,\mathrm{s}$). Indeed, at $t = 2.019\,\mathrm{s}$ when the push is rejected, the DCM is very close to the stance foot location (Fig. 3, top-right) and the DCM offset far from its nominal value. Hence, the optimizer increases stepping time to bring the robot back to its nominal speed.

B. Comparison with a standard MPC walking controller

We compare the robustness of our controller to the approach proposed in [20], which can be considered, together with variations of this technique [26], [52], [11] as a standard, state of the art, walking pattern generator. This controller automatically generates step locations and CoM trajectory for a desired walking velocity. Step timing, however, remains fixed. In the paper, a horizon of $N = 16$ time intervals of length $T = 0.1\,\mathrm{s}$ is used. We note here that solving the quadratic program for this problem with a 1.6 second horizon is considerably more expensive than our optimization approach. Indeed, considering CoM jerk and foot location as decision variables in 16 time step horizon yields $4 \times 16 = 64$ decision variables at each control cycle without timing adaptation, while our approach only has 6 decision variables.

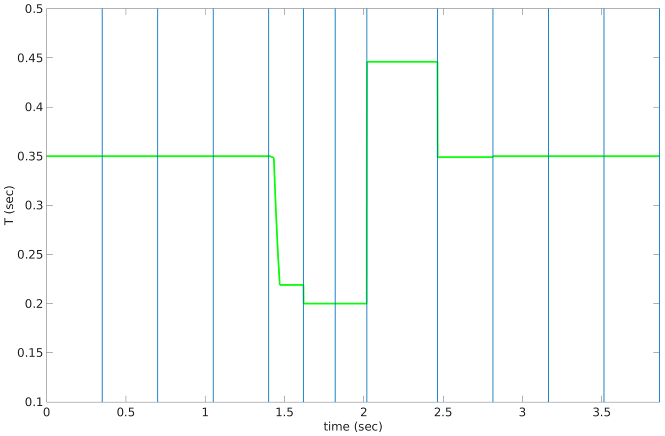

Fig. 4. The adapted step time value T versus the simulation time t. The vertical lines show the times at which the feet switch.

We used the same parameters for both approaches using a LIPM with point contact and computed the maximum push that each approach can recover from in various directions (Fig. 5). For each simulation, a force during $\Delta t = 0.1\,\mathrm{s}$ is applied at the start of a step in which the left foot is stance. We use $T_{nom} = 0.5\,\mathrm{s}$ (computed from the first stage of our algorithm) for both approaches.

Figure 5 shows that our walking controller, which only adapts the next step location and time, can recover from much more severe pushes compared to the approach with a preview of several steps but without step timing adaptation. For side pushes, the direction of the push affects the maximum value of the push that the robot is able to withstand. Indeed, for certain pushes (outward direction, left in this case) the feasible stepping area is more limited than the other direction (inward direction) due to leg self-collisions. The maximum push in the direction $\theta = 75\,\mathrm{deg}$ is larger than the other directions because the feasible area for stepping in this direction is larger than the other directions.

We also tested the controller [20] with minimum step timing T_{min} as nominal behavior and, as expected, it leads to the exact same robustness results as our approach. In the case of a very severe push, both algorithms yield stepping on the boundaries of the feasible area. This confirms our claim that it is sufficient to solely consider the next foot step timing and location. Moreover, in our approach step timing is kept as close as possible to the nominal step time and the shortest step time is only selected when necessary, which should lead to more desirable behaviors in general. Indeed, constantly

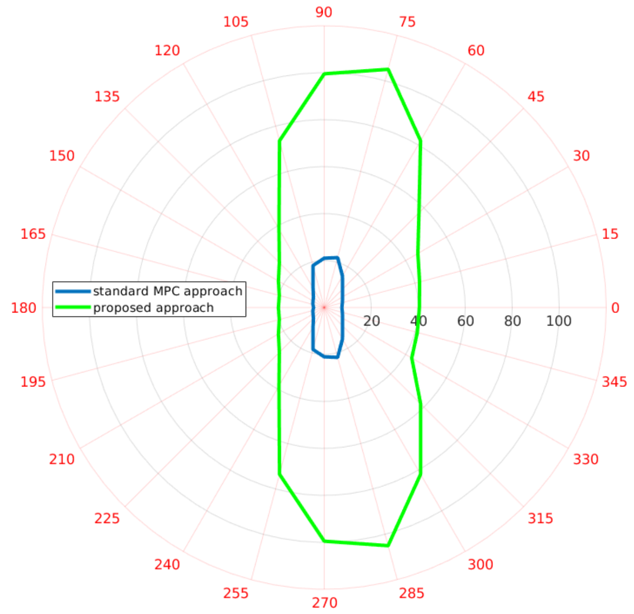

Fig. 5. Comparison of maximum push impulse (in N.s) before falling. $\theta = 90\,\text{deg}$ corresponds to a forward push, while $\theta = 0\,\text{deg}$ and $\theta = 180\,\text{deg}$ represent pushes to the right and left directions, respectively.

stepping with minimum step time puts an extra strain on the system. Actuators will work at their maximum capacity all the time, augmenting the risk of failure and energy consumption. Furthermore, increasing the frequency of stepping can make the system more vulnerable to complex terrains with surface unevenness. In contrast, our algorithm steps at maximum speed only when absolutely necessary.

VI. Humanoid with Passive Ankles Simulation

A. Description of the simulations

In this section, we use a simulation of a Sarcos humanoid with passive ankles in the SL simulation environment [47]. Contacts are simulated with a penalty method with linear springs and dampers. We used 18 contact points for each foot. The controller has only access to the resultant wrench, simulating a 6-axis force sensor. All experiments are performed on a 2.7 GHz intel i7 processor with 16Gb of RAM. The quadratic program (20) is solved using a slightly adapted version of the QuadProg++ software, which implements an active set method described in [16]. Each leg of the robot has 4 active degrees of freedom with passive ankle joints and prosthetic feet. We simulate the passive ankle joints with stiff springs and dampers. Actuation torques are computed using the controller described in Section IV.

We conduct different simulations with various external disturbances. Since the robot has finite size feet (i.e. the CoP can move inside the foot), we set the current contact point u_0 in our controller as the current CoP measurement. The constraints and physical properties are the same as for the LIPM experiments, except that the minimum step duration is now set to $T_{min} = 0.3\,\text{s}$ to account for acceleration limits of the robot. We test two scenarios: push recovery and slippage recovery. In the first scenario, the pelvis is pushed during stepping in different directions. In the second scenario, the stance foot is pushed such that slippage occurs.

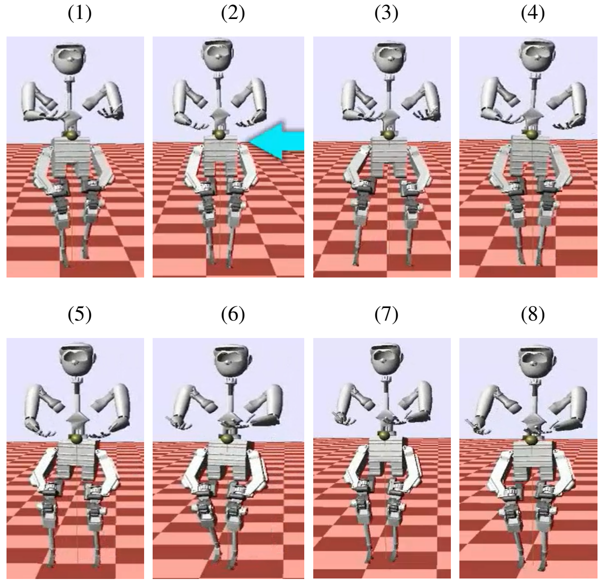

Fig. 6. Push recovery example: the robot walks forward at $v_x = 0.2\,\text{ms}^{-1}$, the pelvis is pushed at $t = 3.7\,\text{s}$ by $F = 200\,\text{N}$ during $\Delta t = 0.1\,\text{s}$.

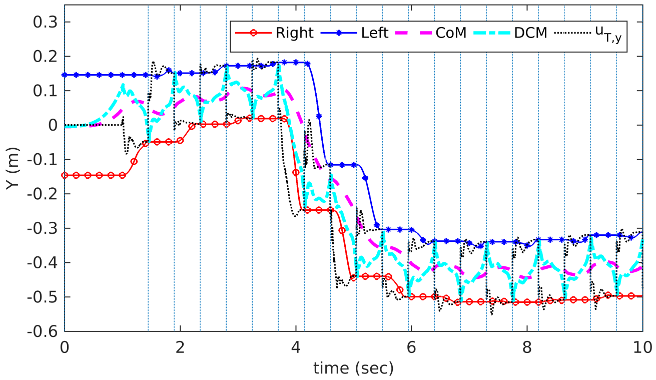

Fig. 7. First push recovery experiment: lateral trajectories during forward walking without step timing adjustment. The desired lateral velocity is zero in this simulation, however, when the push occurs ($F = 200\,\text{N}$, at $t = 3.7\,\text{s}$ during $\Delta t = 0.1\,\text{s}$ which causes an impulse of 20 N.s), the controller sacrifices lateral velocity tracking to recover balance. This push (impulse of 20 N.s) is the maximum lateral (inward) disturbance that the robot could recover from without timing adjustment.

B. Push recovery

We compare our controller with fixed stepping duration and then with timing adaptation to better understand the influence of step timing adaptation on push recovery capabilities. Fig. 6 shows a typical side-step adaptation behavior. To recover from the push, the robot automatically starts stepping to the right direction. Once the push is rejected, the robot resumes its forward walking.

Fig. 7 and Fig. 8 show push recovery capabilities without time adaptation. The maximum lateral (inward) disturbance the robot can withstand is $200\,\text{N}$ for $\Delta t = 0.1\,\text{s}$. We notice that feet trajectories are adapted to reach the desired landing locations. After the push, the controller sacrifices lateral velocity tracking and adjusts foot positions to recover balance. The

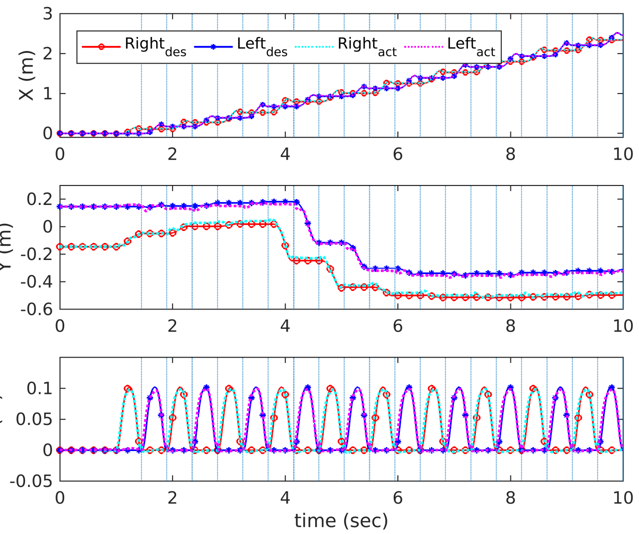

Fig. 8. First push recovery experiment: desired and actual feet trajectories during forward walking without step timing adjustment. The low-level controller tracks the smooth feet trajectories generated with the walking controller.

adapted feet trajectories remain smooth resulting in smooth control input generated by the whole body controller.

Fig. 9 and Fig. 10 show the push recovery results when step timing adaptation is enabled. In this case, the maximum lateral (inward) disturbance that the robot was able to withstand was a push of $390\,\mathrm{N}$ for $\Delta t = 0.3\,\mathrm{s}$, which is nearly six times the impulse that the robot could withstand when timing was not adapted. This is consistent with the results found for the LIP model (Fig. 5 with $\theta = 0\,\mathrm{deg}$). This result further illustrates the importance of timing adaptation for stable walking. Indeed, inspecting (3) we see that the DCM diverges as an exponential of time, therefore taking fast steps (decreasing the step timing) enables an exponential improvement in DCM regulation. This effect is magnified when the robot takes several steps to recover from a disturbance.

We note a slight difference between these two simulation scenarios. Indeed, the vertical lines in the figures show that for the scenario with timing adaptation the push is exerted in the middle of a step ($t = 3.9\,\mathrm{s}$ in Fig. 9), while in the scenario without timing adaptation the disturbance is exerted exactly at the start of a step ($t = 3.7\,\mathrm{s}$ in Fig 7). Since the DCM diverges exponentially fast, the later the push during a specified step, the more difficult the recovery (e. g. the DCM diverges more when time evolves from 0.2 s to 0.4 s than 0 s to 0.2 s). As a result, although the experimental condition is more difficult in the case of timing adaptation, the robot recovers from a push nearly six times larger than the case without timing adaptation.

Fig. 9 shows that both step location and timing are adjusted. To recover from the push, the robot takes five steps with minimum step time to the left on the boundaries of the feasible area. The feet trajectories (Fig. 10) are again adapted very smoothly in the case when step timing is adjusted. Furthermore, the step height is also adapted when step duration is changed, consistently with the constraints specified in (21). Trajectory tracking in the vertical direction degrades compared to the case without timing adjustment (Fig. 8), which is due to an increase in the desired swing foot acceleration. However, such trajectory tracking performance in the vertical direction is sufficient for feasible stepping.

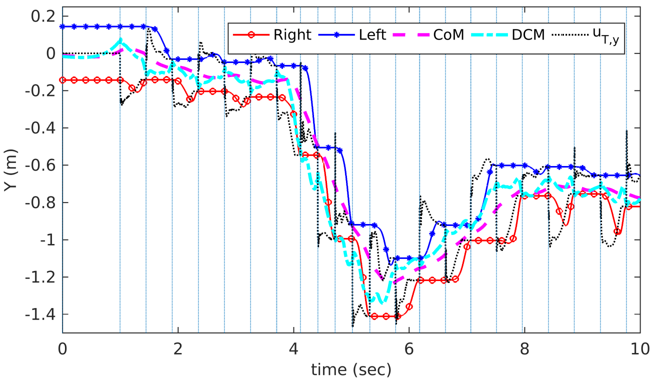

Fig. 9. Second push recovery experiment: The lateral trajectories during stepping in place with both step location and timing adjustment. The desired velocity is zero during this simulation. When the push ($F = 390\,\mathrm{N}$, at $t = 3.9\,\mathrm{s}$ during $\Delta t = 0.3\,\mathrm{s}$) is exerted, the controller adapts both step location and timing to recover from this severe lateral (inward) push. This disturbance (impulse of $117\,\mathrm{N.s}$) is the maximum lateral (inward) push that the robot could recover from with step timing adjustment in our simulations.

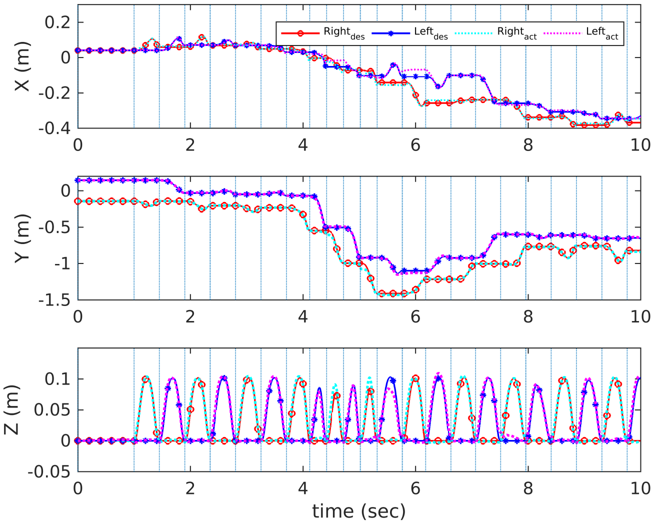

Fig. 10. Second push recovery experiment: The desired and actual feet trajectories during stepping in place with both step location and timing adjustment. The low-level controller tracks the smooth feet trajectories generated by our real-time walking controller. We can see that when the step timing is adapted, the step height is adapted to satisfy the constraints specified in (21).

Fig. 9 shows that after rejecting the exerted push by stepping to the right, the robot starts (around $t = 6\,\mathrm{s}$) to step back in left direction. This behavior can be explained by looking at the angular momentum (see Fig. 11). The stance foot constraints are in a high rank and the whole body controller sacrifices joint posture and generates angular momentum to keep the stance foot steady on the ground. After exerting the push, angular momentum is accumulated until the push is fully rejected (around $t = 6\,\mathrm{s}$). After this, the whole body controller reduces the angular momentum to zero and brings back the joints toward the desired upright upper-body posture. This acts as a disturbance to the walking controller and the robot starts stepping in the opposite direction until the angular momentum is zero and the upper-body is upright.

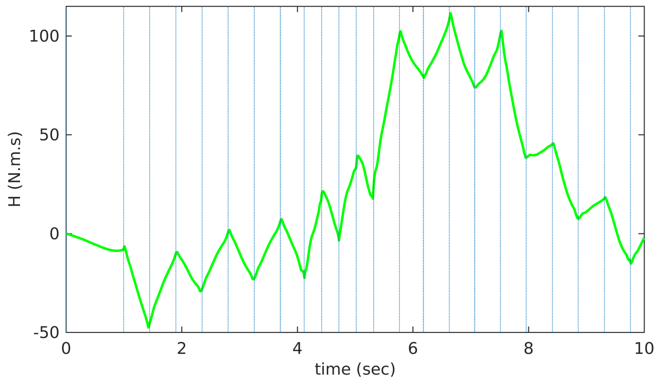

Fig. 11. Second push recovery experiment: The angular momentum trajectory around x-axis during stepping in place with both step location and timing adjustment. The desired velocity is zero during this simulation. When the push ($F = 390\,\mathrm{N}$, at $t = 3.9\,\mathrm{s}$ during $\Delta t = 0.3\,\mathrm{s}$) is exerted, the angular momentum is accumulated, until the push is fully rejected (around $t = 6\,\mathrm{s}$). After this, the whole body controller tries to make the angular momentum zero and bring back the joints as close as possible to the configuration where the upper-body is upright. This acts as a kind of disturbance to the system and the robot starts adapting the step in the opposite direction until the angular momentum is around zero and the upper-body is upright.

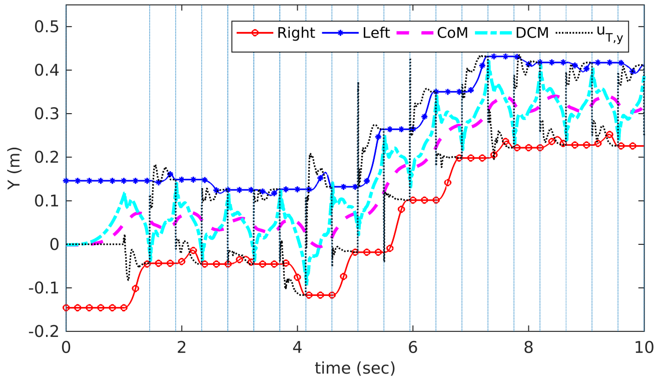

Fig. 12. First slippage recovery experiment: The lateral trajectories during a forward walking without step timing adjustment. The desired lateral velocity is zero during this simulation. The stance foot is pushed laterally by $F = 400\,\mathrm{s}$, at $t = 3.9\,\mathrm{s}$ during $\Delta t = 0.2\,\mathrm{s}$ such that slippage occurs. After the push, the foot locations are adjusted to recover the robot from the push.

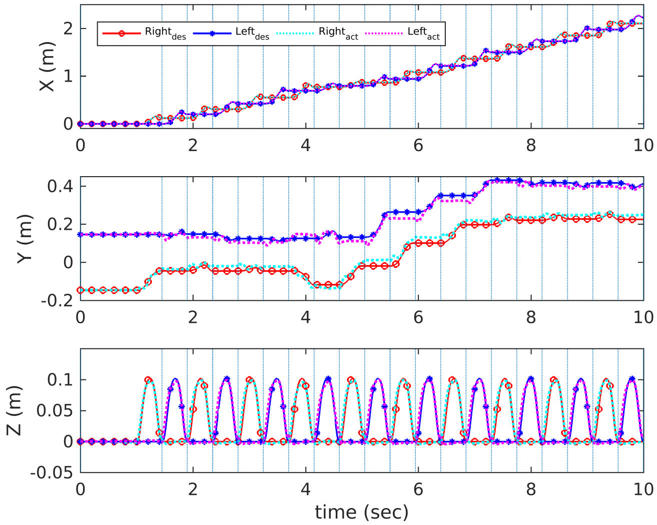

Fig. 13. First slippage recovery experiment: The desired and actual feet trajectories during a forward walking without step timing adjustment. When the push is exerted at $t = 3.9\,\mathrm{s}$, the left foot slips to the left. The right foot trajectory is then adapted to recover the robot from this disturbance.

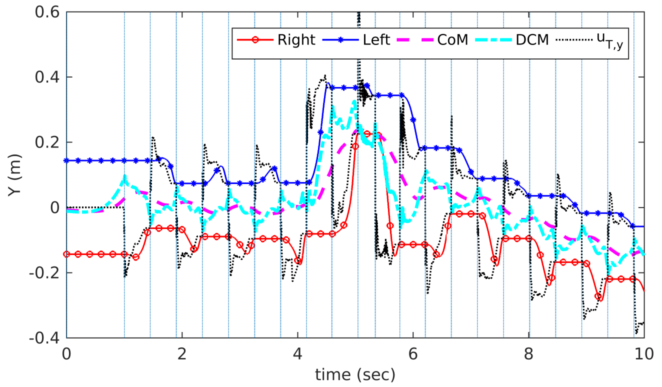

Fig. 14. Second slippage recovery experiment: The lateral trajectories during stepping in place with both step location and timing adjustment. The desired velocity is zero during this simulation. The stance foot is pushed laterally by $F = 930\,\mathrm{N}$, at $t = 3.9\,\mathrm{s}$ during $\Delta t = 0.3\,\mathrm{s}$ such that slippage occurs. After the push, the foot locations are adjusted to recover the robot from the push.High frequency oscillations of the DCM trajectory is due to the huge disturbance on the stance foot. This huge disturbance causes high frequency oscillation of the passive elements in ankles.

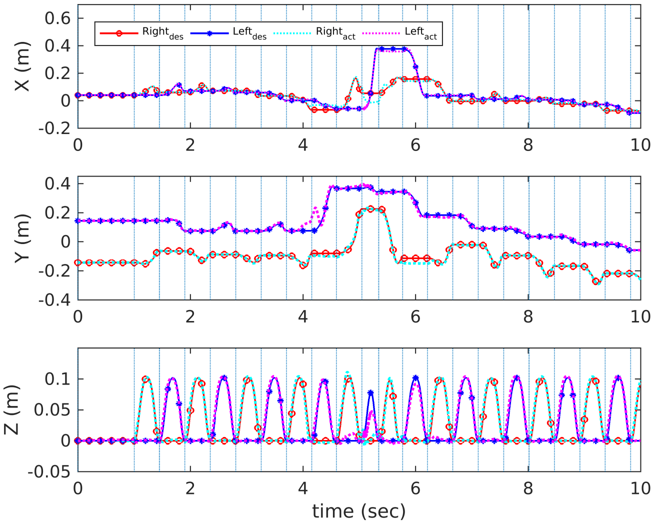

Fig. 15. Second slippage recovery experiment: desired and actual feet trajectories during stepping in place with both step location and timing adjustment. When the push is exerted at $t = 3.9\,\mathrm{s}$, the left foot slips to the left and the next steps locations are adapted to recover from the disturbance.

C. Slippage recovery

We now investigate the slippage recovery capabilities of the controller. Walking controllers based on CoP modulation tend to be very sensitive to foot slippage. Indeed, when the stance foot slips, the assumption of stationary flat foot on the ground is not valid anymore. Since our controller does not rely on CoP control it can recover from large slippage by adjusting the swing foot landing location and time. During walking, we apply strong forces on the stance foot such that it slips.

We compare performance with and without timing adaptation. For the case without timing adaptation, Fig. 12 shows a typical forward walking simulation. Slippage of the left foot can be observed in the actual foot trajectory in the lateral direction (Fig. 13) which causes a faster DCM divergence in the right direction. Hence, the right foot steps further in the right direction to recover from the disturbance. Figure 14 shows the lateral trajectories when step timing adjustment is enabled. Again, we can see a degradation in the foot trajectory tracking in lateral direction (Fig. 15). The controller exploits a combination of step timing and location adjustment to recover from this strong disturbance. Some high frequency oscillations can be seen in the DCM trajectory and the estimated next footprint (Fig. 14), which is due to the large disturbance on the stance foot. This disturbance causes high frequency oscillations of the passive elements in the ankles. Nevertheless, the robot is able to withstand an impulse on the foot which is more than three times larger than the impulse withstood when step timing is not adjusted. This again illustrates the importance of timing adjustment during walking.

VII. Discussion

A. Features of the approach

- *Generality* Our approach can generate robust gaits without modulating the foot CoP. This suggests that our approach can be used for biped robots with active ankles, passive ankle, or point contact foot. We successfully used the controller to control an underactuated biped robot with passive ankle joints and prosthetic feet.
- *Computational efficiency* The size of our optimization problem is drastically smaller than typical MPC approaches using several preview steps [26], [52], [11], [20], at least an order of magnitude smaller in the number of decision variables. Viability is ensured by optimizing the DCM offset without the need to integrate the motion forward over several time steps [54].
- *Controller receding horizon length* Our paper provides an answer to an open question for MPC-based controllers: *how far ahead in time shall our controller optimize motion for?*[56]. It is only sufficient to look at the next step to recover walking from any viable state. Interestingly, several reported experiments on human walking seem to support this argument, even in the presence of obstacles [43], [42]. However, if the terrain is very difficult or constrained (e.g. walking on stepping stones), humans seem to look two steps ahead [42].
- *Step timing vs step location adjustment* To highlight the difference between step location and timing adjustment, let's assume that the swing foot can move in all directions without any kinematic constraints and that the only constraint is a limited foot acceleration. Then, the maximum distance the swing foot can travel in a given duration is achieved using a bang-bang controller (maximum acceleration and deceleration) with equal acceleration and deceleration periods if kinematic constraints are not considered. Specifying step period T and maximum acceleration/deceleration value a_{max}, the swing foot travels $u_T = a_{max} \; \frac{T^2}{4}$ during this period. Assuming $u_0 = 0$ for the current step in the DCM dynamics, at time T the DCM is $\xi_T = \xi_0 e^{\omega_0 T}$. The DCM diverges as an exponential of time while the next step location can only be adapted with a second order time dependency.
- *Robustness against various disturbances* The measured CoP and estimated DCM are employed to adapt the landing location and time of the swing foot but the controller does not try to control the CoP. This feature enables robust walking under strong pushes and stance foot slippage. This feature might also enable the robot to walk over rough terrains, but this is left as a future work. Furthermore, our experiments showed that timing adjustment significantly increased walking robustness.
- *Comparison to existing approaches* Step timing adaptation can serve different purposes, i. e. decreasing the ZMP fluctuation in case of immediate change of step location [38], walking on uneven terrain [40], [41], [31], disturbance recovery [1], [34], [29], [18], respecting acceleration constraint of the swing foot moving on a specified path [6], or collision avoidance for walking in crowds [3]. Despite a variety of goals, the employed approaches are either based on analytical methods and heuristics [41], [38], [40], [36], [8], [18], [31] or optimization [1], [34], [6], [35], [3]. Optimization provides a principled way of taking into account all constraints while optimizing a desired performance cost. However, non-convexity of the system dynamics with respect to step timing can significantly increase computation time and is prone to getting stuck in undesired local minima. Our approach with at least one order of magnitude less decision variables than existing approaches [1], [34], [3], [6], [35] and a convex formulation circumvents both concerns and provides a reliable tool for realtime applications.
- *Viability* In [48], an approach similar to ours [29] is used for guaranteeing viability of the gait by enforcing a desired offset between the capture point and landing location of the swing foot. However, the approach to realize this offset is based on modulating the CoP of the current step, while the next step location and timing are fixed. [46] used a combination of ZMP modulation and step location adjustment to keep the capture point inside the support polygon at the end of the current step (1-step capturability). As we showed in this paper, endowing step location adaptation with timing adjustment and using exact viability constraint in the controller structure can ensure that any initially viable state remains viable.

B. Limitations

- *LIPM assumption*: the viability guarantees of our controller are valid under the assumption of a fixed CoM height and constant angular momentum at the CoM. While these assumptions restrict walking on even ground, since we do not rely on CoP modulation we anticipate that the controller will also be robust to mild rough terrain provided that the low level controller manage the uncertainty in landing. An interesting direction of research would be to use the concept of 3D-DCM and virtual repellent point (VRP) [12] instead to enable our approach to handle 3D maneuvers.
- *Timing constraint* Constraints on step locations are explicit and depend on the kinematic and environment limitations. However, constraints on step timing are a function of maximum acceleration of the swing foot as well as the distance between the current state of the swing

foot and the landing location. As a result, considering a constraint on step timing for the worst case decreases the capabilities of the controller. To circumvent this, we could use more complex, yet linear, models which take into account the swing foot dynamics [49]. Then, we could directly constrain swing foot acceleration instead of step timing. A simpler alternative is to use an empirical linear inequality constraint on the next step location and minimum time as a function of maximum foot speed and remaining stepping time [20].

- *Walking on stepping stones or in scattered environments* While we do not need to consider a multi-step horizon to guarantee viability in unconstrained environments, when the feasible foot locations are very limited, such as waking over stepping stones or in the presence of obstacles, we would need to consider multiple steps in advance to steer the DCM motion in the direction of safe contact points. Our approach can be readily extended to consider this issue, for example one could adapt step timing only in the first step within a desired horizon and optimize for the subsequent step locations with a fixed nominal step timing, therefore retaining a QP.
- *Real robot experiments* Previous works [14], [45] have shown that our modeling assumptions can be successfully used on a real robot. While this suggests that our approach would work as well, real robot experiments, especially for robots with passive ankles or point feet would be most valuable. At the time when the research was conducted, we did not have access to such robots.

VIII. Conclusion

We proposed a walking controller that adapts both step location and timing in real time to generate robust gaits and we showed that for the LIPM, optimizing the next footstep timing and location was sufficient to ensure that viable states remain viable. Comparison with a standard MPC-based walking controller emphasized the importance of step timing adaptation and demonstrated that our approach can be significantly more robust (the algorithm could in our simulation tolerate up to about 5 times larger impulsive pushes) than methods that do not adjust timing, even when those controllers optimize over several preview steps. Simulations in push and slippage recovery scenarios on a humanoid with passive ankles showed the robustness of the controller even without control authority in the ankles and the foot CoP.

Acknowledgment

This research was supported by the European Unions Horizon 2020 research and innovation program (grant agreement 780684 and European Research Councils grant 637935) and in part by the MPI-ETH center for learning systems. We would like to thank the reviewers for their thoughtful comments on a previous version of the manuscript.

References

[1] Z. Aftab, T. Robert, and P.-B. Wieber, "Ankle, hip and stepping strategies for humanoid balance recovery with a single model predictive control scheme," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 159–164.

[2] J.-P. Aubin, "Viability theory. systems & control: Foundations & applications," *Birkhäuser, Boston. doi*, vol. 10, no. 1007, pp. 978–0, 1991.

[3] N. Bohórquez and P.-B. Wieber, "Adaptive step duration in biped walking: a robust approach to nonlinear constraints," in *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference on*. IEEE, 2017, pp. 724–729.

[4] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer, "A collocation method for real-time walking pattern generation," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2007, pp. 1–6.

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

[6] S. Caron and Q.-C. Pham, "When to make a step? tackling the timing problem in multi-contact locomotion by topp-mpc," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 522–528.

[7] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, "A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[8] J. A. Castano, Z. Li, C. Zhou, N. Tsagarakis, and D. Caldwell, "Dynamic and reactive walking for humanoid robots based on foot placement control," *International Journal of Humanoid Robotics*, vol. 13, no. 02, p. 1550041, 2016.

[9] T. De Boer, "Foot placement in robotic bipedal locomotion," Ph.D. dissertation, Technische Universiteit Delft, 2012.

[10] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[11] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1121–1126.

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

[13] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[14] S. Feng, "Online hierarchical optimization for humanoid control," Ph.D. dissertation, Carnegie Mellon University, 2016.

[15] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Robust dynamic walking using online foot step optimization," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 5373–5378.

[16] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical programming*, vol. 27, no. 1, pp. 1–33, 1983.

[17] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," *arXiv preprint arXiv:1905.06144*, 2019.

[18] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, and J. Pratt, "Walking stabilization using step timing and location adjustment on the humanoid robot, atlas," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 667–673.

[19] K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa, "An analytical method for real-time gait planning for humanoid robots," *International Journal of Humanoid Robotics*, vol. 3, no. 01, pp. 1–19, 2006.

[20] 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.

[21] A. Herdt, N. Perrin, and P.-B. Wieber, "Walking without thinking about it," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 190–195.

[22] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[23] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2703–2710.

[24] A. L. Hof, "The extrapolated center of mass concept suggests a simple control of balance in walking," *Human movement science*, vol. 27, no. 1, pp. 112–125, 2008.

[25] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, and J. Hurst, "Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot," *The International Journal of Robotics Research*, p. 0278364916648388, 2016.

[26] 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 *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2003, pp. 1620–1626.

[27] 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 *Intelligent Robots and Systems,IEEE/RSJ International Conference on*. IEEE, 2001, pp. 239–246.

[28] T. Kamioka, H. Kaneko, T. Takenaka, and T. Yoshiike, "Simultaneous optimization of zmp and footsteps based on the analytical solution of divergent component of motion," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1763–1770.

[29] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Step timing adjustment: A step toward generating robust gaits," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 35–42.

[30] M. Khadiv, S. Kleff, A. Herzog, S. A. A. Moosavian, S. Schaal, and L. Righetti, "Stepping stabilization using a combination of dcm tracking and step adjustment," in *2016 4th International Conference on Robotics and Mechatronics (ICROM)*. IEEE, 2016, pp. 130–135.

[31] M. Khadiv, S. A. A. Moosavian, A. Yousefi-Koma, H. Maleki, and M. Sadedel, "Online adaptation for humanoids walking on uncertain surfaces," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 231, no. 4, pp. 245–258, 2017.

[32] M. Khadiv, S. A. A. Moosavian, A. Yousefi-Koma, M. Sadedel, and S. Mansouri, "Optimal gait planning for humanoids with 3d structure walking on slippery surfaces," *Robotica*, vol. 35, no. 3, pp. 569–587, 2017.

[33] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[34] P. Kryczka, P. Kormushev, N. G. Tsagarakis, and D. G. Caldwell, "Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 3352–3357.

[35] M. R. Maximo, C. H. Ribeiro, and R. J. Afonso, "Mixed-integer programming for automatic walking step duration," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 5399–5404.

[36] M. Missura and S. Behnke, "Omnidirectional capture steps for bipedal walking," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 14–20.

[37] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 43, 2012.

[38] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, S. Nakaoka, and H. Hirukawa, "A biped pattern generation allowing immediate modification of foot placement in real-time," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 581–586.

[39] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[40] K. Nishiwaki and S. Kagami, "Strategies for adjusting the zmp reference trajectory for maintaining balance in humanoid walking," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4230–4236.

[41] I.-W. Park, J.-Y. Kim, J. Lee, and J.-H. Oh, "Online free walking trajectory generation for biped humanoid robot khr-3 (hubo)," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1231–1236.

[42] A. E. Patla and J. Vickers, "How far ahead do we look when required to step on specific locations in the travel path during locomotion?" *Experimental brain research*, vol. 148, no. 1, pp. 133–138, 2003.

[43] A. E. Patla and J. N. Vickers, "Where and when do we look as we approach and step over an obstacle in the travel path?" *Neuroreport*, vol. 8, no. 17, pp. 3661–3665, 1997.

[44] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, 2006, pp. 200–207.

[45] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower body humanoid," *The International Journal of Robotics Research*, p. 0278364912452762, 2012.

[46] C. Santacruz and Y. Nakamura, "Reactive stepping strategies for bipedal walking based on neutral point and boundary condition optimization," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3110–3115.

[47] S. Schaal, "The sl simulation and real-time control software package," *University of Southern California*, 2009.

[48] T. Sugihara and T. Yamamoto, "Foot-guided agile control of a biped robot through zmp manipulation," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 4546–4551.

[49] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot-1 st report: Walking gait pattern generation," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1084–1091.

[50] J. Urata, K. Nshiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online walking pattern generation for push recovery and minimum delay to commanded change of direction and speed," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3411–3416.

[51] P.-B. Wieber, "On the stability of walking systems," in *Proceedings of the international workshop on humanoid and human friendly robotics*, 2002.

[52] P. B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 137–142.

[53] P.-B. Wieber, "Viability and predictive control for safe locomotion," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1103–1108.

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

[55] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[56] P. Zaytsev, S. J. Hasaneini, and A. Ruina, "Two steps is enough: no need to plan far ahead for walking balance," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6295–6300.

Appendix A: Derivation of (7)

Using (4) and (5), the LIPM equation in terms of the DCM offset and the next step location is

$$u_T = (\xi_0 - u_0)e^{\omega_0 T} + u_0 - b \tag{26}$$

For a constant walking velocity, the desired DCM offset in the sagittal direction, b_x, is considered to be constant over several steps, therefore we have

$$u_{T,x} = b_x e^{\omega_0 T} + u_{0,x} - b_x \tag{27}$$

since $L = u_{T,x} - u_{0,x}$ over two consecutive steps, we have

$$b_x = \frac{L}{e^{\omega_0 T} - 1} \tag{28}$$

Note that this DCM offset causes a limit cycle for the robot nonlinear (switching) dynamics, in the sense that keeping L and T constant over several steps leads to a steady state forward walking velocity. For a desired sideward walking, the distance between the feet is equal to $l_p + W$ or $l_p - W$ (Fig.

1). Using (26), the equations for the right and left foot DCM offset are

$$\begin{aligned} -(l_p - W) &= b_{y,l}e^{\omega_0 T} - b_{y,r} \\ l_p + W &= b_{y,r}e^{\omega_0 T} - b_{y,l} \end{aligned} \tag{29}$$

Solving (29) for $b_{y,r}$ and $b_{y,l}$ yields

$$\begin{aligned} b_{y,r} &= \frac{l_p}{1+e^{\omega_0 T}} - \frac{W}{1-e^{\omega_0 T}} \\ b_{y,l} &= -\frac{l_p}{1+e^{\omega_0 T}} - \frac{W}{1-e^{\omega_0 T}} \end{aligned} \tag{30}$$

We use the index n to distinguish the right ($n = 1$) and left ($n = 2$) stance foot and write the DCM offset in the latteral direction as

$$b_y = (-1)^n \frac{l_p}{1+e^{\omega_0 T}} - \frac{W}{1-e^{\omega_0 T}} \tag{31}$$

Appendix B: Viability in the lateral direction

In the lateral direction, the feasible area for stepping is not symmetric with respect to the stance foot due to self collision. The maximum allowable DCM offsets for outward and inward directions are then different. Here we call outward direction the one where the legs can experience a self collision, and inward for the opposite direction. The maximum allowable DCM offsets in these directions are

$$b_{y,max,out} = \frac{l_p}{1+e^{\omega_0 T_{min}}} + \frac{W_{max} - W_{min}e^{\omega_0 T_{min}}}{1-e^{2\omega_0 T_{min}}} \tag{32a}$$

$$b_{y,max,in} = \frac{l_p}{1+e^{\omega_0 T_{min}}} + \frac{W_{min} - W_{max}e^{\omega_0 T_{min}}}{1-e^{2\omega_0 T_{min}}} \tag{32b}$$

To show that these values exactly split the state space into the viable and non-viable parts, we again consider the situations where the DCM offsets are less or more than these values for both outward and inward directions. We conduct this analysis for the case where the right foot is stance (the same results directly follow for the other cases). Considering a DCM offset more (less) than the value in (32a) for the outward direction at the start of a step

$$\xi_{y,0} - u_{y,0} = \frac{l_p}{1+e^{\omega_0 T_{min}}} + \frac{W_{max} - W_{min}e^{\omega_0 T_{min}}}{1-e^{2\omega_0 T_{min}}} \pm \epsilon \tag{33}$$

the minimum feasible DCM offset at the end of this step ($b_{y,T}$) is obtained taking a step with the minimum step width (for avoiding self collision) at the minimum step time using (26)

$$\begin{aligned} b_{y,T} =& \left(\frac{l_p}{1+e^{\omega_0 T_{min}}} + \frac{W_{max} - W_{min}e^{\omega_0 T_{min}}}{1-e^{2\omega_0 T_{min}}} \pm \epsilon\right)e^{\omega_0 T_{min}} \\ &- (l_p + W_{min}) \end{aligned} \tag{34}$$

Using this value as the DCM offset at the start of next step, and taking a step with the maximum step width at the minimum step time yield the minimum feasible DCM offset of the next step

$$\begin{aligned} b_{y,2T} = &\left(\frac{l_p}{1+e^{\omega_0 T_{min}}} + \frac{W_{max} - W_{min}e^{\omega_0 T_{min}}}{1-e^{2\omega_0 T_{min}}} \pm \epsilon\right) \\ &e^{2\omega_0 T_{min}} - (l_p + W_{min})e^{\omega_0 T_{min}} + (l_p + W_{max}) \end{aligned} \tag{35}$$

which simplifies into

$$b_{y,2T} = \frac{l_p}{1+e^{\omega_0 T_{min}}} + \frac{W_{max} - W_{min}e^{\omega_0 T_{min}}}{1-e^{2\omega_0 T_{min}}} \pm \epsilon e^{2\omega_0 T_{min}} \tag{36}$$

Comparing (33) with (36) reveals that starting from a DCM offset more than (32a) (positive ϵ in (33)) and after taking two steps on the boundaries of the feasible area at the minimum time, the DCM offset increases by a factor $e^{2\omega_0 T_{min}}$ times ϵ. As a result, any DCM offset more than (32a) for causes divergence. However, starting from a DCM offset less than the value in (32a) (negative ϵ in (33)), there exists one evolution (stepping on the boundaries of the feasible area at the minimum time) that keeps the DCM from diverging. Hence, (32a) for the outward direction exactly splits the state space into viable and non-viable parts.

Majid Khadiv is a postdoc scholar at the Movement Generation and Control Group at the Max-Planck Institute for Intelligent Systems. He received his BSc degree in Mechanical Engineering from Isfahan University of Technology (IUT), in 2010, and his MSc and PhD degree in Mechanical Engineering from K. N. Toosi University of Technology, Tehran, Iran in 2012 and 2017. Majid joined the Iranian national humanoid project, Surena III, and worked as the head of dynamics and control group from 2012 to 2015. He also spent a one-year visiting scholarship under supervision of Dr. Ludovic Righetti at the Autonomous Motion Laboratory (AMD), Max-Planck Institute for Intelligent Systems. His main research interest is control of legged robots.

Alexander Herzog is a roboticist at X, Inc. (Google). He did his PhD at the Max Planck Institute for Intelligent Systems, Tübingen and received his doctorate degree from ETH Zürich. Dr Herzog studied Computer-Science at the Karlsruhe Institute of Technology, in Germany. He visited the Computational Learning and Motor Control Lab (University of Southern California) regularly from 2011 to 2016 where he collaborated in projects on grasp learning and locomotion for humanoid robots. After receiving his Diploma in 2011, he joined the Autonomous Motion Laboratory at the Max-Planck Institute for Intelligent Systems in 2012. During his PhD he worked on contact interaction in whole-body control and grasping for humanoids.

S. Ali. A. Moosavian received his B.Sc. degree in 1986 from Sharif University of Technology and the M.Sc. degree in 1990 from Tarbiat Modaress University (both in Tehran), and his Ph.D. degree in 1996 from McGill University (Montreal, Canada), all in Mechanical Engineering. He is a Professor with the Mechanical Engineering Department at K. N. Toosi University of Technology (KNTU) in Tehran since 1997. His research interests are in the areas of dynamics modeling and motion/impedance control of terrestrial, legged and space robotic systems. He has published more than 200 articles in peer-reviewed journals and conference proceedings. He is a Member of IEEE, and one of the Founders of the ARAS Research Group, and the Manager of Center of Excellence in Robotics and Control at KNTU.

Ludovic Righetti is an Associate Professor in the Electrical and Computer Engineering Department and in the Mechanical and Aerospace Engineering Department at the Tandon School of Engineering at New York University and a Senior Researcher at the Max-Planck Institute for Intelligent Systems in Tübingen, Germany. He holds an engineering diploma in Computer Science and a Doctorate in Science from the Ecole Polytechnique Fédérale de Lausanne (Switzerland). His research focuses on the planning and control of movements for autonomous robots, with a special emphasis on legged locomotion and manipulation.