Dynamic Walking over Rough Terrains by Nonlinear Predictive Control of the Floating-base Inverted Pendulum

Stéphane Caron, Abderrahmane Kheddar

▶ **To cite this version:**

Stéphane Caron, Abderrahmane Kheddar. Dynamic Walking over Rough Terrains by Nonlinear Predictive Control of the Floating-base Inverted Pendulum. 2017. hal-01481052v1

HAL Id: hal-01481052
https://hal.science/hal-01481052v1

Preprint submitted on 2 Mar 2017 (v1), last revised 22 Jul 2017 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

Dynamic Walking over Rough Terrains by Nonlinear Predictive Control of the Floating-base Inverted Pendulum

Stéphane Caron[1] and Abderrahmane Kheddar[1,2]

***Abstract*— We present a real-time rough-terrain dynamic walking pattern generator. Our method automatically finds step durations, which is a critical issue over rough terrains where they depend on terrain topology. To achieve this level of generality, we introduce the Floating-base Inverted Pendulum (FIP) model where the center of mass can translate freely and the zero-tilting moment point is allowed to leave the contact surface. We show that this model is equivalent to the linear-inverted pendulum mode with variable center of mass height, aside from the fact that its equations of motion remain linear. Our design then follows three steps: (i) we characterize the FIP contact-stability condition; (ii) we compute feedforward controls by solving a nonlinear optimization over receding-horizon FIP trajectories. Despite running at 30 Hz in a model-predictive fashion, simulations show that the latter is too slow to stabilize dynamic motions. To remedy this, we (iii) linearize FIP feedback control computations into a quadratic program, resulting in a constrained linear-quadratic regulator that runs at 300 Hz. We finally demonstrate our solution in simulations with a model of the HRP-4 humanoid robot, including noise and delays over both state estimation and foot force control.**

I. Introduction

A walking pattern is *dynamic* when it contains single-support phases that are not statically stable, *i.e.* the center of mass (COM) of the robot leaves the area above contact and undergoes divergent dynamics. These dynamic phases can be used not only to increase the walking speed, but also for stabilization as illustrated by reactive stepping strategies [1], [2]. To measure the dynamic capabilities of a rough-terrain walking pattern generator[1] (RT-WPG), we can measure the relative duration of double-support phases, or the amount of time spent in statically-stable configurations. For instance, our previous RT-WPG [3] spends roughly 40% of its gait in double-support, and about 90% of the time in statically-stable configurations. In contrast, our present RT-WPG, in a similar setup, spends less than 5% of its gait in double-support, and about 40% of the time in statically-stable configurations.

One reason why previous solutions tended to be conservative lies in enforcing contact stability, *i.e.* making sure that contacts sustain while the robot pushes on them to move. To generate contact-stable trajectories, one needs to guarantee that all contact wrenches throughout the motion lie inside their respective contact wrench cones. So far, the full problem has only been solved in whole-body motion generation [4], or more recently in centroidal motion generation [5], [6], [7], where computation time is provided until a solution is found. For real-time control, it is common to reduce the number of variables by regulating the centroidal angular momentum to $\dot{\mathbf{L}}_G = 0$. Doing so simplifies the equations of motion of the COM to:

$$\ddot{\boldsymbol{p}}_G = \lambda(\boldsymbol{p}_G - \boldsymbol{p}_Z) + \boldsymbol{g},$$

with $\boldsymbol{p}_G$ the COM position, λ a positive quantity, $\boldsymbol{p}_Z$ the whole-body zero-tilting moment point (ZMP) and $\boldsymbol{g}$ the gravity vector. When the COM motion is constrained to lie on a plane, λ is constant and we obtain the Linear Inverted Pendulum Mode (LIPM). Predictive control of the COM in the LIPM can be formulated as a quadratic program, where the cost function encodes a number of desired behaviors while inequalities enforce the contact-stability condition: the ZMP lies within the convex hull of contact points. (This condition is actually incomplete; we will derive the complete condition below.) This formulation successfully solved the problem of walking over flat surfaces [8], [9], [10]. However, it did not extend to the rough terrains where the shape of ZMP support areas varies during motion [11].

Solutions for rough terrains were proposed that tracked reference COM or base-link trajectories across contact switches [12], [13], yet they were not designed to compute their own feedforward controls. In pattern generation, developments went in three directions. In one line of work, 3D extensions of the LIPM [14], [15] provided the basis for the first RT-WPGs, but they enforced only two out of the three requirements for contact stability, namely unilaterality and center-of-pressure conditions (friction is missing). Other works went for harder nonlinear optimization problems [16], [17], [18], but again they did not model friction and were only applied to walking on parallel horizontal surfaces. Finally, we recently proposed in [3] an RT-WPG that enforces full contact stability and walks across arbitrary terrains, but at the cost of a conservative problem linearization (a similar idea appeared in the motion generator from [6]).

We now bridge the gap between these three directions with an RT-WPG that is (1) based on a 3D extension of the LIPM for which we (2) derive the full contact stability condition and consequently (3) formulate and solve as a nonlinear optimal-control problem. The latter being experimentally too slow for predictive control, we further derive a constrained linear-quadratic regulator based on the same model for high-frequency stabilization.

*This work is supported in part by H2020 EU project COMANOID http://www.comanoid.eu/, RIA No 645097.

[1]CNRS-UM2 LIRMM, IDH group, UMR5506, Montpellier, France.

[2]CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/RL.

Corresponding author: stephane.caron@normalesup.org

[1]Pattern generators compute both feedforward and feedback walking controls under real-time constraints, as opposed to motion generators, which only compute feedforward controls without time constraints.

II. The Floating-base Inverted Pendulum

Let us consider a biped in single support. We define the surface patch $\mathcal{S}$ as the convex hull of contact points, and denote by $\mathcal{C}$ the contact friction cone. In the pendulum mode, the center of pressure (COP) C is located at the intersection between $\mathcal{S}$ and the central axis of the contact wrench, which is then also a zero-moment axis. Contact breaks when this intersection becomes empty, or switches to another mode when C reaches the boundaries of $\mathcal{S}$. The Newton equation of motion of the COM in the pendulum mode is:

$$\ddot{\boldsymbol{p}}_G \;=\; \lambda(\boldsymbol{p}_G - \boldsymbol{p}_C) + \boldsymbol{g}. \tag{1}$$

In general, this equation is bilinear as the stiffness value λ and COP location $\boldsymbol{p}_C$ are two different components of the time-varying contact wrench. Making time explicit, the differential equation of the COM position is:

$$\ddot{\boldsymbol{p}}_G(t) - \lambda(t)\boldsymbol{p}_G(t) \;=\; -\lambda(t)\boldsymbol{p}_C(t) + \boldsymbol{g}$$

The additional constraint of the LIPM is that $\boldsymbol{p}_G$ and $\boldsymbol{p}_C$ lie in parallel planes separated by a fixed distance $h = \boldsymbol{n} \cdot (\boldsymbol{p}_G - \boldsymbol{p}_C)$, so that $\lambda = (\boldsymbol{n} \cdot \boldsymbol{g})/h$ becomes a constant by Equation (1). Interestingly, Koolen *et al.* [19] recently studied the symmetric problem where $\boldsymbol{p}_C$ is fixed and a variable $\lambda(t)$ is used to stabilize the COM along a cubic polynomial.

A. Contact stability in single support

Proposition 1: A motion of the system (1) in single contact $(\mathcal{S}, \mathcal{C})$ is contact-stable *if and only if*:

$$\lambda \;\in\; \mathbb{R}^+ \tag{2}$$
$$\boldsymbol{p}_C \;\in\; \mathcal{S} \tag{3}$$
$$\boldsymbol{p}_G \;\in\; \boldsymbol{p}_C + \mathcal{C} \tag{4}$$

where $\mathcal{S}$ and $\mathcal{C}$ respectively denote the surface patch and friction cone of the contact.

Proof: This result follows from injecting Equation (1) into the analytical formula of the single-contact wrench cone [20]. See Appendix A for calculations. ∎

The constraints (2)–(4) are written in V-representation for a geometric intuition. Denoting by $\mathbf{S}$ and $\mathbf{C}$ the H-representation matrices of the polygon $\mathcal{S}$ and cone $\mathcal{C}$, respectively, we can formulate these constraints equivalently in H-representation as:

$$-\lambda \;\leq\; 0 \tag{2H}$$
$$\mathbf{S}\boldsymbol{p}_C \;\leq\; 1 \tag{3H}$$
$$\mathbf{C}(\boldsymbol{p}_G - \boldsymbol{p}_C) \;\leq\; 0 \tag{4H}$$

Equation (4)-(4H) provides the condition that is missing in previous works [14], [15], [16], [17], [18] to model friction. It also completes the observation we made in [11] (Figure 6) by showing that, on horizontal floors, the points $\boldsymbol{p}_C$ where $\boldsymbol{p}_G \notin \boldsymbol{p}_C + \mathcal{C}$ are *exactly* those that need to be removed from the convex hull of ground contact points to obtain the ZMP support area. The area thus admits a direct geometric construction:

Corollary 1: When contacts are coplanar, the ZMP support area $\mathcal{Z}$ is the intersection between the surface patch and the backward friction cone rooted at the COM:

$$\mathcal{Z} \;=\; \mathcal{S} \cap (\boldsymbol{p}_G - \mathcal{C}) \tag{5}$$

Proposition 1 also gives us a geometric construction of the COM static-equilibrium polygon in single support:

Corollary 2: The COM static-equilibrium polygon in single support is either:

- empty when the friction cone $\mathcal{C}$ does not contain the vertical $\boldsymbol{g}$, or
- equal to the vertical projection of the surface patch $\mathcal{S}$ onto a horizontal plane.

Proof: This result immediately follows from Proposition 1 by recalling that, in static equilibrium, the COP C is located at the vertical below the COM G. ∎

Proposition 1 shows how contact-stability inequalities, which are bilinear in general, linearize without loss of generality when rewriting the contact wrench in terms of λ and $\boldsymbol{p}_C$. It is the main result we use in this paper, as dynamic walking patterns spend most of the gait cycle in single support. Note that this linearization is not specific to single support: bilinear inequalities in multi-contact are linear whenever the contact wrench is expressed in terms of an attractor/repulsor point and a stiffness coefficient, as we show in Appendix B for the interested reader. The advantage of the single-support case is that we have derived an analytical formula, otherwise we rely on numerical polytope projections.

B. Transferring nonlinearity

Equations (1)–(4) characterize the pendulum mode under full contact stability:

COP-based Inverted Pendulum

$$\ddot{\boldsymbol{p}}_G(t) = \lambda(t)(\boldsymbol{p}_G(t) - \boldsymbol{p}_C(t)) + \boldsymbol{g}$$
$$\text{s.t.} \begin{cases} -\lambda(t) \;\leq\; 0 \\ \mathbf{S}\boldsymbol{p}_C(t) \;\leq\; 1 \\ \mathbf{C}(\boldsymbol{p}_G(t) - \boldsymbol{p}_C(t)) \;\leq\; 0 \end{cases}$$

This system is linearly constrained, but its forward equation of motion is nonlinear due to the product between λ and $\boldsymbol{p}_C$. We transform it by replacing C with the ZMP[2] Z defined by:

$$\boldsymbol{p}_Z \;=\; \boldsymbol{p}_C + \left[1 - \frac{\lambda}{\omega^2}\right](\boldsymbol{p}_G - \boldsymbol{p}_C) \tag{6}$$

where ω^2 denotes a user-defined positive constant, for instance chosen as g/ℓ with ℓ the leg length of the robot. In the pendulum mode, this definition coincides with the Enhanced Centroidal Moment Pivot [15] associated with the constant ω^2. This transformation has the benefit of making the forward equation of motion linear:

$$\ddot{\boldsymbol{p}}_G(t) \;=\; \omega^2(\boldsymbol{p}_G(t) - \boldsymbol{p}_Z(t)) + \boldsymbol{g} \tag{7}$$

It does not eliminate nonlinearity, but merely transfers it to the system's inequality constraints.

[2]Recall that all points of the zero-moment axis (GC) can be called "zero-moment points".

Proposition 2: A motion of the system (7) in single contact $(\mathcal{S}, \mathcal{C})$ is contact-stable if *and only if*:

$$\boldsymbol{p}_Z \in \boldsymbol{p}_G + \mathsf{rays}(\mathcal{S} - \boldsymbol{p}_G) \quad (8)$$
$$\boldsymbol{p}_G \in \boldsymbol{p}_Z + \mathcal{C} \quad (9)$$

where $\mathcal{S}$ and $\mathcal{C}$ respectively denote the surface patch and friction cone of the contact.

Proof: We proceed by double-implication between the systems (1)–(4) and (7)–(9). ($\Rightarrow$) Let us rewrite Equation (6) as

$$\boldsymbol{p}_Z - \boldsymbol{p}_G = \frac{\lambda}{\omega^2}(\boldsymbol{p}_C - \boldsymbol{p}_G) \quad (10)$$

In this form, it is clear that (2) $\wedge$ (3) $\Rightarrow$ (8). Left-multiplying by $\mathbf{C}$, this equation further shows that $\mathbf{C}(\boldsymbol{p}_G - \boldsymbol{p}_Z)$ and $\mathbf{C}(\boldsymbol{p}_G - \boldsymbol{p}_C)$ have the same sign, therefore (2) $\wedge$ (4) $\Rightarrow$ (9). ($\Leftarrow$) For the reciprocal implication, let us write the inverse transform on the stiffness coefficient:

$$\lambda = \omega^2 \frac{\boldsymbol{n} \cdot (\boldsymbol{p}_G - \boldsymbol{p}_Z)}{\boldsymbol{n} \cdot (\boldsymbol{p}_G - \boldsymbol{p}_C)} = \omega^2 \frac{\boldsymbol{n} \cdot (\boldsymbol{p}_G - \boldsymbol{p}_Z)}{\boldsymbol{n} \cdot \boldsymbol{p}_G - a} \quad (11)$$

where $\boldsymbol{n} \cdot \boldsymbol{p} = a$ is the equation of the supporting plane of the surface patch $\mathcal{S}$. Note that $\boldsymbol{n}$ is both the plane normal and the inner axis-vector of the friction cone $\mathcal{C}$. Thus, with the implicit assumption that the COM is not located below contact (which is not physically possible), $\boldsymbol{n} \cdot \boldsymbol{p}_G > a$ and (9) imply that $\lambda \geq 0$. Then, Equation (10) shows once again that $\mathbf{C}(\boldsymbol{p}_G - \boldsymbol{p}_Z)$ and $\mathbf{C}(\boldsymbol{p}_G - \boldsymbol{p}_C)$ have the same sign, so that (9) $\Rightarrow$ (4). Finally, the COP being located at the intersection between (GZ) and the supporting plane of $\mathcal{S}$, (8) implies (3) by construction. ■

Denoting by $\{V_i\}$ the vertices of the surface patch $\mathcal{S}$, the two conditions (8)–(9) can be written equivalently in H-representation as:

$$\forall i,\ \overrightarrow{V_iV_{i+1}} \cdot (\overrightarrow{GV_i} \times \overrightarrow{V_iZ}) \leq 0 \quad (8H)$$
$$\mathbf{C}(\boldsymbol{p}_G - \boldsymbol{p}_Z) \leq 0 \quad (9H)$$

Using the transform (6), we have therefore reformulated the COP-based inverted pendulum into an equivalent ZMP-based model where the ZMP is allowed to leave the surface patch. We coin this model the Floating-based Inverted Pendulum (FIP):

Floating-base Inverted Pendulum

$$\ddot{\boldsymbol{p}}_G(t) = \omega^2(\boldsymbol{p}_G(t) - \boldsymbol{p}_Z(t)) + \boldsymbol{g}$$
$$\text{s.t.} \begin{cases} \forall i,\ \overrightarrow{V_iV_{i+1}} \cdot (\overrightarrow{GV_i} \times \overrightarrow{V_iZ}) \leq 0 \\ \mathbf{C}(\boldsymbol{p}_G - \boldsymbol{p}_Z) \leq 0 \end{cases}$$

The forward equation of motion of the FIP is linear, which will prove useful when we implement a constrained linear-quadratic regulator (LQR) using the single-shooting method. Its friction constraint is linear as well. Its ZMP constraint is bilinear and involves a COM-varying boundary cone, but it can be handled efficiently by present primal-dual interior point methods.

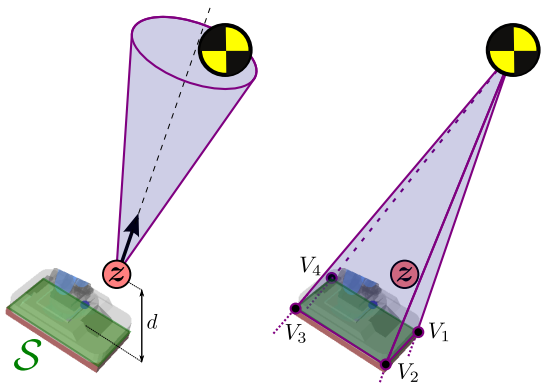

Fig. 1. The two necessary and sufficient conditions for contact stability in the Floating-base Inverted Pendulum model. **Friction (left):** the COM belongs to the contact friction cone $\mathcal{C}$ projected from the ZMP $\boldsymbol{z}$. **Center of pressure (right):** the ZMP $\boldsymbol{z}$ belongs to the cone projected from the COM and containing the vertices of the contact surface $\mathcal{S}$.

III. Nonlinear Predictive Control of the FIP

We now formulate FIP predictive control as a nonlinear program (NLP). Our main motivation in switching from convex [3] to non-convex optimization is twofold: on the one hand, solving the bilinear COP constraint (8H) without approximation, and on the other hand, deriving step timings as output variables rather than user-defined parameters. This second feature is crucial over rough terrains, where proper timings depend on terrain topology. Adapting step timings has been recently realized for walking on horizontal floors using quadratic programming [21], but it had not been demonstrated yet on uneven terrains.

A. Multiple shooting formulation

As in [5], we formulate the nonlinear predictive control (NMPC) problem by direct multiple shooting. A receding horizon over future system states is divided into N time intervals of durations $\Delta t[k]$, so that each interval $k \in \{0, \ldots, N-1\}$ starts at time $t[k] = \sum_{j<k} \Delta t[j]$. We use brackets $x[i]$ to index NLP variables and subscripts x_i for constants. The variables of our NLP are:

- $\boldsymbol{p}_G[k]$: COM position at time $t[k]$,
- $\dot{\boldsymbol{p}}_G[k]$: COM velocity at time $t[k]$,
- $\boldsymbol{p}_Z[k]$: ZMP position at time $t[k]$,
- $\Delta t[k]$: step duration, bounded by $[\Delta t_{\mathsf{min}}, \Delta t_{\mathsf{max}}]$.

The differential equation of the FIP (7) is solved over each interval with constant $\boldsymbol{p}_Z(t) = \boldsymbol{p}_Z[k]$ to obtain the matching conditions:

$$\boldsymbol{p}_G[k+1] = \boldsymbol{p}_G[k] + \dot{\boldsymbol{p}}_G[k]\frac{\mathrm{sh}[k]}{\omega} + \ddot{\boldsymbol{p}}_G[k]\frac{\mathrm{ch}[k]-1}{\omega^2} \quad (12)$$
$$\dot{\boldsymbol{p}}_G[k+1] = \dot{\boldsymbol{p}}_G[k]\mathrm{ch}[k] + \ddot{\boldsymbol{p}}_G[k]\frac{\mathrm{sh}[k]}{\omega} \quad (13)$$

where the decision variables $\boldsymbol{p}_Z[k]$ and $\Delta t[k]$ appear in:

$$\ddot{\boldsymbol{p}}_G[k] = \omega^2(\boldsymbol{p}_G[k] - \boldsymbol{p}_Z[k]) + \boldsymbol{g} \quad (14)$$
$$\mathrm{ch}[k] = \cosh(\omega \Delta t[k]) \quad (15)$$
$$\mathrm{sh}[k] = \sinh(\omega \Delta t[k]) \quad (16)$$

The next constraints that need to be enforced over at collocation points are friction and COP inequalities (8)–(9). It is important to note here that contact-stability is only checked at collocation times $t[k]$, as done in the vast majority of present works that solve constrained optimal control problems [5], [6], [7], [9], [10], [11], [13], [16], [18]. This does not guarantee that the constraints will not be violated between $t[k]$ and $t[k+1]$.

Having at hand a solver able to handle both linear and nonlinear constraints, we tried two variants for the friction cone (9):

- **FC1:** the linear constraints (9H) corresponding to the linear approximations of the friction cone $\mathcal{C}$;
- **FC2:** the second-order isotropic friction cone, *i.e.* without linear approximation, which can be written as:

$$\|\overrightarrow{ZG}\|_2^2 - (1+\mu^2)(\overrightarrow{ZG}\cdot\boldsymbol{n}) \;\leq\; 0. \tag{17}$$

Using a second-order inequalities reduces the constraint dimension: an n-sided friction pyramid FC1 yields n inequalities in the NLP, versus only one with a second-order cone. In our implementation, computations are roughly 10% faster when using FC2 compared to FC1.

B. Boundary conditions

We include both initial and terminal conditions in our control problem:

- $\boldsymbol{p}_G[0]$ is equal to the estimated COM position at the beginning of the control cycle,
- $\dot{\boldsymbol{p}}_G[0]$ is equal to the estimated COM velocity at the beginning of the control cycle,
- $\boldsymbol{p}_Z[N-1] = \boldsymbol{\xi}^d_{\mathsf{end}}$ the desired capture point at the end of the preview window.

The ability to define capture points is another advantage of the FIP compared to models with nonlinear forward equations of motion. From Equation (7), and following the derivation from *e.g.* [15], the instantaneous capture point in the FIP is:

$$\boldsymbol{\xi}(t) \;=\; \boldsymbol{p}_G(t) + \frac{\dot{\boldsymbol{p}}_G(t)}{\omega} + \frac{\boldsymbol{g}}{\omega^2}. \tag{18}$$

The boundary value $\boldsymbol{\xi}^d_{\mathsf{end}}$ is derived from the next contact location and a reference walking velocity v^d provided by the user. Speficically, if $\boldsymbol{p}'_F$ is the next contact location and $(\boldsymbol{t}, \boldsymbol{b}, \boldsymbol{n})$ the next contact frame, then:

$$\boldsymbol{\xi}^d_{\mathsf{end}} = \boldsymbol{p}'_F + \frac{v^d \boldsymbol{t}}{\omega}. \tag{19}$$

It corresponds to a desired COM position G^d located at the vertical above F', along with a forward COM velocity equal to v^d. Note that, due to the COP constraint (8), v^d should be small enough compared to ω so that $\boldsymbol{\xi}^d_{\mathsf{end}}$ belongs to the surface patch $\mathcal{S}$.

The terminal condition $\boldsymbol{p}_Z[N-1] = \boldsymbol{\xi}^d_{\mathsf{end}}$ is used so that, if the NMPC stops providing updated preview trajectories for some reason (which may happen as we are solving non-convex problems), at least regulation around the latest successful preview will steer the system to a stop. One could replace this simple post-preview behavior with more general *boundedness constraints* [22], based *e.g.* on heuristic post-preview ZMP trajectories based on terrain topology.

C. Cost function

The cost function of our NLP is a weighted combination of three integral and one terminal terms:

$$\int_{t=0}^{T} (w_Z \|\boldsymbol{p}_Z - \boldsymbol{p}_F\|^2 + w_G \|\ddot{\boldsymbol{p}}_G\|^2 + w_T)\mathrm{d}t + w_\xi \|\xi[N] - \xi^d_{\mathsf{end}}\|^2$$

- $\int \|\boldsymbol{p}_Z - \boldsymbol{p}_F\|^2 \mathrm{d}t$, where F is the center of the supporting foot (note that there may be two different supporting foot in a preview window due to contact switches). This term biases trajectories toward the nominal behavior defined by ω^2. We give it a weight of $w_Z = 10^{-5}$.
- $\int \|\ddot{\boldsymbol{p}}_G\|^2 \mathrm{d}t$, a regularization term used to avoid unnecessarily high accelerations. We give it a weight of $w_G = 10^{-3}$.
- $T = \int 1 \mathrm{d}t$, the total duration of the trajectory. This term plays a significant role in balancing the acceleration regularization term, which otherwise generates local minima where the system tries to avoid moving altogether. We give it a weight of $w_T = 10^{-2}$.
- $\|\boldsymbol{\xi}[N] - \boldsymbol{\xi}^d_{\mathsf{end}}\|^2$, where $\boldsymbol{\xi}[N]$ is the capture point defined by the COM state $(\boldsymbol{p}_G[N], \dot{\boldsymbol{p}}_G[N])$ at the end of the preview window. This term is the state analog of the ZMP boundary condition. The problem is better conditioned when it is put in the cost function rather than as a hard constraint. We give it a weight of $w_\xi = 1$.

D. Contact switches

Contrary to horizontal-floor solutions that preview several future footsteps [9], [10], our NMPC previews exactly one step ahead during single-support phases (we use our conservative linear MPC [3] during double-support phases). Its NLP does not model the swing foot trajectory. Rather, it relies on an estimate of the time to heel strike, which we construct as in [23]:

- A polynomial path is interpolated between $\boldsymbol{p}_S$ and the next foot step (spherical linear interpolation is used to interpolation swing-foot orientation as well).
- Time-Optimal Path Parameterization (TOPP) [24] is used to generate a path timing under conservative foot acceleration constraints.
- The duration T_{swing} of the retimed path is taken as estimate for the time to heel strike.

The N time intervals of the receding horizon are then split into two categories. The first half of them is dedicated to the swing interval $[0, T_{\mathsf{swing}}]$ until heel strike, where contact stability is enforced with respect to the current supporting foot, while for the second half it is enforced with respect to the next foothold. This assignment is matched with step durations by the last constraint of our NLP:

$$\sum_{k=0}^{N/2} \Delta t[k] \;\geq\; T_{\mathsf{swing}}. \tag{20}$$

E. Implementation details

We release our source code at [25] for open review and reproducibility. In this implementation, we construct NLPs using the CasADi symbolic framework [26] and solve them with the primal-dual interior-point solver IPOPT [27]. Major settings that allowed us to reach fair computation times include:

- Using MX rather than SX CasADi symbols.
- Using the MA27 or MA97 linear solvers within IPOPT.
- Capping the CPU time and number of solver iterations to 100 ms and 100, respectively. When these budgets are exceeded, the solver has most likely diverged away from any feasible solution.
- Using the adaptive rather than monotone update strategy for the barrier parameter, which made computations roughly 40% faster.

With this implementation, it takes roughly 10 to 40 ms to solve an NMPC problem (see Section V for details). These computation times are on the same scale as those reported in the warm-started phase of [5]. We don't suffer a significant cold start in our setting, most likely because the problem we solve is smaller (single-contact) and we have reformulated its structure concisely with the FIP model.

F. Tunings for variable time steps

Each step duration $\Delta t[k]$ in the NLP is lower and upper bounded by two parameters Δt_{min} and Δt_{max} that affect solver performances. Expectedly, computation time increases with Δt_{max}, but this parameter cannot be too low as the problem becomes infeasible below a certain threshold $\Delta t^{\mathsf{lim}}_{\mathsf{max}}$. Figure 2 shows how computation times[3] are influenced by varying Δt_{max} at the beginning of a feasible single-support phase. A "sweet range" extends from 200 to 500 ms, with the best (also riskiest) performance obtained close to the threshold. Below this range, computation times and failure rate increase beyond usable values. In practice, values of $\Delta t^{\mathsf{lim}}_{\mathsf{max}}$ ranged between 100 ms and 300 ms during the gait cycle. We therefore adopted a uniform setting of $\Delta t_{\mathsf{max}} = 350\,\mathrm{ms}$.

The influence of Δt_{min} is of a different nature. First, we note that Δt_{min} cannot be equal to zero in practice: all integral terms in the cost function bias convergence toward $\Delta t[k] \to 0$, so that the solver then ends up in local optima that shrink the receding horizon rather than exploring motion. Setting Δt_{min} to the control cycle gives good performance in practice. Larger values further improve computation times at the beginning of single-support phases, but jeopardize convergence in the middle of the step where T_{swing} becomes small. We dealt with this case by reducing Δt_{min} on the fly for swing-interval time steps when the bound incurs an overshoot of the time to heel strike:

if $\sum_{k=0}^{N/2} \Delta t[k] > (1 + \frac{1}{4}) T_{\mathsf{swing}}$ **then**
  $\forall k \leq N/2,\ \Delta t_{\mathsf{min}}[k] \leftarrow \Delta t_{\mathsf{min}}[k]/2$
end if

[3]All computations reported in this paper were run on a personal laptop computer, CPU: Intel(R) Core(TM) i7-6500U CPU @ 2.50 Ghz.

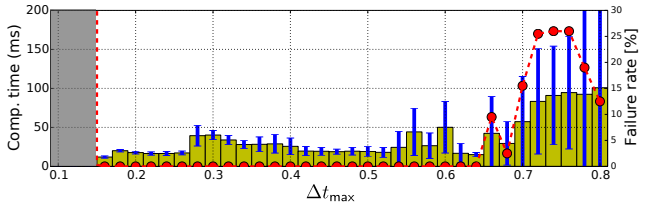

Fig. 2. Effect of Δt_{max} on computation times (yellow) and failure rate (red) of the NLP solver, for a feasible step and $N = 10$ collocation points. Each bar includes a standard deviation estimate (blue line) computed over 200 runs. Above a minimum value $\Delta t^{\mathsf{lim}}_{\mathsf{max}}$, the problem becomes feasible and all runs should ideally converge to a solution. In practice, computations become unstable for large values of Δt_{max}.

In a motion generation scenario where computation time is abundant, one could devise global strategies such as a bisection search to tune Δt_{min} and Δt_{max}. The heuristic above is rather a local parameter search spread over control cycles. It has the benefit of incurring no additional cost.

G. Failure rate accross the gait cycle

While computation times look promising, another metric suggests that nonlinear optimization is not sufficient in itself to solve the NMPC problem: its failure rate, in our case, is around 40%. This means that, on average one out of two to three control cycles, the solver either does not terminate, or returns a certificate of infeasibility, or converges to an aberrant solution. We found that the terminal capture-point error $\|\boldsymbol{\xi}[N] - \boldsymbol{\xi}^d_{\mathsf{end}}\|$ is a good indicator of this latter case, and chose to discard all previews where this error is above 10 cm (a liberal value close to the half-length of HRP-4's footprint).

IV. Constrained Linear-Quadratic Regulation

To cope with the 40% of situations where the nonlinear optimization fails to produce a new preview on time, we design a constrained linear-quadratic regulator that updates the last available preview $\boldsymbol{p}^d_G, \dot{\boldsymbol{p}}^d_G, \boldsymbol{p}^d_Z$ into a new feasible one starting from the current state. In order to cast the regulation problem as a quadratic program, this reference trajectory is first resampled into M time steps of equal duration ΔT. Let us define residual states and controls of the FIP as:

$$\Delta \boldsymbol{x}[k] = \begin{bmatrix} \boldsymbol{p}_G[k] - \boldsymbol{p}^d_G[k] \\ \dot{\boldsymbol{p}}_G[k] - \dot{\boldsymbol{p}}^d_G[k] \end{bmatrix} \tag{21}$$

$$\Delta \boldsymbol{z}[k] = \boldsymbol{p}_Z[k] - \boldsymbol{p}^d_Z[k] \tag{22}$$

The discretized linear dynamics of these FIP residuals are:

$$\Delta \boldsymbol{x}[k+1] = \mathbf{A}\Delta \boldsymbol{x}[k] + \mathbf{B}\Delta \boldsymbol{z}[k] \tag{23}$$

$$\mathbf{A} = \begin{bmatrix} \cosh(\omega\Delta T)\mathbf{E} & \sinh(\omega\Delta T)/\omega\mathbf{E} \\ \omega\sinh(\omega\Delta T)\mathbf{E} & \cosh(\omega\Delta T)\mathbf{E} \end{bmatrix} \tag{24}$$

$$\mathbf{B} = \begin{bmatrix} (1-\cosh(\omega\Delta T))\mathbf{E} \\ -\omega\sinh(\omega\Delta T)\mathbf{E} \end{bmatrix} \tag{25}$$

where $\mathbf{E}$ is the 3×3 identity matrix. Inequality constraints over $\boldsymbol{p}_G$ and $\boldsymbol{p}_Z$ translate into similar constraints over the residuals $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{z}$. The friction constraint (9H) becomes:

$$[\,\mathbf{C}\ \ \mathbf{0}\,]\,\Delta\boldsymbol{x}[k] - \mathbf{C}\Delta\boldsymbol{z}[k] \ \leq\ \mathbf{C}(\boldsymbol{p}^d_Z[k] - \boldsymbol{p}^d_G[k]) \tag{26}$$

Let us define $\sigma_i[k] \stackrel{\text{def}}{=} -\overrightarrow{V_iV_{i+1}} \cdot (\overrightarrow{G^d[k]V_i} \times \overrightarrow{V_iZ^d[k]})$ the positive slackness of the COP constraint on the i^{th} vertex in the reference trajectory. Expanding (8H) yields, in geometric form (we omit indexes $[k]$ to alleviate notations and write $\Delta\boldsymbol{p}$ the first three coordinates of $\Delta\boldsymbol{x}$):

$$\overrightarrow{V_iV_{i+1}} \cdot (\overrightarrow{G^dV_i} \times \Delta\boldsymbol{z} + \overrightarrow{V_iZ^d} \times \Delta\boldsymbol{p} + \Delta\boldsymbol{z} \times \Delta\boldsymbol{p}) \;\leq\; \sigma_i[k] \quad (27)$$

And in matrix form:

$$\Delta\boldsymbol{p}[k]^T \mathbf{H}_i \Delta\boldsymbol{z}[k] + \boldsymbol{h}_P[k]^T \Delta\boldsymbol{p}[k] + \boldsymbol{h}_Z[k]^T \Delta\boldsymbol{z}[k] \leq \sigma[k] \quad (28)$$

with $\mathbf{H}_i$ the cross-product matrix of $\overrightarrow{V_iV_{i+1}}$ and

$$\boldsymbol{h}_P[k] \stackrel{\text{def}}{=} \mathbf{H}_i(\boldsymbol{p}_Z^d[k] - \boldsymbol{p}_{V_i}),$$
$$\boldsymbol{h}_Z[k] \stackrel{\text{def}}{=} \mathbf{H}_i(\boldsymbol{p}_{V_i} - \boldsymbol{p}_G^d[k]).$$

At this point, one could put polyhedral bounds on $\Delta\boldsymbol{z}$ or $\Delta\boldsymbol{p}$ and solve a (bigger) conservative linearized system. This is *e.g.* the approach followed in [3], [6] where COM trajectories were boxed into user-defined volumes. However, contrary to these previous works the problem at hand here is on *residuals*. Intuitively, if $\|\Delta\boldsymbol{p}\|$ and $\|\Delta\boldsymbol{z}\|$ are both small, then $\|\Delta\boldsymbol{p} \times \Delta\boldsymbol{z}\|$ should be orders of magnitude smaller than the linear term $\|\overrightarrow{G^dV_i} \times \Delta\boldsymbol{z} + \overrightarrow{V_iZ^d} \times \Delta\boldsymbol{p}\|$. We therefore chose to neglect this residual cross-product term, resulting in a linear COP constraint:

$$\boldsymbol{h}_P[k]^T \Delta\boldsymbol{p}[k] + \boldsymbol{h}_Z[k]^T \Delta\boldsymbol{z}[k] \leq \sigma[k] \quad (29)$$

After implementing the whole pipeline, we checked down the line the validity of this assumption and found that, in the simulation framework described in the next section (which includes noise and delays in both control and state estimation) the ratio

$$\frac{\|\Delta\boldsymbol{p} \times \Delta\boldsymbol{z}\|}{\|\overrightarrow{G^dV_i} \times \Delta\boldsymbol{z} + \overrightarrow{V_iZ^d} \times \Delta\boldsymbol{p}\|}$$

is equal on average to 0.005 with a standard deviation of 0.005 with a sample size of 10,000 corresponding to five minutes of locomotion. That is, the cross-product term is roughly two orders of magnitude smaller than the linear one, which *a posteriori* legitimates our assumption.

Coming back to problem formulation, our constrained LQR is finally cast as a quadratic program with cost function:

$$\underset{\{\Delta\boldsymbol{z}[k]\}}{\text{minimize}} \; \sum_{k=0}^{M-1} \left(w_{xc} \|\Delta\boldsymbol{x}[k]\|^2 + w_z \|\Delta\boldsymbol{z}[k]\|^2 \right) + w_{xt} \|\Delta\boldsymbol{x}[M]\|^2 \quad (30)$$

$$\text{subject to } \forall k, (23) \wedge (26) \wedge (29) \quad (31)$$

We solve it using a classical single-shooting formulation, see *e.g.* [13], [3] for details.[4] In experiments, we set the terminal weight to $w_{xt} = 1$ and the cumulative weights to $w_{xc} = w_z = 10^{-3}$.

[4] `https://github.com/vsamy/preview_controller` provides an open-source C++/Python implementation of this method.

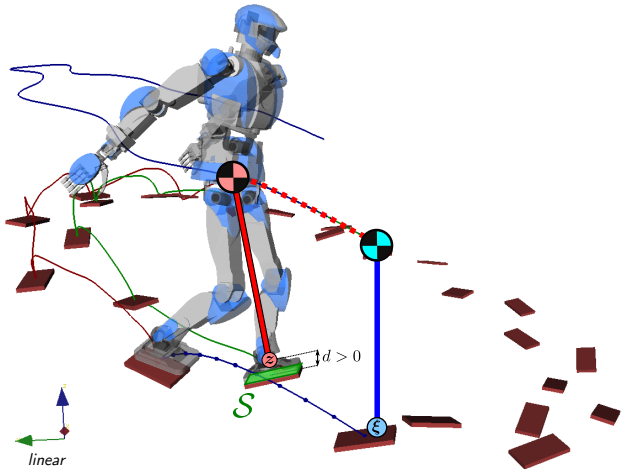

Fig. 3. Walking pattern generation over an elliptic staircase with tilted steps. At each control cycle, a new trajectory (dotted line) is computed via nonlinear optimization for the Floating-base Inverted Pendulum (red: current state, blue: desired state at the end of the preview window). In this model, the ZMP $\boldsymbol{z}$ can leave the surface patch $\mathcal{S}$ and the COM can move freely in 3D while keeping linear equations of motion.

V. Simulations

We validated the proposed method in simulation with a model of the HRP-4 humanoid robot. Our benchmark test is a randomly-generated elliptic staircase that includes all the characteristics that we deem important of rough-terrain locomotion: going up, forward and down using tilted contacts (no two contacts are coplanar).

Our simulations use *pymanoid*[5], an extension of OpenRAVE for humanoid robotics. Compared to the results reported in [3], these new simulations model both noise and delay in ZMP control and COM state estimation:

- **State estimation:** zero-mean noise with amplitudes of $10\,\text{cm}\,\text{s}^{-1}$ on position and $10\,\text{cm}\,\text{s}^{-2}$ on velocity. Nominal delay is set to $50\,\text{ms}$.
- **ZMP control:** zero-mean noise with amplitude $10\,\text{mm}\,\text{ms}^{-1}$. Control delay, *i.e.* the characteristic duration before a new command is achieved, is set to $20\,\text{ms}$.

Pattern generation is supervised by a finite state machine that alternates single and double support phases. In double support, the conservative multi-contact controller from [3] is used with the current step target as terminal condition. When the NMPC (Section III) running in the background finds a trajectory traversing the next step, the state machine switches to the next swing phase (single support).

In single support, the multi-contact controller is replaced by the LQR from Section IV. When the NMPC successfully finds a new preview, usually with a delay m between 0 and 3 control cycles, this preview is resampled and sent to the LQR as a new reference. The LQR then produces a new preview trajectory which is sent to the foot ZMP controller. The numbers of NMPC and LQR steps are set respectively to $N = 10$ and $M = 30$. With this design, our

[5] `https://github.com/stephane-caron/pymanoid`

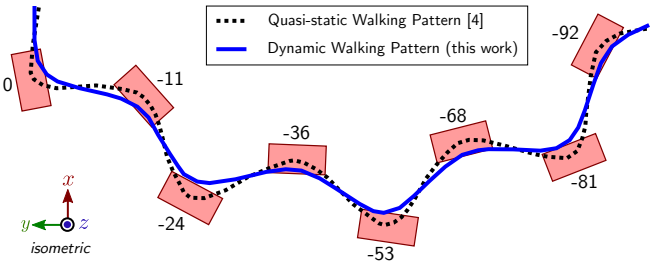

Fig. 4. Difference in COM trajectories between this work (blue line) and our previous multi-contact walking pattern generator [3] (blash dotted line). Note that the perspective is *isometric*, not linear. Footholds correspond to the downward part of the elliptic staircase depicted in Figure 3. Numbers next to them indicate their z-coordinate in cm. The new trajectory is dynamic as the COM goes only marginally over the edges of the footholds, as opposed to the quasi-static one where it nears the vertical of the foot center at each step.

pattern generator is able to locomote the humanoid accross the elliptic staircase depicted in Figure 3. However, when disabling the linear-quadratic regulator, the robot only walks a couple of steps before the NMPC becomes numerically unstable. Sample outcomes are shown in the accompanying videos. For a first-hand experience, we encourage the reader to try out the accompanying source code that we release at [25].

One important aspect in these simulations is to perform an independent check of contact-wrench feasibility at every time step. Indeed, as mentioned in Section III, constraints are only enforced at collocation points. Optimal solutions may then violate constraints in between these points, and additional validation is needed to make sure that this does not happen. This point is particularly critical in our NMPC where we use a small number of variable-duration steps, and all the more justifies the addition of an LQR with finer discretization.

TABLE I

PERFORMANCE OF THE NMPC AND LQR CONTROLLERS OVER TWO FULL CYCLES ON THE ELLIPTIC STAIRCASE.

Function	# Calls	# Successes	Time (ms)
Build NMPC	115	115	25 ± 8.5
Solve NMPC	2000	1452	21 ± 11
Build LQR	1975	1975	1.9 ± 0.2
Solve LQR	1975	1975	1.0 ± 0.4

Table I reports computation times for both the NMPC and LQR controllers over two full cycles on the elliptic staircase. Building NMPC problems only occurs around contact switches, the same nonlinear problem structure being otherwise re-used between control cycles. In this scenario, the robot called the double-support controller roughly once every two steps to handle the extra control cycles needed by the NMPC to complete its computations.

VI. CONCLUSION

We presented a real-time rough-terrain walking pattern generator that is able to adjust its step timings automatically. Our solution rests upon the floating-base inverted pendulum, a model with linear equations of motion and where contact stability can be checked using simple geometric constructions. We developed a nonlinear predictive controller that computes feedforward walking trajectories at roughly 30 Hz, as well as a constrained linear-quadratic regulator computing feedback controls one order of magnitude faster. The source code to reproduce this work is released at [25].

REFERENCES

[1] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, J. Sola, E. Yoshida, N. Mansard, K. Yokoi, and J.-P. Laumond, "Reactive stepping to prevent falling for humanoids," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*. IEEE, 2009, pp. 528–534.

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

[3] S. Caron and A. Kheddar, "Multi-contact walking pattern generation based on model preview control of 3d com accelerations," in *Humanoid Robots, 2016 IEEE-RAS International Conference on*, November 2016.

[4] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.

[5] J. Carpentier, S. TonneauG, M. Naveau, O. Stasse, and N. Mansard, "A versatile and efficient pattern generator for generalized legged locomotion," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3555–3561.

[6] H. Dai and R. Tedrake, "Planning robust walking motion on uneven terrain via convex optimization," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 579–586.

[7] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots*, 2016.

[8] 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. IEEE, 2003, pp. 1620–1626.

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

[10] 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.

[11] S. Caron, Q.-C. Pham, and Y. Nakamura, "Zmp support areas for multi-contact mobility under frictional constraints," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 67–80, Feb 2017.

[12] H. Hirukawa, S. Hattori, S. Kajita, K. Harada, K. Kaneko, F. Kanehiro, M. Morisawa, and S. Nakaoka, "A pattern generator of humanoid robots walking on a rough terrain," in *IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2181–2187.

[13] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, "Model preview control in multi-contact motion-application to a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4030–4035.

[14] Y. Zhao and L. Sentis, "A three dimensional foot placement planner for locomotion in very rough terrains," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2012, pp. 726–733.

[15] 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.

[16] 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.

[17] D. Serra, C. Brasseur, A. Sherikov, D. Dimitrov, and P.-B. Wieber, "A newton method with always feasible iterates for nonlinear model predictive control of walking in a multi-contact situation," in *Humanoid*

Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on. IEEE, 2016, pp. 932–937.

[18] K. Van Heerden, "Real-time variable center of mass height trajectory planning for humanoids robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 135–142, 2017.

[19] T. Koolen, M. Posa, and R. Tedrake, "Balance control using center of mass height variation: limitations imposed by unilateral contact," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016.

[20] S. Caron, Q.-C. Pham, and Y. Nakamura, "Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas," in *IEEE International Conference on Robotics and Automation*. IEEE, 2015.

[21] 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.

[22] L. Lanari and S. Hutchinson, "Planning desired center of mass and zero moment point trajectories for bipedal locomotion," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, pp. 637–642.

[23] S. Caron and Q.-C. Pham, "When to make a step? tackling the timing problem in multi-contact locomotion by topp-mpc," preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01363757

[24] Q.-C. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *IEEE Transactions on Robotics*, vol. 30, pp. 1533–1540, 2014.

[25] [Online]. Available: https://github.com/stephane-caron/dynamic_walking

[26] J. Andersson, "A General-Purpose Software Framework for Dynamic Optimization," PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.

[27] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

APPENDIX

A. Proof of Proposition 1

In this Appendix, all coordinates are taken with respect to the local contact frame. The analytical formula of the contact wrench cone at the origin O of this frame is given by [20]:

$$|f^x| \leq \mu f^z,\ |f^y| \leq \mu f^z \tag{32}$$

$$|\tau_O^x| \leq Y f^z,\ |\tau_O^y| \leq X f^z \tag{33}$$

$$\tau_{\mathsf{min}} \leq \tau_O^z \leq\ \tau_{\mathsf{max}} \tag{34}$$

where the two bounds on the yaw torque are defined as:

$$\tau_{\mathsf{min}} \stackrel{\text{def}}{=} -\mu(X+Y)f^z + |Y f^x - \mu\tau_O^x| + |X f^y - \mu\tau_O^y|$$
$$\tau_{\mathsf{max}} \stackrel{\text{def}}{=} +\mu(X+Y)f^z - |Y f^x + \mu\tau_O^x| - |X f^y + \mu\tau_O^y|$$

In the pendulum mode, the contact wrench is equal to:

$$\begin{array}{rclcrcl} f^x &=& \lambda(x_G - x_C) & & \tau_O^x &=& y_C f^z \\ f^y &=& \lambda(y_G - y_C) & & \tau_O^y &=& -x_C f^z \\ f^z &=& \lambda z_G & & \tau_O^z &=& x_C f^y - y_C f^x \end{array}$$

Injecting these equations into (32)–(34) yields two sets of equations (we used a symbolic calculator to avoid painstaking hand calculations here). First,

$$|x_C| \leq X,\ |y_C| \leq Y \tag{35}$$

$$|x_G - x_C| \leq \mu z_G,\ |y_G - y_C| \leq \mu z_G \tag{36}$$

And second, after rearranging all terms suitably:

$$0 \leq (X + x_C)(\mu z_G - (y_C - y_G)) + (Y + y_C)(\mu z_G + (x_C - x_G))$$
$$0 \leq (X + x_C)(\mu z_G + (y_C - y_G)) + (Y + y_C)(\mu z_G - (x_C - x_G))$$
$$0 \leq (X - x_C)(\mu z_G - (y_C - y_G)) + (Y + y_C)(\mu z_G - (x_C - x_G))$$
$$0 \leq (X - x_C)(\mu z_G + (y_C - y_G)) + (Y + y_C)(\mu z_G + (x_C - x_G))$$
$$0 \leq (X - x_C)(\mu z_G + (y_C - y_G)) + (Y - y_C)(\mu z_G - (x_C - x_G))$$
$$0 \leq (X + x_C)(\mu z_G - (y_C - y_G)) + (Y - y_C)(\mu z_G - (x_C - x_G))$$
$$0 \leq (X + x_C)(\mu z_G + (y_C - y_G)) + (Y - y_C)(\mu z_G + (x_C - x_G))$$
$$0 \leq (X - x_C)(\mu z_G - (y_C - y_G)) + (Y - y_C)(\mu z_G + (x_C - x_G))$$

All right-hand side terms in this second set can be writen as $ab + cd$, where a, b, c, d are positive slackness variables from the first set of inequalities (35)–(36). Therefore, all constraints in the second set are redundant, and the contact wrench cone in irreducible form is given by (35)–(36). We conclude by noting how (35) corresponds to $\boldsymbol{p}_C \in \mathcal{S}$ while (36) represents $\boldsymbol{p}_G \in \boldsymbol{p}_C + \mathcal{C}$.

B. Contact stability with attractors and repulsors

Let $\mathbf{A}_O$ denote the inequality matrix of the contact wrench cone taken with respect to a fixed point O. In the pendulum mode, contact stability can be written [3] in terms of the position and acceleration of the COM as

$$(\boldsymbol{a} + \boldsymbol{a}_O \times \boldsymbol{p}_G) \cdot (\ddot{\boldsymbol{p}}_G - \boldsymbol{g})\ \leq\ 0 \tag{37}$$

over all rows $(\boldsymbol{a}, \boldsymbol{a}_O)$ of the inequality matrix $\mathbf{A}_O$. These expressions are bilinear and not positive-semidefinite in general, which precludes their direct use with *e.g.* convex optimization. There is however one interesting setting where these inequalities linearize without loss of generality:

Proposition 3: If the COM control law follows a proportional attractor or repulsor H with stiffness $k \in \mathbb{R}$, that is

$$\ddot{\boldsymbol{p}}_G = k(\boldsymbol{p}_H - \boldsymbol{p}_G), \tag{38}$$

then the set of contact-stable positions $\boldsymbol{p}_G$ is a polyhedral cone rooted at the apex $\boldsymbol{\nu} := \boldsymbol{p}_H - \boldsymbol{g}/k$.

Proof: Injecting the control law (38) into (37) yields:

$$k(\boldsymbol{a} + \boldsymbol{a}_O \times \boldsymbol{p}_G) \cdot (\boldsymbol{p}_H - \boldsymbol{p}_G - \boldsymbol{g}/k)\ \leq\ 0 \tag{39}$$

Defining $\boldsymbol{\nu} := \boldsymbol{p}_H - \boldsymbol{g}/k$, this expression expands to:

$$-k(\boldsymbol{a} + \boldsymbol{a}_O \times (\boldsymbol{p}_G - \boldsymbol{\nu} + \boldsymbol{\nu})) \cdot (\boldsymbol{p}_G - \boldsymbol{\nu}) \leq 0 \tag{40}$$
$$-k(\boldsymbol{a} + \boldsymbol{a}_O \times \boldsymbol{\nu}) \cdot (\boldsymbol{p}_G - \boldsymbol{\nu}) \leq 0 \tag{41}$$

using the fact that $(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{b} = 0$ (the scalar triple product is a Gram determinant). We recognize the expression of a linear polyhedral cone in $\boldsymbol{p}_G$ with apex $\boldsymbol{\nu}$. ■

This property applies to the following two cases:

- **Repulsor:** $k = -\omega^2 < 0$ and H is the Virtual Repellent Point defined by Englsberger et al. [15]. Then, Equation (41) defines the cone $\mathcal{C}_{\mathsf{VRP}}(H, \omega^2)$ of sustainable COM positions when the VRP is located at $\boldsymbol{p}_H$.
- **Marginal attractor:** $k > 0$ and $\boldsymbol{p}_H = \boldsymbol{p}_G^d$ is a desired COM location. In this case, Equation (41) defines the cone of COM positions that can be steered toward $\boldsymbol{p}_G^d$ for a given stiffness k.

Stabilizing the COM around a reference position $\boldsymbol{p}_G^d$ requires variable VRPs in the first approach and variable stiffness with the second one.