DOI: 10.48011/asba.v2i1.1029

Biped Walking Control Based on Quadratic Programming and Differential Inequalities [1]

Stella Diniz Urban * **Bruno Vilhena Adorno** **

* *Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil (e-mail: stelladurban@gmail.com)*
** *Department of Electrical Engineering, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil (e-mail: adorno@ufmg.br)*

Abstract :
This paper presents a novel method to control a quasi-static bipedal walking based on quadratic programming and differential inequalities using geometric primitives. We allow the center of mass to move anywhere inside the support polygon during the walking cycle, as opposed to classic methods, which usually rely on tracking a desired trajectory for the zero moment point. The constraints keep the robot balance, the pelvis above a minimum height, and prevent the violation of joint limits during the complete walking cycle. Simulation results using the legs of the Poppy humanoid robot show that the trajectories of the closed-loop system converge to the desired center of mass position during the double support phase and the swing foot's trajectories converge to the desired pose during the single support phase while all constraints are obeyed.

Resumo :
Este artigo apresenta um novo método para controlar uma caminhada bípede quasi-estática baseado em programação quadrática e em desigualdades diferenciais usando primitivas geométricas. Permite-se que o centro de massa se mova para qualquer lugar dentro do polígono de suporte durante o ciclo de caminhada, diferentemente dos métodos clássicos, que geralmente dependem do rastreamento de uma trajetória desejada para o ponto de momento zero. As restrições também mantêm o equilíbrio do robô, a pelvis acima de uma altura mínima e evitam a violação dos limites das juntas durante o ciclo completo da caminhada. Os resultados da simulação, usando as pernas do robô humanoid Poppy, mostram que as trajetórias do sistema em malha fechada convergem para a posição desejada do centro de massa durante a fase de suporte duplo e as trajetórias do pé de balanço convergem para a pose desejada durante a fase de suporte simples, enquanto todas as restrições foram respeitadas.

Keywords:
Bipedal walk; constraints; zero moment point; joint limits.

Palavras-chaves:
Caminhada bípede; restrições; ponto de momento zero; limite de juntas.

1. INTRODUCTION

Humanoid robots have been widely studied in the last thirty years (Yoshida (2018)). Among their advantages are their mobility and their great number of degrees of freedom, since it allows the definition of more complex tasks. These robots can perform a wider variety of tasks than the mobile-base terrestrial robots, such as climbing stairs (Ching-Long Shih (1999)).

The walking advantage comes along with a challenge: the walking task is successful if and only if the robot's swinging foot has reached its desired pose and the whole mechanism has not fallen on the floor. The human-like walking, an example of successful biped walking in nature, has been studied over the years, from its biomechanical system and musculoskeletal structure to the walking and running trajectory data (Kim et al. (2007)).

To address the biped locomotion stability problem, Vukobratovic and Juricic (1969) propose a gait synthesis using a preliminary idea of the concept of Zero Moment Point (ZMP). Vukobratović and Stokć (1975) formalize the definition of ZMP as the point where all active forces can be reduced to a single resultant as well as the point where the horizontal components of the resultant moment are zero. Many works have based their gait synthesis on the ZMP because it is an important indicator of the dynamic balance of the biped (Vukobratović et al. (2006)). Basically, the control law must keep the ZMP inside the support polygon (SP), which is the convex hull of all parts of the robot in contact with the ground, during the gait.

[1] This work was supported by the Brazilian agencies CAPES, CNPq, and FAPEMIG.

For instance, Kajita and Tani (1991) model the biped as a linear inverted pendulum, control the robot to walk in rugged terrains, and uses the ZMP information to calculate feet dimensions to guarantee the robot balance.

Kajita et al. (2003) introduce a walking pattern generation using a ZMP trajectory generated by preview control to control the projection of the (CoM) onto the floor. That work produces an optimized trajectory for the CoM, which is overly restrictive because, in order to keep the balance, it suffices to maintain the CoM inside the SP. The SP is usually much larger than a single trajectory, which means that even if the CoM projection onto the ground is outside the ZMP trajectory, the robot is still balanced as long as that projection is inside the SP. Later, Herdt et al. (2010) extends the work of Kajita et al. (2003) by modifying the predictive controller to automatically calculate the future footsteps and additionally imposes constraints to guarantee factibility and stability.

1.1 Statement of Contributions

This paper proposes a controller of the quasi-static bipedal gait using quadratic programming with differential inequality constraints (Marinho et al. (2019)). The control inputs to perform the walking task are generated while obeying all the robot balance constraints. To accomplish that, the CoM is always kept inside the SP, which is modeled in the double support phase (DSP) by using delimiting planes and in the single support phase (SSP) by using an infinite cylinder. During all walking phases, the pelvis is also constrained above a minimum height to prevent the robot from squatting. Also, the joints angle limits are taken into consideration explicitly inside the control law to match the actual robot physical specifications, thus ensuring a feasible control input. The differential inequality constraints are defined according to the Grönwall's lemma (Gronwall (1919)), which bounds the solution to those differential inequalities to decreasing exponential functions and guarantees a smooth behavior when approaching the borders of the constraint. Compared to our previous preliminary work (Urban and Adorno (2019)), which was presented as a two-page *extended abstract*, the current work takes into account the constraints to enforce the joint limits and the pelvis minimum height; is simulated on V-REP, which is used for visualization of the robot walking; and includes all technical details to ensure reproducibility of our proposed techniques.

2. MATHEMATICAL BACKGROUND

Our method uses largely dual quaternion algebra, which offers several advantages over other mathematical frameworks (Adorno (2017)). Since dual quaternions are used to represent rigid motions and geometrical primitives such as Plücker lines, planes, and cylinders, we take advantage of dual quaternion algebra during robot modeling and control, as well as to impose the constraints related to the biped walking task since it simplifies the representation of the primitives and the calculation of the distance Jacobians between such primitives. As a result, we obtain a straightforward and compact framework for the quasi-static biped walking task.

2.1 Quaternions and Dual Quaternions

The set of quaternions is defined as

$$\mathbb{H} \triangleq \left\{ h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k} : h_1, h_2, h_3, h_4 \in \mathbb{R} \right\},$$

where $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ are imaginary units such that $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1$. The dual quaternion set is defined as $\mathcal{H} \triangleq \left\{ \boldsymbol{h} + \varepsilon \boldsymbol{h}' : \boldsymbol{h}, \boldsymbol{h}' \in \mathbb{H}, \varepsilon^2 = 0, \varepsilon \neq 0 \right\}$, where ε is the dual unit (Adorno (2017)). Given a dual quaternion $\underline{\boldsymbol{h}} = h_1 + h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k} + \varepsilon\left(h_1^{'} + h_2^{'}\hat{\imath} + h_3^{'}\hat{\jmath} + h_4^{'}\hat{k}\right)$, we define the real part as $\mathrm{Re}\left(\underline{\boldsymbol{h}}\right) \triangleq h_1 + \varepsilon h_1'$, the imaginary part as $\mathrm{Im}\left(\underline{\boldsymbol{h}}\right) \triangleq h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k} + \varepsilon\left(h_2^{'}\hat{\imath} + h_3^{'}\hat{\jmath} + h_4^{'}\hat{k}\right)$, and the conjugate as $\underline{\boldsymbol{h}}^* \triangleq \mathrm{Re}\left(\underline{\boldsymbol{h}}\right) - \mathrm{Im}\left(\underline{\boldsymbol{h}}\right)$.

Dual quaternions with unit norm, called unit dual quaternions, satisfy $\underline{\boldsymbol{h}}\underline{\boldsymbol{h}}^* = \underline{\boldsymbol{h}}^*\underline{\boldsymbol{h}} = 1$ and represent rigid motions composed of rotation and translation. The position is given by a pure quaternion, which is defined as a quaternion with real part equal to zero; for instance, $\boldsymbol{p} = p_1\hat{\imath} + p_2\hat{\jmath} + p_3\hat{k}$. The orientation is represented by a quaternion $\boldsymbol{r}$ with unit norm and the pose is given by

$$\underline{\boldsymbol{x}} = \boldsymbol{r} + \varepsilon\frac{1}{2}\boldsymbol{p}\boldsymbol{r}. \tag{1}$$

A sequence of rigid motions is given by a sequence of dual quaternion multiplications. For instance, consider the unit dual quaternions $\underline{\boldsymbol{x}}_1^0$ and $\underline{\boldsymbol{x}}_2^1$ that represent the rigid motions (translation plus rotation) from frame $\mathcal{F}_0$ to frame $\mathcal{F}_1$ and from $\mathcal{F}_1$ to $\mathcal{F}_2$, respectively. The rigid motion from $\mathcal{F}_0$ to $\mathcal{F}_2$ is given by $\underline{\boldsymbol{x}}_2^0 = \underline{\boldsymbol{x}}_1^0\underline{\boldsymbol{x}}_2^1$.

Given two pure quaternions $\boldsymbol{a}$ and $\boldsymbol{b}$, the inner product $\langle \boldsymbol{a}, \boldsymbol{b} \rangle \triangleq -\left(\boldsymbol{a}\boldsymbol{b} + \boldsymbol{b}\boldsymbol{a}\right)/2$ and the cross product $\boldsymbol{a} \times \boldsymbol{b} \triangleq \left(\boldsymbol{a}\boldsymbol{b} - \boldsymbol{b}\boldsymbol{a}\right)/2$ have the same geometrical interpretation of the inner and cross products of vectors in $\mathbb{R}^3$. Also, the pure quaternion $\boldsymbol{h} = h_2\hat{\imath} + h_3\hat{\jmath} + h_4\hat{k}$ can be mapped to $\mathbb{R}^3$ as $\mathrm{vec}_3\,\boldsymbol{h} = \begin{bmatrix} h_2 & h_3 & h_4 \end{bmatrix}^T$ and the dual quaternion $\underline{\boldsymbol{h}}$ can be mapped to $\mathbb{R}^8$ as $\mathrm{vec}_8\,\underline{\boldsymbol{h}} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & h_1' & h_2' & h_3' & h_4' \end{bmatrix}^T$.

2.2 Differential Kinematics

Consider a serial kinematic chain, whose end-effector pose is given by the unit dual quaternion in (1). The differential forward kinematics (DFK) maps the joints velocities to the end-effector (generalized) velocities, and is given by $\mathrm{vec}_8\,\dot{\underline{\boldsymbol{x}}} = \boldsymbol{J}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}}$, where $\boldsymbol{J}\left(\boldsymbol{q}\right) \in \mathbb{R}^{8\times n}$ is the Jacobian matrix and $\boldsymbol{q} \in \mathbb{R}^n$ is the vector of robot configurations. From the DFK and (1), it is possible to find the Jacobians that satisfy $\mathrm{vec}_3\,\dot{\boldsymbol{p}} = \boldsymbol{J}_p\dot{\boldsymbol{q}}$ and $\mathrm{vec}_4\,\dot{\boldsymbol{r}} = \boldsymbol{J}_r\dot{\boldsymbol{q}}$ (Adorno et al. (2010)).

3. BIPEDAL WALKING MODELING

In the quasi-static bipedal walking, we assume low velocities and small accelerations, which implies from the equations of Kajita et al. (2003) that the ZMP can be approximated by the CoM projection onto the ground.

The bipedal walking is divided into two phases: the single support phase (SSP) and the double support phase (DSP), which are alternated during the walking cycle and represent a change in the SP model and in control dynamics.

Fig. 1 shows the complete walking cycle and how the phases alternate. The walking cycle is divided in four phases, as shown in Fig. 1. In phase 1, the robot is in DSP and the CoM projection is moved to the convex hull of the left foot. In phase 2, the robot is in SSP and the right foot swings onwards to perform the first footstep. In phase 3, the robot is again in DSP and the CoM is transferred from the left to the right foot. In phase 4 , the left foot swings from the back to the front towards the desired location, while the CoM is inside the right foot SP. Phases 1 to 4 repeat until the end of the walk.

3.1 Double Support Phase

The DSP is the starting phase, where the robot has both feet in contact with the floor and the CoM projection is transferred to the new support foot. Fig 2 shows both feet area in *grey* and the convex hull in *light green*, which are combined into the SP during the DSP, as shown by the area delimited by the red line. The SP can be modeled by as many planes as necessary to represent it more faithfully. However we simplify it by selecting two planes: the tip plane $\underline{\boldsymbol{\pi}}_{\text{tip}}$ and the back plane $\underline{\boldsymbol{\pi}}_{\text{back}}$. The two additional planes in the outer borders that would be needed to complete the rectangle enclosing the feet are not necessary because the constrained motion controller, described in Section 4, ensures closed-loop exponential stability. Therefore, when the CoM projection is transferred from one foot to the other, it never goes toward the outer border of the first foot. Moreover, as we enforce a first-order closed-loop error dynamics, there is no overshoot, hence the CoM projection does not reach the outer border of the second foot either.

A plane $\underline{\boldsymbol{\pi}}$ is completely represented, within dual quaternion algebra, by its normal $\boldsymbol{n}$ and its distance d to a reference frame such that $\underline{\boldsymbol{\pi}} = \boldsymbol{n} + \epsilon d$ (Adorno (2017)).

3.2 Single Support Phase

In the SSP, the robot foot swings from the back to the front to reach the desired location. The SP is composed of the sole of the support foot. We model the SP as the intersection of an infinite vertical cylinder with the ground plane, as shown in Fig. 3. Although more conservative than the original SP, which is given by the support foot's convex hull, this new SP is very fast to compute within the constrained control law presented in Section 4. An infinite cylinder is completely defined by the center line $\underline{\boldsymbol{l}}$ and the cylinder radius r_{CoM}. Given a pure quaternion $\boldsymbol{l}$ that defines the line direction and an arbitrary point $\boldsymbol{p}$ on the line, $\underline{\boldsymbol{l}} \triangleq \boldsymbol{l} + \varepsilon\,(\boldsymbol{p} \times \boldsymbol{l})$ (Adorno (2017)). Therefore, the infinite cylinder is defined as the pair $(\underline{\boldsymbol{l}}_{\text{CoM}}, r_{\text{CoM}})$.

3.3 Additional constraints

In addition to the aforementioned constraints, which are enforced to maintain a stable gait, other constraints also play an important role to enhance the overall behavior. For instance, Fig. 2 shows the circular region formed by the intersection between a vertical cylinder and the ground plane $\underline{\boldsymbol{\pi}}_{\text{ground}}$. During the DSP, when the base frame ($\mathcal{F}_0$) is on the left foot, the right foot tip is allowed to slide inside that circular region, giving more freedom to the robot gait. Conversely, when the base frame is on the right foot, a similar region is defined for the left foot. Also, we enforce the feet to stay on the ground plane at all times during the DSP. These constraints are necessary because we control only the CoM, which means the end foot can rise above the floor and move around to perform the CoM task. Although the foot could be constrained to a point, constraining it into the circular region provides a much less conservative solution. but we wanted to be more flexible regarding the end foot.

Moreover, a horizontal plane (see Fig. 4) is also used to keep the pelvis above a minimum height, d_{pelvis}, during the whole walking cycle.

Last, the angles of the robot joints are constrained during the complete walking cycle to respect their mechanical limits, such that the minimum and maximum values for the joints are $\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max} \in \mathbb{R}^n$.

3.4 Bipedal Model

In order to perform a quasi-static gait, in which the projection of the CoM onto the ground must be always kept inside the SP, only the kinematic model is necessary. To that aim, we consider the mechanism composed of the two legs. First, we define the support foot as the first link in the kinematic chain, and then attach the first link of the swing leg to the last link of the support leg in a process called *serialization* (Adorno (2011)), as shown on the *left* of Fig. 5. For instance, let the left leg be the support leg and its forward kinematics with respect to $\mathcal{F}_0$ be given by $\underline{\boldsymbol{x}}_{\text{left}} \triangleq \underline{\boldsymbol{x}}_{\text{left}}(\boldsymbol{q}_{\text{left}})$, as shown in Fig. 5, where $\boldsymbol{q}_{\text{left}} \in \mathbb{R}^n$ is the configuration vector of the left leg. Analogously, the forward kinematics of the right leg with respect to $\mathcal{F}_6$ is given by $\underline{\boldsymbol{x}}_{\text{right}} \triangleq \underline{\boldsymbol{x}}_{\text{right}}(\boldsymbol{q}_{\text{right}})$, with $\boldsymbol{q}_{\text{right}} \in \mathbb{R}^n$. The pose of the swing foot with respect to the support foot is given by $\underline{\boldsymbol{x}}_{\text{swing},r} = \underline{\boldsymbol{x}}^*_{\text{left}}\underline{\boldsymbol{x}}^5_6\underline{\boldsymbol{x}}_{\text{right}}$ and the corresponding differential forward kinematics of the whole chain is given by $\text{vec}_8\,\dot{\underline{\boldsymbol{x}}}_{\text{swing},r} = \boldsymbol{J}_{\text{swing},r}\dot{\boldsymbol{q}}$, where $\boldsymbol{q}_r = \begin{bmatrix}\boldsymbol{q}^T_{\text{left}} & \boldsymbol{q}^T_{\text{right}}\end{bmatrix}^T \in \mathbb{R}^{2n}$ (Adorno (2011)) and thus $\boldsymbol{J}_{\text{swing},r} \in \mathbb{R}^{8\times 2n}$. Analogously, when the right and left feet are the support and swing feet, respectively, we obtain $\underline{\boldsymbol{x}}_{\text{swing},l} = \underline{\boldsymbol{x}}^*_{\text{right}}\underline{\boldsymbol{x}}^6_5\underline{\boldsymbol{x}}_{\text{left}}$, with $\text{vec}_8\,\dot{\underline{\boldsymbol{x}}}_{\text{swing},l} = \boldsymbol{J}_{\text{swing},l}\dot{\boldsymbol{q}}_l$, where $\boldsymbol{q}_l = \begin{bmatrix}\boldsymbol{q}^T_{\text{right}} & \boldsymbol{q}^T_{\text{left}}\end{bmatrix}^T$.

3.5 CoM Position and CoM Jacobian

To calculate the CoM, we analyze the relationship between a link CoM and the joint that moves it, which involves two relevant transformations, as shown in Fig. 5. The first is the transformation $\underline{\boldsymbol{x}}^0_i(\boldsymbol{q})$ from the reference frame $\mathcal{F}_0$ to frame $\mathcal{F}_i$ at the end of the ith link. The second is the constant transformation $\underline{\boldsymbol{x}}^i_{c_i}$ from frame $\mathcal{F}_i$ to the frame $\mathcal{F}_{c_i}$ at the CoM of the ith link. Therefore, the transformation between frames $\mathcal{F}_i$ and $\mathcal{F}_{c_i}$ is given by

$$\underline{\boldsymbol{x}}^0_{c_i} = \underline{\boldsymbol{x}}^0_i\underline{\boldsymbol{x}}^i_{c_i}. \tag{2}$$

The CoM Jacobian is obtained from the time derivative of (2) as

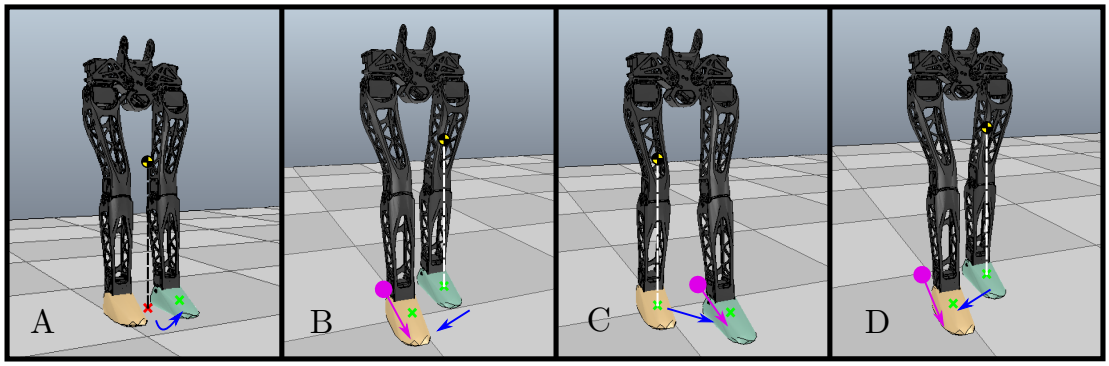

Figure 1. Walking phases: A) initial robot configuration and the first DSP. The *red* cross shows that the CoM projection starts outside the left foot convex hull, the *green* cross is the desired CoM projection at the end of the DSP, and the *blue* arrow shows this transference; B) the first SSP, where the right foot swings from the back to the front, as shown by the *pink* arrow. The *green* cross represents the CoM inside the allowable region, the blue arrow shows the next DSP transference; C) second SSP, where the left foot swings from the back to the front towards the desired pose as shown by the pink arrow, while the CoM is inside the convex hull of the right foot as shown by the green cross, the blue arrow shows the next DSP; and in D the cycle starts again.

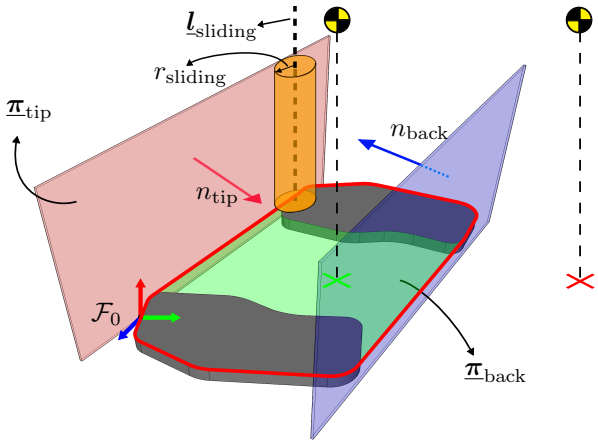

Figure 2. The DSP original SP is composed of feet area in *grey* and the area between feet in *light green*. The convex hull area is delimited by the *red* line, but we model the SP as two planes touching the tip ($\underline{\boldsymbol{\pi}}_{\text{tip}}$) in *pink* and the back ($\underline{\boldsymbol{\pi}}_{\text{back}}$) of feet in *blue*. The cylinder with radius r_{sliding}, constraining the sliding of the right foot, is on the tip of the right foot. When the CoM is inside the planes, shown by the green cross, the robot is statically stable; when outside, as shown by the red cross, the robot is not statically stable.

$$\begin{aligned}\operatorname{vec}_8 \dot{\underline{\boldsymbol{x}}}^0_{c_i} &= \bar{\boldsymbol{H}}_8\left(\underline{\boldsymbol{x}}^i_{c_i}\right)\operatorname{vec}_8 \dot{\underline{\boldsymbol{x}}}^0_i = \bar{\boldsymbol{H}}_8\left(\underline{\boldsymbol{x}}^i_{c_i}\right)\boldsymbol{J}_i\dot{\boldsymbol{q}}_i \\ &= \bar{\boldsymbol{J}}_{c_i}\dot{\boldsymbol{q}}_i = \underbrace{\left[\bar{\boldsymbol{J}}_{c_i}\ \boldsymbol{0}^{8\times n-i}\right]}_{\boldsymbol{J}_{c_i}}\dot{\boldsymbol{q}}, \end{aligned} \tag{3}$$

where we used the fact that $\dot{\underline{\boldsymbol{x}}}^i_{c_i} = 0$, $\boldsymbol{J}_i \in \mathbb{R}^{8\times i}$ and $\boldsymbol{q}_i \in \mathbb{R}^i$ are the Jacobian and the joint velocities vector up to joint i, and $\bar{\boldsymbol{H}}_8 : \mathcal{H} \to \mathbb{R}^{8\times 8}$ is an operator that satisfies $\operatorname{vec}_8\left(\underline{\boldsymbol{h}}_1\underline{\boldsymbol{h}}_2\right) = \bar{\boldsymbol{H}}_8\left(\underline{\boldsymbol{h}}_2\right)\operatorname{vec}_8 \underline{\boldsymbol{h}}_1$ (Adorno (2017)). Moreover, the CoM position of the chain composed of the two legs and the pelvis, which has n degrees of freedom, is given by the weighted sum (Oliveira and Adorno (2015))

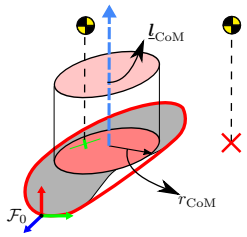

Figure 3. The SSP convex hull is the area delimited by the *red* line, but the SP is simplified to the *pink* area determined by the intersection between the infinite cylinder ($\underline{\boldsymbol{l}}_{\text{CoM}}, r_{\text{CoM}}$) and the ground plane. When the CoM is inside the SP, shown by the *green* cross, the robot is statically stable; when outside, as shown by the *red* cross, the robot is not statically stable.

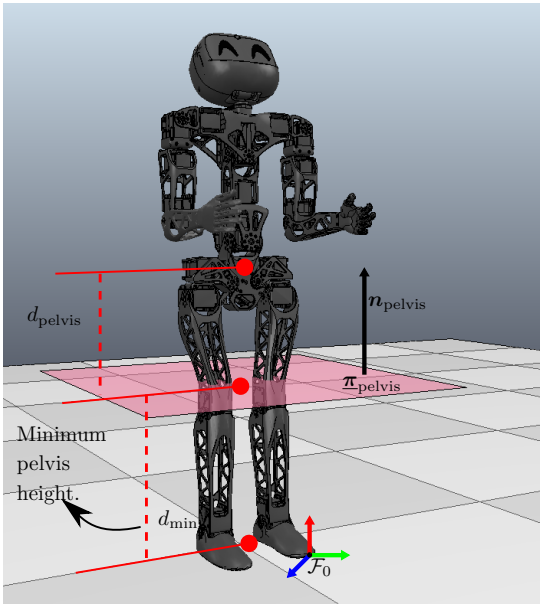

Figure 4. Pelvis plane $\underline{\boldsymbol{\pi}}_{\text{pelvis}}$ with normal $\boldsymbol{n}_{\text{pelvis}}$ and distance d_{pelvis} from the reference frame.

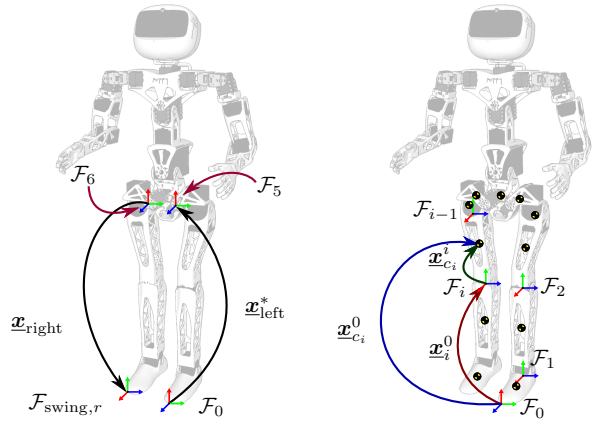

Figure 5. Biped robot: on the *left* the robot serialization, in which the first chain is the reversed left leg chain ($\underline{\boldsymbol{x}}^*_{\text{left}}$), followed by the transformation ($\underline{\boldsymbol{x}}^5_6$) from frame $\mathcal{F}_5$ to frame $\mathcal{F}_6$—attached to the bases of the left and right legs, respectively—, followed by the right leg chain ($\underline{\boldsymbol{x}}_{\text{right}}$); on the *right*, the CoM position of all links that are used to calculate the CoM of the whole kinematic chain composed of the two legs.

$$\operatorname{vec}_3 \boldsymbol{p}_{\text{CoM}} = \frac{1}{M}\sum_{i=0}^{n} m_i \operatorname{vec}_3 \boldsymbol{p}_{c_i}, \tag{4}$$

where m_i is the mass of the ith link, $\boldsymbol{p}_{c_i}$ is the translational part of $\underline{\boldsymbol{x}}^0_{c_i}$, and M is the total mass of the mechanism. The CoM of the support foot, $\boldsymbol{p}_{c_0}$, is also taken into account because it contributes to the total mass of the mechanism. Finally, obtaining the translational component $\boldsymbol{J}_{p_{ci}}$ of $\boldsymbol{J}_{c_i}$ that satisfies $\operatorname{vec}_3 \dot{\boldsymbol{p}}_{c_i} = \boldsymbol{J}_{p_{c_i}}\dot{\boldsymbol{q}}$, as described in Section 2.2, we obtain

$$\operatorname{vec}_3 \dot{\boldsymbol{p}}_{\text{CoM}} = \frac{1}{M}\sum_{i=0}^{n} m_i \operatorname{vec}_3 \dot{\boldsymbol{p}}_{c_i} = \underbrace{\left(\frac{1}{M}\sum_{i=0}^{n} m_i \boldsymbol{J}_{p_{c_i}}\right)}_{\boldsymbol{J}_{\text{CoM}}}\dot{\boldsymbol{q}}, \tag{5}$$

where $\boldsymbol{J}_{\text{CoM}} \in \mathbb{R}^{3\times n}$ is the CoM Jacobian of the complete chain.

4. CONSTRAINED CONTROLLER

To control the gait while respecting all the constraints, we use a constrained kinematic controller based on quadratic programming, where all the nonlinear geometric constraints are enforced as linear differential inequalities on the control inputs, the so-called vector field inequalities (Marinho et al. (2019)).

The control input $\boldsymbol{u}$ is given by

$$\begin{aligned} \boldsymbol{u} \in \underset{\dot{\boldsymbol{q}}}{\operatorname{argmin}} \quad & \|\boldsymbol{J}_{\text{task}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \eta\tilde{\boldsymbol{x}}_{\text{task}}\|_2^2 + \lambda^2\|\dot{\boldsymbol{q}}\|_2^2 \\ \text{subject to} \quad & \boldsymbol{W}\dot{\boldsymbol{q}} \leq \boldsymbol{w} \\ & \boldsymbol{W}_{\text{eq}}\dot{\boldsymbol{q}} = \boldsymbol{w}_{\text{eq}}, \end{aligned} \tag{6}$$

where $\tilde{\boldsymbol{x}}_{\text{task}}$ is the task-space error vector, $\boldsymbol{J}_{\text{task}}(\boldsymbol{q})$ is the task Jacobian matrix, $\boldsymbol{q} \in \mathbb{R}^n$ is the joint vector, $\eta \in (0,\infty)$ is the controller gain, and $\lambda \in (0,\infty)$ is a damping factor to ensure that the joints velocities are minimized. Furthermore, the matrix $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ and the vector $\boldsymbol{w} \in \mathbb{R}^m$ define the inequality constraints whereas $\boldsymbol{W}_{\text{eq}} \in \mathbb{R}^{l\times n}$ and $\boldsymbol{w}_{\text{eq}} \in \mathbb{R}^l$ define the equality constraints.

The main idea of the vector field inequalities, which are used in the inequality constraints, is to define a differentiable function $d : \mathbb{R}^n \to \mathbb{R}$ that provides the (signed) distance between two geometric primitives as a function of the robot joints. Therefore, given a constant reference distance d_{ref} and $\tilde{d}(\boldsymbol{q}) \triangleq d(\boldsymbol{q}) - d_{\text{ref}}$, it is possible to show that the inequality $\dot{\tilde{d}}(\boldsymbol{q}) \geq -\eta_d\tilde{d}(\boldsymbol{q})$, where $\eta_d \in (0,\infty)$, ensures that, if $\tilde{d}(\boldsymbol{q}(0)) \geq 0$ then $\tilde{d}(\boldsymbol{q}(t)) \geq e^{-\eta_d t}\tilde{d}(\boldsymbol{q}(0))$ for all $t \geq 0$. Conversely, when the inequality is reversed, that is $\dot{\tilde{d}}(\boldsymbol{q}) \leq -\eta_d\tilde{d}(\boldsymbol{q})$, if $\tilde{d}(\boldsymbol{q}(0)) \leq 0$ then $\tilde{d}(\boldsymbol{q}(t)) \leq e^{-\eta_d t}\tilde{d}(\boldsymbol{q}(0))$ for all $t \geq 0$ (Marinho et al. (2019)). Taking the time derivative from the distance function $\tilde{d}(\boldsymbol{q})$, we obtain $\dot{\tilde{d}}(\boldsymbol{q}) = \boldsymbol{J}_d\dot{\boldsymbol{q}}$, where $\boldsymbol{J}_d \triangleq \partial\tilde{d}(\boldsymbol{q})/\partial\boldsymbol{q}$, which can be used in (6).

What determines the type of differential inequality to be used—i.e., positive ($\geq$) or negative distances ($\leq$)—depends on the relationship between the geometric primitives. Given a point and a plane, a positive distance determines that the point is on the side of the plane to where its normal is pointing. For instance, in Fig. 2, both normals $\boldsymbol{n}_{\text{tip}}$ and $\boldsymbol{n}_{\text{back}}$ points to the convex hull. Therefore, the distances between the green CoM projection and the planes are both positive. On the other hand, the distance of the red CoM projection to $\underline{\boldsymbol{\pi}}_{\text{tip}}$ is positive, but the distance to $\underline{\boldsymbol{\pi}}_{\text{back}}$ is negative. Analogously, the distance between a point inside a cylinder and the cylinder boundary is negative, whereas the distance between a point outside the cylinder and the cylinder boundary is positive.

On the other hand, an equality constraint such as $\dot{\tilde{d}} = -\eta\tilde{d}$ determines that the distance between the geometric primitives will converge exponentially to zero.

Therefore, the main challenge is to define suitable task variables $\boldsymbol{J}_{\text{task}}(\boldsymbol{q})$ and $\tilde{\boldsymbol{x}}_{\text{task}}$ and the corresponding constraint variables $\boldsymbol{W}, \boldsymbol{w}, \boldsymbol{W}_{\text{eq}}, \boldsymbol{w}_{\text{eq}}$ based on the geometrical primitives and control objectives described in Section 3.

4.1 DSP controller

From Section 3.1, during the DSP, the goal is to transfer the CoM projection from one foot to the other while it is between the planes $\underline{\boldsymbol{\pi}}_{\text{tip}}$ and $\underline{\boldsymbol{\pi}}_{\text{back}}$. In addition, both feet are on the ground plane $\underline{\boldsymbol{\pi}}_{\text{ground}}$, the tip of the end foot (i.e., the one to where the CoM projection is being moved) is confined to the cylinder $(\underline{\boldsymbol{l}}_{\text{sliding}}, r_{\text{sliding}})$, the pelvis is above the plane $\underline{\boldsymbol{\pi}}_{\text{pelvis}}$, and the joint angles are limited by the joint range $[-\phi_{\text{safe},i}, \phi_{\text{safe},i}]$ for every ith joint. Using the signed distance functions and the corresponding Jacobians proposed by Marinho et al. (2019), namely the distance between points and cylinders and between points and planes, we define the following constraints:

$$\dot{\tilde{d}}_{\text{sliding}} = \boldsymbol{J}_{\text{sliding}}\dot{\boldsymbol{q}} \leq -\eta_{\text{sliding}}\tilde{d}_{\text{sliding}} = w_{\text{sliding}}, \tag{7}$$

where $\tilde{d}_{\text{sliding}}$ is the signed distance between the end-foot tip and the sliding cylinder $(\underline{\boldsymbol{l}}_{\text{sliding}}, r_{\text{sliding}})$ such that a negative distance means that the point is inside the cylinder, the gain $\eta_{\text{sliding}} \in (0,\infty)$ determines how fast the cylinder boundary can be approached from the inside, and $\mathbb{R}^{1\times n} \ni \boldsymbol{J}_{\text{sliding}} = \partial\tilde{d}_{\text{sliding}}(\boldsymbol{q})/\partial\boldsymbol{q}$;

$$-\boldsymbol{J}_{\text{ground}}\dot{\boldsymbol{q}} = \eta_{\text{ground}}\tilde{d}_{\text{ground}} = w_{\text{ground}}, \tag{8}$$

where $\tilde{d}_{\text{ground}}$ is the signed distance between the end-foot tip and the ground plane $\underline{\boldsymbol{\pi}}_{\text{ground}}$, the gain $\eta_{\text{ground}} \in (0, \infty)$ determines how fast the foot should return to the ground in case it raises above the ground plane, and $\mathbb{R}^{1\times n} \ni \boldsymbol{J}_{\text{ground}} = \partial \tilde{d}_{\text{ground}}(\boldsymbol{q})/\partial \boldsymbol{q}$;

$$\dot{\tilde{d}}_{\text{tip}} \geq -\eta_{\text{tip}} \tilde{d}_{\text{tip}} \implies -\boldsymbol{J}_{\text{tip}} \dot{\boldsymbol{q}} \leq \eta_{\text{tip}} \tilde{d}_{\text{tip}} = w_{\text{tip}}, \tag{9}$$

where $\tilde{d}_{\text{tip}}$ is the signed distance between the CoM projection and the tip plane $\underline{\boldsymbol{\pi}}_{\text{tip}}$, the gain $\eta_{\text{tip}} \in \mathbb{R}$ determines how fast the CoM projection is allowed to approach the tip plane from its positive side (i.e., the side where the plane normal points at), and $\mathbb{R}^{1\times n} \ni \boldsymbol{J}_{\text{tip}} = \partial \tilde{d}_{\text{tip}}(\boldsymbol{q})/\partial \boldsymbol{q}$;

$$\dot{\tilde{d}}_{\text{back}} \geq -\eta_{\text{back}} \tilde{d}_{\text{back}} \implies -\boldsymbol{J}_{\text{back}} \dot{\boldsymbol{q}} \leq \eta_{\text{back}} \tilde{d}_{\text{back}} = w_{\text{back}}, \tag{10}$$

where $\tilde{d}_{\text{back}}$ is the signed distance between the CoM projection and the back plane $\underline{\boldsymbol{\pi}}_{\text{back}}$, the gain $\eta_{\text{back}} \in \mathbb{R}$ determines how fast the CoM projection is allowed to approach the back plane from its positive side, and $\mathbb{R}^{1\times n} \ni \boldsymbol{J}_{\text{back}} = \partial \tilde{d}_{\text{back}}(\boldsymbol{q})/\partial \boldsymbol{q}$;

$$\dot{\tilde{d}}_{\text{pelvis}} \geq -\eta_{\text{pelvis}} \tilde{d}_{\text{pelvis}} \Longrightarrow -\boldsymbol{J}_{\text{pelvis}} \dot{\boldsymbol{q}} \leq \eta_{\text{pelvis}} \tilde{d}_{\text{pelvis}} = w_{\text{pelvis}}, \tag{11}$$

where $\tilde{d}_{\text{pelvis}}$ is the signed distance between the pelvis position and the pelvis plane $\underline{\boldsymbol{\pi}}_{\text{pelvis}}$, the $\eta_{\text{pelvis}} \in \mathbb{R}$ determines how fast the CoM is allowed to approach the pelvis plane from above, and $\mathbb{R}^{1\times n} \ni \boldsymbol{J}_{\text{tip}} = \partial \tilde{d}_{\text{tip}}(\boldsymbol{q})/\partial \boldsymbol{q}$; and, finally,

$$\dot{\boldsymbol{q}} \leq -\eta_{\phi}(\boldsymbol{q} - \boldsymbol{q}_{\max}) = \boldsymbol{w}_{\phi_{\max}}, \tag{12}$$

$$-\dot{\boldsymbol{q}} \leq \eta_{\phi}(\boldsymbol{q} - \boldsymbol{q}_{\min}) = \boldsymbol{w}_{\phi_{\min}}, \tag{13}$$

where $\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max} \in \mathbb{R}^n$ are the vectors of minimum and maximum values for the joints.

The goal during DSP is to transfer the CoM projection between the feet, which means to control only the x and y coordinates of the CoM. Therefore, in order to generate the control input (6), we define $\boldsymbol{J}_{\text{task}} \triangleq \boldsymbol{J}_{\text{CoM}_{(\text{x,y})}}$, where $\boldsymbol{J}_{\text{CoM}_{(\text{x,y})}}$ is given by the first and second rows of $\boldsymbol{J}_{\text{CoM}}$ in (5). In addition, $\tilde{\boldsymbol{x}}_{\text{task}} \triangleq \text{vec}_3(\boldsymbol{p}_{\text{CoM}_{(\text{x,y})}} - \boldsymbol{p}_{\text{CoM}d_{(x,y)}})$ with $\boldsymbol{p}_{\text{CoM}d_{(x,y)}}$ being the desired projection of the CoM onto the ground and $\boldsymbol{p}_{\text{CoM}_{(\text{x,y})}}$ being its current value, which is obtained from the first and the second rows of $\text{vec}_3\, \boldsymbol{p}_{\text{CoM}}$ in (4). Furthermore, the constraint variables are defined as

$$\begin{aligned}
\boldsymbol{W} &= \begin{bmatrix} \boldsymbol{J}_{\text{sliding}}^T & -\boldsymbol{J}_{\text{tip}}^T & -\boldsymbol{J}_{\text{back}}^T & -\boldsymbol{J}_{\text{pelvis}}^T & \boldsymbol{I}_n & -\boldsymbol{I}_n \end{bmatrix}^T, \\
\boldsymbol{w} &= \begin{bmatrix} w_{\text{sliding}} & w_{\text{tip}} & w_{\text{back}} & w_{\text{pelvis}} & \boldsymbol{w}_{\phi_{\max}}^T & \boldsymbol{w}_{\phi_{\min}}^T \end{bmatrix}^T, \\
\boldsymbol{W}_{\text{eq}} &= -\boldsymbol{J}_{\text{ground}}, \\
\boldsymbol{w}_{\text{eq}} &= w_{\text{ground}}.
\end{aligned}$$

4.2 SSP controller

During the SSP, the goal is to place the end foot to the next location while the CoM projection is inside the cylinder $(\underline{\boldsymbol{l}}_{\text{CoM}}, r_{\text{CoM}})$. As in the DSP, in the SSP, the pelvis is above the plane $\underline{\boldsymbol{\pi}}_{\text{pelvis}}$ and the ith joint angle is inside the interval $[-\phi_{\text{safe},i}, \phi_{\text{safe},i}]$ for every joint. Also, we define the constraint

$$\dot{\tilde{d}}_{\text{CoM}} = \boldsymbol{J}_{\text{CoM}} \dot{\boldsymbol{q}} \leq -\eta_{\text{CoM}} \tilde{d}_{\text{CoM}} = w_{\text{CoM}}, \tag{14}$$

where $\tilde{d}_{\text{CoM}}$ is the signed distance between the CoM projection and the support polygon cylinder $(\underline{\boldsymbol{l}}_{\text{CoM}}, r_{\text{CoM}})$ such that a negative distance means that the CoM projection is inside the support polygon, the gain $\eta_{\text{CoM}} \in (0, \infty)$ determines how fast the CoM projection can approach the SP boundary, and $\mathbb{R}^{1\times n} \ni \boldsymbol{J}_{\text{CoM}} = \partial \tilde{d}_{\text{CoM}}(\boldsymbol{q})/\partial \boldsymbol{q}$. The additional constraints are given by (11) and (12), respectively. Therefore,

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{J}_{\text{CoM}}^T & -\boldsymbol{J}_{\text{pelvis}}^T & \boldsymbol{I}_n & -\boldsymbol{I}_n \end{bmatrix}^T$$

and $\boldsymbol{w} = \begin{bmatrix} w_{\text{CoM}} & w_{\text{pelvis}} & \boldsymbol{w}_{\phi_{\max}}^T & \boldsymbol{w}_{\phi_{\min}}^T \end{bmatrix}^T$.

Since in the SSP the goal is to swing the end foot from the current position to a forward position, the control input is generated using (6), where $\boldsymbol{J}_{\text{task}} \triangleq \boldsymbol{J}_{\text{swing,r}}$ is the Jacobian matrix of the coupled kinematic chain composed of the left leg followed by the right leg, when the base frame is on the left foot, and $\tilde{\boldsymbol{x}}_{\text{task}} \triangleq \text{vec}_8\left(\underline{\boldsymbol{x}}_{\text{swing},r} - \underline{\boldsymbol{x}}_d\right)$, where $\underline{\boldsymbol{x}}_{\text{swing},r}$ is the right-foot pose and $\underline{\boldsymbol{x}}_d$ is its desired value. Analogously, when the base frame is on the right foot, $\boldsymbol{J}_{\text{task}} \triangleq \boldsymbol{J}_{\text{swing,l}}$ and $\tilde{\boldsymbol{x}}_{\text{task}} \triangleq \text{vec}_8\left(\underline{\boldsymbol{x}}_{\text{swing},l} - \underline{\boldsymbol{x}}_d\right)$, where $\boldsymbol{J}_{\text{swing,l}}$ is the Jacobian of the coupled kinematic chain composed of the right leg followed by the left leg and $\underline{\boldsymbol{x}}_{\text{swing},l}$ is the left-foot pose.

5. SIMULATION RESULTS AND DISCUSSIONS

We evaluated, in simulation, the control strategy for the complete walking cycle on the Poppy [1] humanoid legs using MATLAB software, the DQ Robotics library (Adorno and Marques Marinho (2020)), and V-REP (Rohmer et al. (2013)) for visualization. All robot joints are equal to zero in the beginning of the first walking cycle. The CoM, end foot, base foot, and pelvis start inside the allowable regions to ensure the feasibility of (6). Since the robot is small, with foot width and length equal to 7 cm and 9.34 cm, respectively, we defined a small step length during SSP. More specifically, each step has 5 cm on the horizontal saggital axis. The controller parameters in (6) are $\eta = 0.05$ and $\lambda = 0.1$. The constraints were defined as described in Sections 4.1 and 4.2, with $\eta_{\text{sliding}} = 5$, $\eta_{\text{ground}} = 5$, $\eta_{\text{pelvis}} = 500$, $\eta_{\text{tip}} = 10$, $\eta_{\text{back}} = 20$, $\eta_{\phi} = 10$, and $\eta_{\text{CoM}} = 5$.

During the DSP, the geometrical constraints were defined as $\underline{\boldsymbol{\pi}}_{\text{ground}} = \hat{k}$, $(\underline{\boldsymbol{l}}_{\text{sliding}}, r_{\text{sliding}}) = (\hat{k} + \varepsilon(\boldsymbol{p}_{\text{endtip}} \times \hat{k}), 0.04)$, where $\boldsymbol{p}_{\text{endtip}}$ is the end-foot tip position, and $\underline{\boldsymbol{\pi}}_{\text{pelvis}} = \hat{k} + \varepsilon 0.2$. The $\underline{\boldsymbol{\pi}}_{\text{tip}}$ and $\underline{\boldsymbol{\pi}}_{\text{back}}$ were defined such that they are orthogonal to the ground plane and pass through points at both the feet tips and backs, respectively. During the SSP, the geometrical constraint related to the SP was defined as $(\underline{\boldsymbol{l}}_{\text{CoM}}, r_{\text{CoM}}) = (\hat{k} + \varepsilon(\boldsymbol{p}_{\text{basecom}} \times \hat{k}), 0.05)$, where $\boldsymbol{p}_{\text{basecom}}$ is the base foot CoM.

The control objective is switched from the DSP to the SSP when the error stops decreasing and the CoM projection is within the SSP cylinder $(\underline{\boldsymbol{l}}_{\text{CoM}}, r_{\text{CoM}})$ of the end foot. Conversely, the control objective is switched from the SSP to the DSP when the error is below a threshold of 0.001.

In order to update the joints, a first-order numerical integration (Euler's method) is performed using an integration step of 0.005 s.

[1] https://www.poppy-project.org/en/robots/poppy-humanoid

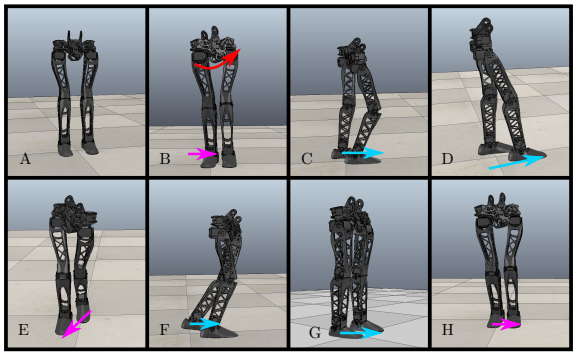

Figure 6. Complete cycle. A) Initial configuration (DSP); B) End of the DSP; C) the right foot swings towards the desired location, which is achieved in D; E) the CoM is transferred from the left SP to the right SP; F) the left foot swings toward the desired location, which is achieved in G; H), finally, the cycle starts again.

5.1 Closed-loop behavior of the complete walking cycle

Fig. 6 shows the first complete walking cycle, in which the CoM projection was always confined to the SP, ensuring a stable gait. In the first DSP, the CoM projection started inside the SP, but outside both feet, and then it was transferred from the old to the new support foot. Fig. 7A shows that this first part was completed with a steady-state error as the CoM projection could not be driven to the desired value due to the constraints. Nonetheless, the CoM projection entered into the SP of the next SSP because the steady-state error was smaller than the threshold given by the *dashed-red* line, which represents the distance between the centerline and the boundary of the next SSP cylinder.

During the first DSP, the CoM projection was kept inside the SP, which is approximated by the region between the tip plane and back plane, as indicated by the positive distances to each plane, namely $d_{\text{feet\_tip}}$ and $d_{\text{feet\_back}}$ in Figs. 7D and E. The constraint on the end-foot tip was also obeyed, as the end-foot tip stayed inside the sliding cylinder, which is shown by the curve below the threshold in Fig. 7B. The steady-state distance between the end-foot and the ground plane was approximately 0.01 mm due to discretization effects.

Following the first DSP, the first SSP starts after the error in the DSP stops decreasing. Fig. 8A shows that the swing foot (i.e. the right one) is placed at the desired location as the pose error norm decreases exponentially to zero. Moreover, the CoM was kept inside the SP during the entire SSP (Fig.8B), as well as the pelvis was above the minimum height (Fig.8B).

For the next DSP, the projection of the CoM was transferred from the left foot to the right foot and the behavior was analogous to the first behavior, as shown by the green curves in Fig. 7. The second DSP is followed by the second SSP, in which the behavior was also analogous to the first SSP, as shown by the green curves in Fig. 8.

Finally, Fig. 9 shows that the joints limits were respected during the complete walking cycle.

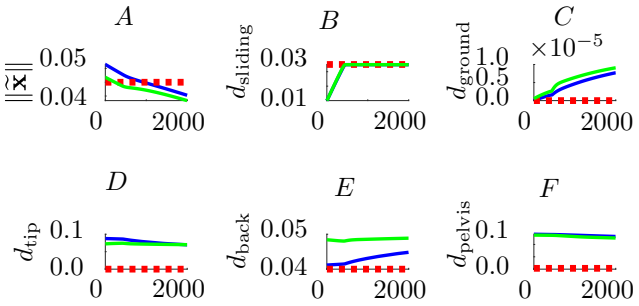

Figure 7. Time-response during the DSP for one complete cycle (first DSP in *blue* and second in *green*). A) the norm of the CoM projection error and the boundary of the next SSP in *dashed-red* line; B) end foot pose distance from sliding cylinder main axis with corresponding boundary in *dashed-red* line; *C*) foot pose distance from the floor plane; D) the distance of the CoM from the plane formed by feet tip; E) CoM distance from the plane formed by back tip; F) and distance of the pelvis to the plane.

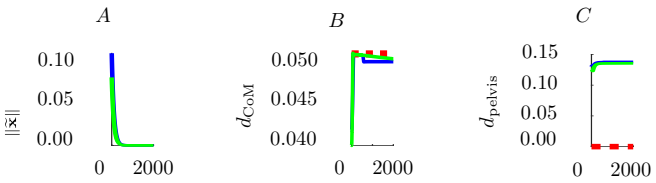

Figure 8. Time response during the first SSP (right leg, *blue curve*) and during the second SSP (the left leg, *green curve*). On the *left*, the foot pose error norm. In the *middle*, the distance of the CoM projection to the boundary of the SP, which is represented by the *dashed-red line* ($d_{\text{CoM}} = 0$ means that the CoM is on the middle of the SP). On the *right*, the distance of the pelvis to the minimum height allowed.

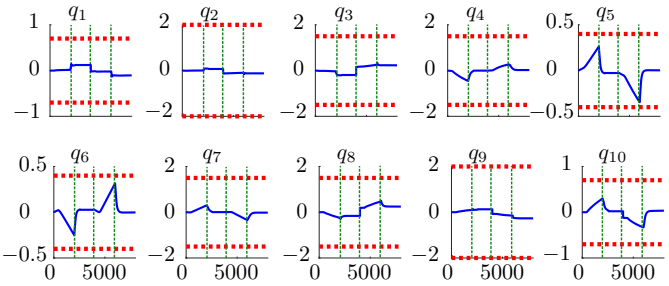

Figure 9. Joint angles during the complete gait cycle. The *vertical dashed lines* indicate the transition from DSP to SSP and from SSP to DSP. The *horizontal dashed lines* indicate the joint limits.

6. CONCLUSION

This paper has proposed a closed-loop constrained controller for the quasi-static bipedal walking based on quadratic programming and differential inequalities. The controller takes into account constraints on the joints, CoM, swing foot, and pelvis. The SP is modeled by using geometric primitives and the constraints are defined by using the differential inequalities associated to those primitives. This enables the control of the CoM, during

the DSP, and the swing foot, during the SSP, without explicitly specifying the trajectory for the CoM. In addition, differently from classic approaches in which the CoM must follow a trajectory, our technique is less conservative as the CoM is bounded by the SP, but it does not follow any predefined trajectory. As a result, we do not need to calculate a CoM trajectory and the implementation is much simpler than the one proposed by Kajita et al. (2003).

All constraints were respected during the entire walking cycle, although the discretization played a relatively negligible role in the floor plane constraint. More specifically, during the DSP, both feet were supposed to stay flat on the ground, but disturbances from the discretization of the control signal would make the base foot raise up to 0.1 mm.

One disadvantage of our method is that closed-loop *asymptotic* stability is not guaranteed due to the constraints. Therefore, if the steps are too large, it may be the case that the closed-loop system reaches a local minimum and the step is not concluded. Nonetheless, the robot would not fall in those cases because the constraints ensure that the CoM projection is always inside the SP, thus enforcing static balance. The quasi-static walking is also a disadvantage of our approach since it is slower than the other state-of-the-art works, which are based on dynamic walking. Besides, we did not study the system behavior in the presence of disturbances, as done in the work of Maximo et al. (2016). Therefore, we do not guarantee the robot balance under such circumstances. Further works will focus on implementing the controller on the Poppy humanoid robot available at the Mechatronics, Control, and Robotics group at UFMG and the extension of our technique to dynamic gait.

REFERENCES

Adorno, B. (2017). Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra – Part I: Fundamentals. URL `https://hal.archives-ouvertes.fr/hal-01478225v1`.

Adorno, B.V. (2011). *Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration [Contribution à la manipulation à deux bras : des manipulateurs à la collaboration homme-robot]*. Ph.d. dissertation, Université Montpellier 2, Montpellier, France,.

Adorno, B.V., Fraisse, P., and Druon, S. (2010). Dual position control strategies using the cooperative dual task-space framework. In *2010 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 3955–3960. IEEE. doi:10.1109/IROS.2010.5650218.

Adorno, B.V. and Marques Marinho, M. (2020). DQ Robotics: A Library for Robot Modeling and Control. *IEEE Robot. Autom. Mag.*, 0–0. doi:10.1109/MRA.2020.2997920.

Ching-Long Shih (1999). Ascending and descending stairs for a biped robot. *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, 29(3), 255–268. doi:10.1109/3468.759271.

Gronwall, T.H. (1919). Note on the Derivatives with Respect to a Parameter of the Solutions of a System of Differential Equations. *Ann. Math.*, 20(4), 292. doi:10.2307/1967124.

Herdt, A., Diedam, H., Wieber, P.B., Dimitrov, D., Mombaur, K., and Diehl, M. (2010). Online Walking Motion Generation with Automatic Footstep Placement. *Adv. Robot.*, 24(5-6), 719–737. doi:10.1163/016918610X493552.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. (2003). Biped walking pattern generation by using preview control of zero-moment point. In *2003 IEEE Int. Conf. Robot. Autom. (Cat. No.03CH37422)*, volume 2, 1620–1626. IEEE. doi:10.1109/ROBOT.2003.1241826.

Kajita, S. and Tani, K. (1991). Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode. In *Proceedings. 1991 IEEE Int. Conf. Robot. Autom.*, volume 2, 1405–1411. IEEE Comput. Soc. Press. doi:10.1109/ROBOT.1991.131811.

Kim, J.Y., Park, I.W., and Oh, J.H. (2007). Walking Control Algorithm of Biped Humanoid Robot on Uneven and Inclined Floor. *J. Intell. Robot. Syst.*, 48(4), 457–484. doi:10.1007/s10846-006-9107-8.

Marinho, M.M., Adorno, B.V., Harada, K., and Mitsuishi, M. (2019). Dynamic Active Constraints for Surgical Robots Using Vector-Field Inequalities. *IEEE Trans. Robot.*, 35(5), 1166–1185. doi:10.1109/TRO.2019.2920078.

Maximo, M.R.O.A., Ribeiro, C.H.C., and Afonso, R.J.M. (2016). Mixed-integer programming for automatic walking step duration. In *2016 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, volume 2016-Novem, 5399–5404. IEEE. doi:10.1109/IROS.2016.7759794.

Oliveira, A.C. and Adorno, B.V. (2015). Balance Control of a Humanoid Robot Based on the Cooperative Dual Task-space Framework. *XII Simpósio Bras. Automação Intel.*, 485–490.

Rohmer, E., Singh, S.P.N., and Freese", M. (2013). "CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework". In *"Proc. of The International Conference on Intelligent Robots and Systems (IROS)"*. Www.coppeliarobotics.com.

Urban, S.D. and Adorno, B.V. (2019). Constraint-based Gait Control Using Differential Inequalities. *Work. Appl. Dual Quaternion Algebr. to Robot.*, 1–2. doi:10.5281/zenodo.3566763.

Vukobratović, M. and Stokć, D. (1975). Postural stability of anthropomorphic systems. *Math. Biosci.*, 25(3-4), 217–236. doi:10.1016/0025-5564(75)90004-8.

Vukobratović, M., Borovac, B., and Potkonjak, V. (2006). ZMP: A REVIEW OF SOME BASIC MISUNDERSTANDINGS. *Int. J. Humanoid Robot.*, 03(02), 153–175. doi:10.1142/S0219843606000710.

Vukobratovic, M. and Juricic, D. (1969). Contribution to the Synthesis of Biped Gait. *IEEE Trans. Biomed. Eng.*, BME-16(1), 1–6. doi:10.1109/TBME.1969.4502596.

Yoshida, E. (2018). Robots that look like humans: A brief look into humanoid robotics. *Mètode Rev. difusió la Investig.*, 2019(9), 143–151. doi:10.7203/metode.9.11405.