Whole-body MPC and sensitivity analysis of a real time foot step sequencer for a biped robot Bolt

Constant Roux, Côme Perrot, Olivier Stasse

Whole-body MPC and sensitivity analysis of a real time foot step sequencer for a biped robot Bolt

Constant Roux[1], Côme Perrot[1], Olivier Stasse[1,2]

***Abstract*—This paper presents a novel controller for the bipedal robot Bolt. Our approach leverages a whole-body model predictive controller in conjunction with a footstep sequencer to achieve robust locomotion. Simulation results demonstrate effective velocity tracking as well as push and slippage recovery abilities. In addition to that, we provide a theoretical sensitivity analysis of the footstep sequencing problem to enhance the understanding of the results.**

***Index Terms*—Biped robot, Whole-body Model Predictive Control, Parameter sensitivity**

I. Introduction

A. Context

Bipedal robotics, with its origins tracing back to the end of the last century, has witnessed a significant surge in recent years. This trend, driven by technological advancements in areas such as actuation, and computing, has opened up a flurry of new potential applications [1], [2]. However, these new solutions require efficient controllers that can fully exploit the hardware's specificities to maximize utility.

Historically, trajectory optimization (TO) has been widely used for the control of bipedal robots. Early models, such as the Linear Inverted Pendulum (LIP) or centroidal Model Predictive Control (centroidal MPC), evolved into more sophisticated approaches like whole-body MPC (WB MPC) **li_cafe**, [3]–[5]. However, as the complexity of control problems increases, the limitations of MPC become apparent. These limitations include limited reactivity to unforeseen disturbances and insufficient computational capacity. Even though warm starting the solver near the optimal solution can drastically reduce computation time [6], these challenges persist.

Reinforcement Learning (RL) techniques have emerged as a promising alternative [7]–[9]. RL nonetheless presents notable limitations. Bipedal robots are inherently very unstable, complicating the learning process. Additionally, RL requires vast amounts of data and computational time, and often lacks performance guarantees in real-world conditions. RL algorithms can also be sensitive to variations in the training environment, limiting their robustness and generalizability.

In response to these challenges, hybrid methods, combining the strengths of trajectory optimization and reinforcement learning, are gaining popularity. Approaches presented in [10], [11] aim to merge the best of both worlds to achieve optimal performance in terms of stability and adaptability. The enthusiasm around hybrid methods highlights the relevance of traditional MPC methods, especially when dealing with highly unstable humanoids. However, in most applications, they require an additional trajectory planner, which may not account for the dynamics of the MPC, to provide a reference trajectory for the robot.

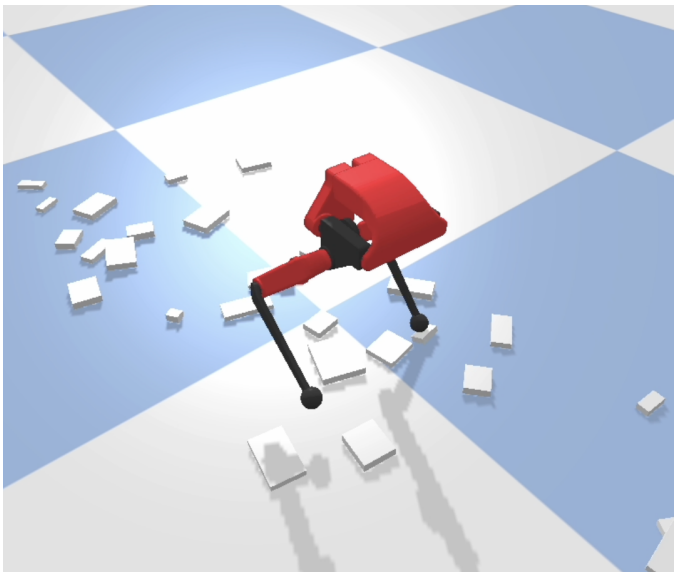

Fig. 1: Bipedal robot bolt walking in PyBullet using whole-body MPC on cluttered terrain.

Despite increasingly precise modeling and step sequencers based on simplified yet effective models, such as the Divergent Component of Motion (DCM) [12], [13], these planners are still necessary to provide robust heuristics for bipedal walking. An example of such architecture is presented in [14]. It employs a step planner followed by a foot trajectory controller and a whole-body instantaneous controller. Similarly, [15] conducted research using a low-frequency step planner (10 Hz) and a whole-body controller. A method based on iterative Linear Quadratic Gaussian (iLQG) [16] provides the capabilities to find foot sequences autonomously, but struggles to efficiently work on real hardware [17]. However, DDP with a rigid contact formulation such as the one formulated in [18] and tested on TALOS in [5] shows that if a sequence of contacts is given, WB MPC is doable on a real size humanoid robot.

B. Contributions

In this paper, propose to build upon the work of Khadiv et al. [14] by replacing the combination of the whole-body instantaneous controller and the footstep trajectory controller with a unified whole-body trajectory controller. Our approach

[1]Gepetto Team, LAAS-CNRS, Université de Toulouse, France.
[2]Artificial and Natural Intelligence Toulouse Institute, France, first-name.surname@laas.fr

aims to operate both the step sequencer and the whole-body model predictive controller (WB MPC) at the same frequency of 100 Hz, contrary to Yeganegi et al. [15]. This approach leads to the emergence of footstep trajectories, as demonstrated in previous works [4], [19], thereby simplifying robot programming and explicitly accounting for the angular momentum induced by foot motions.

Furthermore, we propose a parameter sensitivity analysis of the step sequencer based on the method proposed by Fiacco [20]. The primary aim of this analysis is to provide analytical validation of intuitions regarding the interpretation of results. Secondly, it will serve as a preliminary basis for future research.

The described technique was implemented in simulation on the Bolt robot, an affordable bipedal hardware platform designed for students and scholars. This platform utilizes an actuator concept developed by [21].

The contributions of this work are as follows:

- Integration of a step sequencer with a fixed-horizon WB MPC framework.
- Simulation of the Bolt robot in PyBullet [22] under various perturbation scenarios to evaluate the control strategy.
- Comprehensive sensitivity analysis of the step sequencer's response to disturbances in the DCM.

II. Footsteps Sequencer

A. Review of previous work

Khadiv et al. [14] proposed to compute the position and timing of the next step so that the robot's center of mass adheres to a desired velocity command while stabilizing the DCM. The DCM is the unstable component of the linear inverted pendulum model, whose dynamic equations are recalled as follows:

$$\dot{\mathbf{c}} = w_0(\boldsymbol{\zeta} - \mathbf{c}) \tag{1a}$$

$$\dot{\boldsymbol{\zeta}} = w_0(\boldsymbol{\zeta} - \mathbf{p}_0) \tag{1b}$$

where $\mathbf{c} \in \mathbb{R}^2$ is the position of the robot's center of mass, $\omega_0 \in \mathbb{R}$ is the natural frequency of the pendulum ($\omega_0 = \sqrt{\frac{g}{z_c}}$, with g being the gravitational constant and z_c the fixed height of the center of mass), $\boldsymbol{\zeta} \in \mathbb{R}^2$ is the robot's DCM, and $\mathbf{p}_0 \in \mathbb{R}^2$ is the position of the support foot.

To compute the best guest for the next step, Khadiv et al. [14] propose solving the following linearly constrained multi-objective QP problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \begin{cases} g_i(\mathbf{x}) \geq 0, & i = 1, \dots, 6 \\ \hat{h_j}(\mathbf{x}) = 0, & j = 1, \dots, 2 \end{cases} \end{aligned} \tag{2}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{p}_T^\top & \Gamma(T) & \mathbf{b}_T^\top \end{pmatrix}^\top$ with $\mathbf{p}_T = \begin{pmatrix} p_{T,x} & p_{T,y} \end{pmatrix}^\top \in \mathbb{R}^2$ representing the position of the next step, $\Gamma(T) = e^{w_0 T} \in \mathbb{R}$ the timing of the next step's contact, and $\mathbf{b}_T \in \mathbb{R}^2$ the DCM offset at the next step's contact instant ($\mathbf{b}_T = \begin{pmatrix} b_{T,x} & b_{T,y} \end{pmatrix}^\top = \boldsymbol{\zeta}_T - \mathbf{p}_T$, where $\boldsymbol{\zeta}_T$ is the DCM at the next step's contact instant). The multi-objective function is given as follows:

$$\begin{aligned} f(\mathbf{x}) = \alpha_1 & \left\| \mathbf{p}_T - \mathbf{p}_0 - \begin{pmatrix} l_{\text{nom}} \\ w_{\text{nom}} \end{pmatrix} \right\|^2 \\ & + \alpha_2 \left| \Gamma(T) - \Gamma(T_{\text{nom}}) \right|^2 \\ & + \alpha_3 \left\| \mathbf{b}_T - \begin{pmatrix} b_{x,\text{nom}} \\ b_{y,\text{nom}} \end{pmatrix} \right\|^2 \end{aligned} \tag{3a}$$

with α_1, α_2, α_3 as the weights of the different objectives, l_{nom} and w_{nom} as the desired step length and width respectively, $\Gamma(T_{\text{nom}})$ the desired contact time, and $b_{x,\text{nom}}$, $b_{y,\text{nom}}$ the desired DCM offsets whose analytical expression is available in Appendix A of [14]. The inequality constraints are given as follows:

$$g_1(\mathbf{x}) = p_{T,x} - p_{0,x} - l_{\min} \tag{4a}$$

$$g_2(\mathbf{x}) = l_{\max} - p_{T,x} + p_{0,x} \tag{4b}$$

which are constraints limiting the step length between $l_{\min}$ and $l_{\max}$,

$$g_3(\mathbf{x}) = p_{T,y} - p_{0,y} - w_{\min} \tag{4c}$$

$$g_4(\mathbf{x}) = w_{\max} - p_{T,y} + p_{0,y} \tag{4d}$$

which are constraints limiting the step width between $w_{\min}$ and $w_{\max}$,

$$g_5(\mathbf{x}) = \Gamma(T) - \Gamma(T_{\min}) \tag{4e}$$

$$g_6(\mathbf{x}) = \Gamma(T_{\max}) - \Gamma(T) \tag{4f}$$

which are constraints limiting the step contact time between $\Gamma(T_{\min})$ and $\Gamma(T_{\max})$.

The following equations are equality constraints ensuring that the DCM follows the dynamics imposed by Eq. (1b):

$$\hat{h_1}(\mathbf{x}) = p_{T,x} + b_{T,x} - p_{0,x} - (\hat{\zeta}_x - p_{0,x}) e^{-w_0 t} \Gamma(T) \tag{5a}$$

$$\hat{h_2}(\mathbf{x}) = p_{T,y} + b_{T,y} - p_{0,y} - (\hat{\zeta}_y - p_{0,y}) e^{-w_0 t} \Gamma(T) \tag{5b}$$

with t the time elapsed since the last foot contact and $\hat{\boldsymbol{\zeta}}$ the measured DCM. Note that, as explained in [23], the DCM offset is not hard constrained because this could make the problem infeasible under certain conditions. In addition, w will sometimes be referred to as w_{left} or w_{right} depending on whether the next step is executed by the left foot or the right foot.

B. Sensitivity analysis to disturbances on the measured DCM

We now propose to analyze the sensitivity of the optimal solution to perturbations on the measured DCM. These perturbations can model measurement noise or an external force that would alter the value of the DCM.

Introducing a disturbance on the measured DCM such that $\hat{\boldsymbol{\zeta}} = \boldsymbol{\zeta} + \boldsymbol{\theta}$, where $\boldsymbol{\zeta} \in \mathbb{R}^2$ is the real value of the DCM and $\boldsymbol{\theta} \in \mathbb{R}^2$ is a disturbance, the problem (2) then becomes:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \begin{cases} g_i(\mathbf{x}) \geq 0, & i = 1, \dots, 6 \\ h_j(\mathbf{x}) + \theta_j c_j(\mathbf{x}) = 0, & j = 1, \dots, 2 \end{cases} \end{aligned} \tag{6}$$

$$h_1(\mathbf{x}) = p_{T,x} + b_{T,x} - p_{0,x} - (\zeta_x - p_{0,x})e^{-w_0 t}\Gamma(T) \quad (7a)$$
$$h_2(\mathbf{x}) = p_{T,y} + b_{T,y} - p_{0,y} - (\zeta_y - p_{0,y})e^{-w_0 t}\Gamma(T) \quad (7b)$$

$$c_1(\mathbf{x}) = -e^{-w_0 t}\Gamma(T) \quad (8a)$$
$$c_2(\mathbf{x}) = -e^{-w_0 t}\Gamma(T) \quad (8b)$$

The theoretical result of sensitivity of a QP problem with respect to a parameter is provided by [20]. In the following, we use the superscript * on an expression to denote that the latter is evaluated at $\boldsymbol{\theta} = 0$. Noting that (a) the functions of Equation (6) are twice differentiable, (b) the second-order sufficiency conditions hold at $\mathbf{x}^*$, (c) $\{\nabla_{\mathbf{x}} g_i^*\}, i = 1, \ldots, 6$, $\{\nabla_{\mathbf{h}} g_j^*\}, j = 1, \ldots, 2$ are linearly independent, and (d) $u_i^* > 0$ when $g_i^*(\mathbf{x}) = 0$ where u_i and w_i are the Lagrangian multipliers, then the following set of equations is satisfied at $(\mathbf{x}, \mathbf{u}, \mathbf{w}) = (\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*)$, $\boldsymbol{\theta} = 0$:

$$\begin{aligned} &\nabla_{\mathbf{x}} f - \sum_{i=1}^{6} u_i \nabla_{\mathbf{x}} g_i + \sum_{j=1}^{2} w_j (\nabla_{\mathbf{x}} h_j + \theta_j \nabla_{\mathbf{x}} c_j) = 0 \\ &u_i g_i(\mathbf{x}) = 0, \qquad i = 1, \ldots, 6 \\ &h_j(\mathbf{x}) + \theta_j c_j(\mathbf{x}) = 0, \quad j = 1, 2 \end{aligned} \quad (9)$$

Let $F(\mathbf{x}, \boldsymbol{\theta})$ denote the function composed of the terms on the left-hand side of Eq. 9. Applying the implicit function theorem, we deduce that:

$$\frac{\partial(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*)}{\partial \boldsymbol{\theta}} = -J_{F^*}^{-1}(\mathbf{x}^*, \mathbf{u}^*, \mathbf{w}^*) J_F(\boldsymbol{\theta}) \quad (10)$$

$$J_{F^*}(\mathbf{x}^*) = \begin{pmatrix} \text{diag}(\alpha_1, \alpha_1, \alpha_2, \alpha_3, \alpha_3) & -G^* & H^* \\ \text{diag}(u_i^*)(G^*)^\top & \text{diag}(g_i^*) & [0]_{6\times 2} \\ (H^*)^\top & [0]_{2\times 6} & [0]_{2\times 2} \end{pmatrix} \quad (11)$$

$$J_F(\theta) = \begin{pmatrix} C^* \text{diag}(w_j^*) \\ [0]_{6\times 2} \\ \text{diag}(c_j^*) \end{pmatrix} \quad (12)$$

where $G^* = \begin{pmatrix} \nabla_{\mathbf{x}} g_1^* & \cdots & \nabla_{\mathbf{x}} g_6^* \end{pmatrix}$, $H^* = \begin{pmatrix} \nabla_{\mathbf{x}} h_1^* & \nabla_{\mathbf{x}} h_2^* \end{pmatrix}$ and $C^* = \begin{pmatrix} \nabla_{\mathbf{x}} c_1^* & \nabla_{\mathbf{x}} c_2^* \end{pmatrix}$.

Eq. (10) allows us to numerically evaluate the sensitivity of the optimal solution of the QP problem with respect to perturbations on the DCM. For example, Fig. 2 presents the optimal solutions $p_{T,y}$ and $b_{T,y}$ when subjected to perturbations following a Gaussian distribution $\mathcal{N}(0, 0.005)$, with no active inequality constraints, using the input parameters of the problem from Table I. We notice that these solutions form planes in the perturbation space, with the plane for $p_{T,y}$ having a steeper slope than that for $b_{T,y}$. This result, reflected in $(\frac{\partial p_{T,y}^*}{\partial \theta_y} = 5.18) > (\frac{\partial b_{T,y}^*}{\partial \theta_y} = 5.18e-3)$, matches the design choice we made to have $\alpha_3 \gg \alpha_1$. Tuning the weight of the multi-objective function in that way should lead the robot to prioritize balancing over speed tracking which is confirmed by the theoretical sensitivity analysis.

TABLE I: Parameters of the step sequencer used for walking

Parameter	min	nom	max
$(\alpha_1, \alpha_2, \alpha_3)$	-	(1e3, 1, 1e6)	-
z_c	-	0.31	-
p_0	-	(-0.12, 0.10)	-
t	-	0.229	-
$\hat{\boldsymbol{\zeta}}$	-	(-0.12, -0.07)	-
l	-0.3	0.1	0.3
w_{left}	-0.40	-0.25	-0.10
w_{right}	0.10	0.25	0.40
T	0.1	0.3	1.0

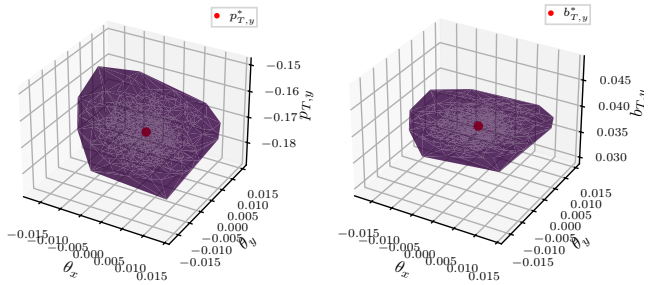

Fig. 2: Solution space example of $p_{T,y}$ (left) and $b_{T,y}$ (right) as a function of the DCM disturbance $\boldsymbol{\theta} \sim \mathcal{N}(0, 0.005)$ generated with 1000 samples. The optimal solution is represented in red, and no inequality constraints is active.

C. Fixed-Horizon MPC Sequencer

To enable the WB MPC to predict a state trajectory throughout its prediction horizon (see Section III), the step sequence has to extend at least to the end of the MPC's horizon. To generate a step sequence up to an appropriate time H_u, we propose to iteratively solve problem (2) and reuse the solution computed at time k as the initial condition of the problem solved at time $k+1$. This process can be carried out until a contact time exceeding the desired time horizon is obtained. Following an MPC scheme, this strategy is used before every iteration of the WB MPC to generate a new reference step sequence. The problem can be summarized as follows:

$$\begin{aligned} \min_{\mathbf{x}_{k+1}} \quad & f(\mathbf{x}_{k+1}, \mathbf{x}_k) \\ \text{s.t.} \quad & \begin{cases} g_i(\mathbf{x}_{k+1}, \mathbf{x}_k) \geq 0, & i = 1, \ldots, 6 \\ h_j(\mathbf{x}_{k+1}, \mathbf{x}_k) = 0, & j = 1, \ldots, 2 \end{cases} \end{aligned} \quad (13)$$

$$\begin{aligned} f(\mathbf{x}_{k+1}, \mathbf{x}_k) = \alpha_1 & \left\| \mathbf{p}_{T_{k+1}} - \mathbf{p}_{T_k} - \begin{pmatrix} l_{\text{nom}} \\ w_{k,\text{nom}} \end{pmatrix} \right\|^2 \\ & + \alpha_2 \left| \Gamma(T_{k+1}) - \Gamma(T_k + T_{\text{nom}}) \right|^2 \\ & + \alpha_3 \left\| \mathbf{b}_{T_{k+1}} - \begin{pmatrix} b_{k,x,\text{nom}} \\ b_{k,y,\text{nom}} \end{pmatrix} \right\|^2 \end{aligned} \quad (14a)$$

$$g_1(\mathbf{x}_{k+1}, \mathbf{x}_k) = p_{T_{k+1},x} - p_{T_k,x} - l_{\min} \tag{15a}$$
$$g_2(\mathbf{x}_{k+1}, \mathbf{x}_k) = l_{\max} - p_{T_{k+1},x} + p_{T_k,x} \tag{15b}$$
$$g_3(\mathbf{x}_{k+1}, \mathbf{x}_k) = p_{T_{k+1},y} - p_{T_k,y} - w_{k,\min} \tag{15c}$$
$$g_4(\mathbf{x}_{k+1}, \mathbf{x}_k) = w_{k,\max} - p_{T_{k+1},y} + p_{T_k,y} \tag{15d}$$
$$g_5(\mathbf{x}_{k+1}, \mathbf{x}_k) = \Gamma(T_{k+1}) - \Gamma(T_k + T_{\min}) \tag{15e}$$
$$g_6(\mathbf{x}_{k+1}, \mathbf{x}_k) = \Gamma(T_k + T_{\max}) - \Gamma(T_{k+1}) \tag{15f}$$

$$h_1(\mathbf{x}_{k+1}, \mathbf{x}_k) = p_{T_{k+1},x} + b_{T_{k+1},x} - p_{T_k,x} - (\hat{\zeta}_{k,x} - p_{T_k,x})e^{-w_0 T_k}\Gamma(T_{k+1}) \tag{16a}$$
$$h_2(\mathbf{x}_{k+1}, \mathbf{x}_k) = p_{T_{k+1},y} + b_{T_{k+1},y} - p_{T_k,y} - (\hat{\zeta}_{k,y} - p_{T_k,y})e^{-w_0 T_k}\Gamma(T_{k+1}) \tag{16b}$$

where $\mathbf{x}_{k+1}$ represents the next solution and $\mathbf{x}_k$ the previous solution. The parameter w_k varies depending on whether the next step is taken by the left foot or the right foot.

Algorithm 1 summarizes the MPC step sequencer:

Algorithm 1 Footsteps sequencer

$t \leftarrow t_{\text{mea}}$, $t_0 \leftarrow t_{\text{mea}}$, $\hat{\boldsymbol{\zeta}}_k \leftarrow \boldsymbol{\zeta}_{\text{mea}}$
$\mathbf{x}_k \leftarrow \begin{pmatrix} \mathbf{p}_{\text{ini}}^\top & \Gamma(t_{\text{mea}}) & [0]_{1\times 2} \end{pmatrix}^\top$
$S \leftarrow \emptyset$
repeat
  $\mathbf{x}_{k+1} \leftarrow$ Solve (13) using $\mathbf{x}_k$
  $t \leftarrow T_{k+1}$, $\hat{\boldsymbol{\zeta}}_k \leftarrow \mathbf{p}_{T_{k+1}} + \mathbf{b}_{T_{k+1}}$
  $S \leftarrow S \cup \mathbf{x}_{k+1}$
  $\mathbf{x}_k \leftarrow \mathbf{x}_{k+1}$
until $t \geq t_0 + H_u$
Return S

where t_{mea} is the time at which the algorithm is called, $\boldsymbol{\zeta}_{\text{mea}}$ is the measured DCM at $t = t_{\text{mea}}$, and $\mathbf{p}_{\text{ini}}$ is the position of the foot currently in support. S is the list containing the K steps of the generated sequence. The value of the nominal DCMs and the changes in w depending on whether the next step is taken by the left foot or the right foot are not detailed here for the sake of clarity. Fig. 3 illustrates an example of a step sequence generated using the parameters from Table I and a horizon H_u of 3 s for walking at a velocity of $V_x^* = 0.3\ m.s^{-1}$. The figure also shows that all hard constraints are satisfied, and the DCM remains non-divergent over time. This sequence can subsequently serve as a reference for WB MPC.

III. Whole-body MPC

A. Discretized Optimal Control Formulation

Given the step sequence generated by the sequencer, we now aim to compute a torque sequence to control the robot. To account for all dynamic effects, we employ a whole-body

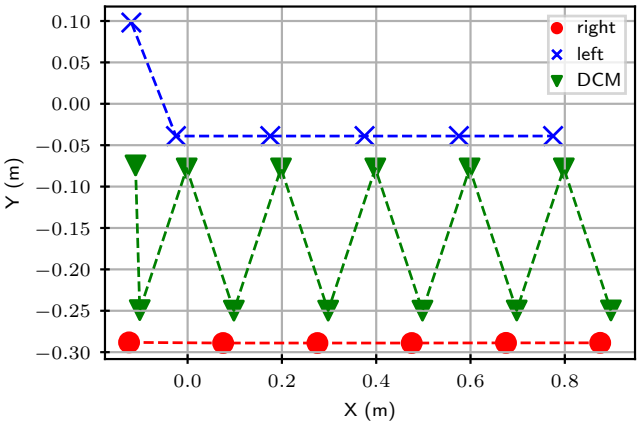

Fig. 3: Position of the feet and DCM in the Cartesian plane along a generated step sequence. The walking sequence was generated with $H_u = 3\ s$ using the input parameters from Table I. The imposed speed is $V_x^* = \frac{l_{\text{nom}}}{T_{\text{nom}}} = 0.33 m.s^{-1}$. All constraints are satisfied, and the nominal speed is achieved.

MPC, which iteratively solves the Optimal Control Problem (OCP) described as follows:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & \sum_{i=0}^{N-1} l_i(\mathbf{x_i}, \mathbf{u_i}) + l_N(\mathbf{x_N}) \\ \text{s.t.} \quad & \mathbf{x}_0 = \mathbf{x}_{init} \\ & f_i(\mathbf{x_i}, \mathbf{u_i}) = \mathbf{x_{i+1}} \quad \forall i \in [0, N-1] \end{aligned} \tag{17}$$

where $\mathbf{x}$ and $\mathbf{u}$ represent the decision variables for the state and control inputs, respectively. N denotes the discrete horizon length (discretization of H_u), while $l_i(\mathbf{x}, \mathbf{u})$ corresponds to the running costs and $l_N(\mathbf{x_N})$ the terminal cost. The initial state, $\mathbf{x}_0$, is set to $\mathbf{x}_{\text{init}}$, and $f_i(\mathbf{x}_i, \mathbf{u}_i)$ represents the whole-body dynamics from time i to $i+1$, including the contact dynamics constraints for both the left and right foot. The specific time instances i for these contacts are provided for each foot by the step sequencer. Note that the meaning of $\mathbf{x}$, $\mathbf{u}$, and f differs from the previous section, where they represented different quantities.

B. Costs functions

The running cost $l_i(\mathbf{x_i}, \mathbf{u_i})$ is the sum of the following costs:

1) *Foot height tracking cost*: The cost of foot height tracking ensures that the robot lifts its foot appropriately. It is described by the following equation:

$$\ell^f_{track}(\mathbf{x}) = \left\| x_z^f - x_z^{f^*} \right\|^2_{w_{track}} \tag{18}$$

where f denotes the flying foot, $\mathbf{x}$ the state of the robot, x_z^f, $x_z^{f^*} \in \mathbb{R}$ respectively the measured and desired flying foot height, and $w_{track} \in \mathbb{R}$ a weighting hyperparameter.
The reference trajectory of the foot height is a polynomial function ensuring that the height and the velocity of the foot height are zero at both the initial and final

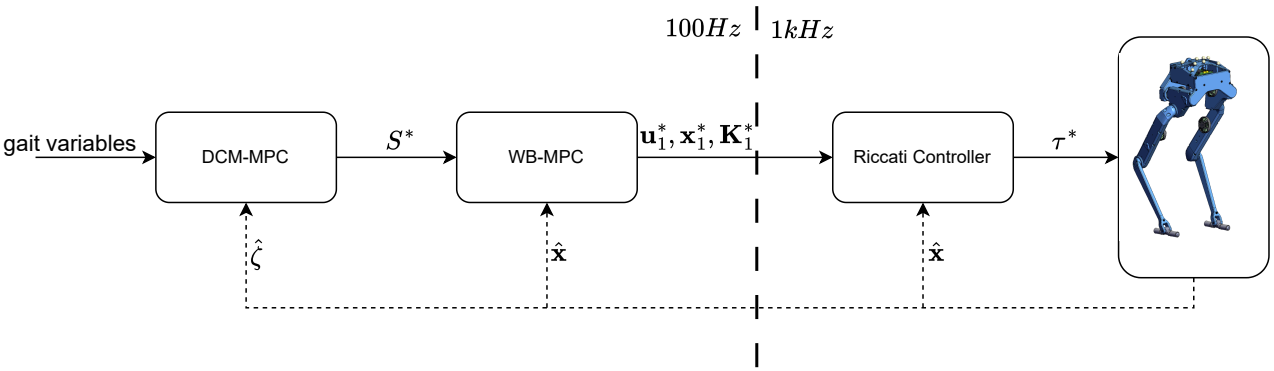

Fig. 4: Simplified architecture of Bolt's walking controller.

instants, and that the foot reaches a predefined height at the midpoint of the duration, as described by:

$$x_z^{f^*}(t) = \frac{16H}{t_f^4}t^4 - \frac{32H}{t_f^3}t^3 + \frac{16H}{t_f^2}t^2 \tag{19}$$

where $t \in \mathbb{R}$ is a variable ranging from 0 to $t_f \in \mathbb{R}$, t_f is the total duration of the step, and $H \in \mathbb{R}$ is the maximum foot height at $\frac{t_f}{2}$.
Note that the foot contact positions calculated by the sequencer are not predefined, they naturally emerge from the solver used to address the WB MPC problem.

2) *Regularization costs*: A regularization cost around zero command indirectly limits the torques applied to the joints. It is implemented according to the following equation:

$$\ell_{\mathbf{u}}(\mathbf{u}) = \|\mathbf{u}\|^2_{w_{\mathbf{u}}} \tag{20}$$

where $w_{\mathbf{u}} \in \mathbb{R}$ is a weighting hyperparameter.
Additionally, a regularization cost of the state around a reference posture is added to ensure that the robot's posture tends towards it. The cost is described as:

$$\ell_X(\mathbf{x}) = \|(\mathbf{x} - \mathbf{x}^*)\|^2_{w_{\mathbf{x}}} \tag{21}$$

where $\mathbf{x}^* \in \mathbb{R}^{n_\mathbf{x}}$ (where $n_\mathbf{x}$ is the size of the robot state vector) is the desired reference posture, and $w_\mathbf{x} \in \mathbb{R}^{n_\mathbf{x} \times n_\mathbf{x}}$ is a weighting hyperparameter.

3) *Boundary cost*: A barrier cost is added to the control to ensure that it never exceeds the physical limits of the robot's actuators. The cost is given as follows:

$$\ell^f_{bu}(\mathbf{x}) = \|\min(\max(\mathbf{u}, \bar{\mathbf{u}}), \underline{\mathbf{u}})\|^2_{w_{bu}} \tag{22}$$

where $\underline{\mathbf{u}} \in \mathbb{R}^{n_\mathbf{u}}$ (where $n_\mathbf{u}$ is the size of the robot control vector) and $\bar{\mathbf{u}} \in \mathbb{R}^{n_\mathbf{u}}$ are respectively the lower and upper boundaries of the robot torque outputs. $w_{bu} \in \mathbb{R}$ is a weighting hyperparameter.

The cost $l_N(\mathbf{x_N})$ is the sum of the previous state-dependent costs along with an additional cost on the position of the center of mass:

$$\ell_{\text{CoM}}(\mathbf{x}) = \|\mathbf{c} - \mathbf{c}^*\|^2_{w_{\text{CoM}}} \tag{23}$$

where $\mathbf{c} \in \mathbb{R}^3$ is the position of the CoM, $\mathbf{c}^* \in \mathbb{R}^3$ is the reference position of the CoM, and $w_{\text{CoM}} \in \mathbb{R}$ is a weighting hyperparameter.

The reference for the CoM is calculated by solving Eq. (1a):

$$\mathbf{c}^* = (\hat{\mathbf{c}}_{K-1} - \hat{\zeta}_{K-1})e^{w_0(T_{K-1} - H_u)} + \hat{\zeta}_{K-1} \tag{24}$$

where the subscript $_{K-1}$ represents the index of the last step whose contact is before the horizon of the list S. This ensures that the robot's center of mass reaches a stable position while adhering to the desired setpoint.

C. Control pipeline

The full control structure consists of two parallel processes (see Fig. 4):

- The high-level process runs at 100 Hz and solves the DCM-MPC and WB-MPC using OSQP and Crocoddyl [18], respectively. To account for computational delay, which roughly corresponds to the interval between two WB-MPC nodes (roughly 10 ms), the first node of the computed solution is skipped and the high-frequency controller directly receives results from the second node.
- The optimization process yields a by-product that is leveraged for low-level control of the robot at a frequency of 1 kHz. We designate these as Riccati gains throughout the remainder of this paper. The control law, derived from [19], is given by:

$$\tau = \mathbf{u}_1 + K_1(\hat{\mathbf{x}} - \mathbf{x_1}) \tag{25}$$

where $\mathbf{u}_1 \in \mathbb{R}^{n_\mathbf{u}}$ and $\mathbf{x}_1 \in \mathbb{R}^{n_\mathbf{x}}$ are the optimal control input and state computed by the solver, respectively. $\hat{\mathbf{x}} \in \mathbb{R}^{n_\mathbf{x}}$ is the robot's measured state at 1 kHz. Finally, the matrix $K_1 \in \mathbb{R}^{6 \times (n_\mathbf{x} - 1)}$ is the Riccati gain matrix at node 1 given by:

$$K_1 \triangleq \frac{\partial \mathbf{u}}{\partial \mathbf{x}}\Big|_{\mathbf{x}_1} \tag{26}$$

This setup ensures high-frequency feedback and robust real-time control.

IV. Simulation Results

A. Setup

The objective of these experiments in simulation is to evaluate the robot's ability to follow a given velocity command, demonstrate robustness against external perturbations, and adapt its gait dynamically. The simulation is conducted in PyBullet on an Apple Mac M3 MAX CPU clocked at 4.05 GHz with 16 cores.

To enhance efficiency, a simplified model of the Bolt robot is employed[1]. The dynamic parameters and foot geometries remain unchanged to ensure consistency for sim-to-real transfer. Further details on the robot's hardware can be found in [21].

Initial conditions are identical for all simulations: Bolt starts with both feet on the ground and in the configuration specified by the SRDF. Three scenarios are designed:

1) *Walking with and without disturbance*: Bolt follows a walking command while experiencing occasional external perturbations.
2) *Cluttered terrain walking*: Bolt walks on a cluttered terrain.
3) *Velocity reference transition*: Bolt transitions from a velocity command to another.

B. Nominal walking with and without perturbations

We propose to compare the walking motion of Bolt with a target speed command $v_X^* = 3.3\,m.s^{-1}$ with and without perturbations. Fig. 5 shows the y-axis positions of the CoM, DCM and feet positions of bolt over time. The top plot depicts unperturbed walking over a two-second interval, while the bottom plot shows walking with a perturbation of $6.3\,\mathrm{N}$ applied at the base for $0.1\,\mathrm{s}$ (equivalent to $0.63\,\mathrm{N.s}$) at $t = 4\,\mathrm{s}$. We observe that Bolt is capable of rejecting the perturbation by adjusting the step sequence and subsequently resumes nominal walking.

Next, we analyze the base velocity of Bolt. Fig. 6 shows the base velocities in the x and y directions over time, both without and with the same perturbation as before. The top plot presents the unperturbed velocities, whereas the bottom plot shows the velocities under perturbation. We observe that the accuracy of x-axis velocity tracking decreases at the onset of the perturbation and returns to the target velocity once the perturbation is rejected. This behavior is expected because, as explained in Section II-B, in case of a DCM perturbation, the solver prioritizes maintaining Bolt's balance at the cost of velocity tracking. Additionally, we note on the top plot an average error $\epsilon_{\bar{\mathbf{V}}} = \left|\bar{\mathbf{V}} - \mathbf{V}^*\right| = \begin{pmatrix}5.2 & 1.1\end{pmatrix}^\top\ mm.s^{-1}$ in the velocity tracking. It is mainly due to discrepancies between the LIPM model and the simulator model, particularly the fact that angular momentum is neglected in the LIPM model. Solutions exploring 3D DCM-based methods [12] address this issue, however such an extension is beyond the scope of our work.

[1]https://github.com/Gepetto/example-robot-data/tree/master/robots/bolt_description

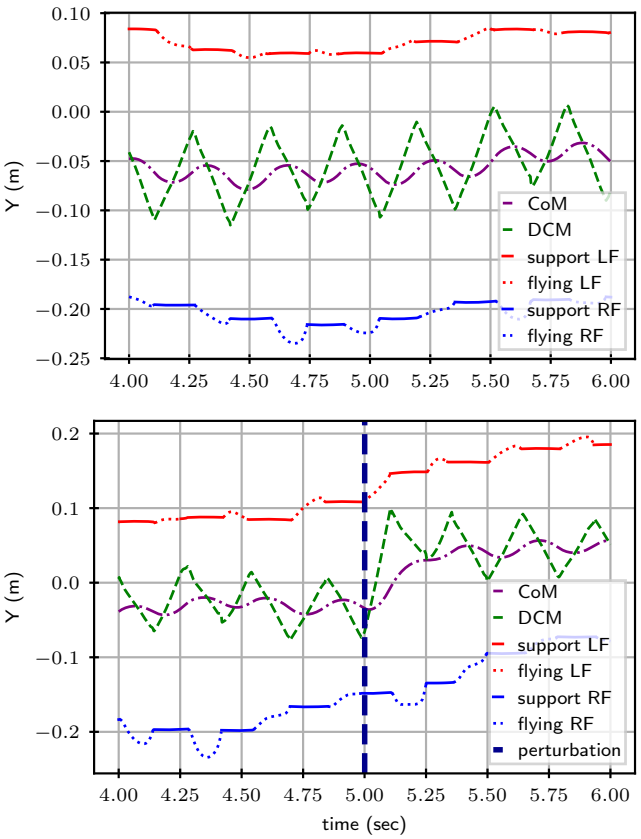

Fig. 5: Comparison of the CoM (purple), the DCM (green), and the feet positions (red for left, blue for right) along the y-axis over time during walking at $V_x^* = 0.33\,m.s^{-1}$. Top: Unperturbed walking. Bottom: Walking perturbed at $t = 5\,\mathrm{s}$ by an applied force of $F = 6.3\,\mathrm{N}$ along the y-axis at the robot's base for 0.1 s.

C. Nominal walking on cluttered terrain

We now demonstrate the robustness of the controller on Bolt against slips and rough terrain (see Fig. 1) with a reference speed $V_x^* = 0.3\,m.s^{-1}$. The perturbations involve the generation of rectangular parallelepipeds with side lengths ranging from 1 cm to 5 cm and heights ranging from 5 mm to 8 mm, placed on the robot's path, with an average density of 10 obstacles per square meter. Fig. 7 shows the y axis position of the CoM, the DCM, and the foot positions over time, highlighting an instance of left foot slippage. We observe that the controller adapts the step sequence to compensate for slips and reject the resulting perturbations.

D. Transition from standing to walking

Fig. 8 shows the robot's base velocity along the x axis over time. The command initially sets the velocity to $0\ m.s^{-1}$ (step in place), transitioning to $0.3\ m.s^{-1}$ at $t = 5\mathrm{s}$. We measure a rise time of 4.35s for the robot to reach its final velocity value. This delay can be attributed to a significant increase in angular moments caused by the velocity change, leading

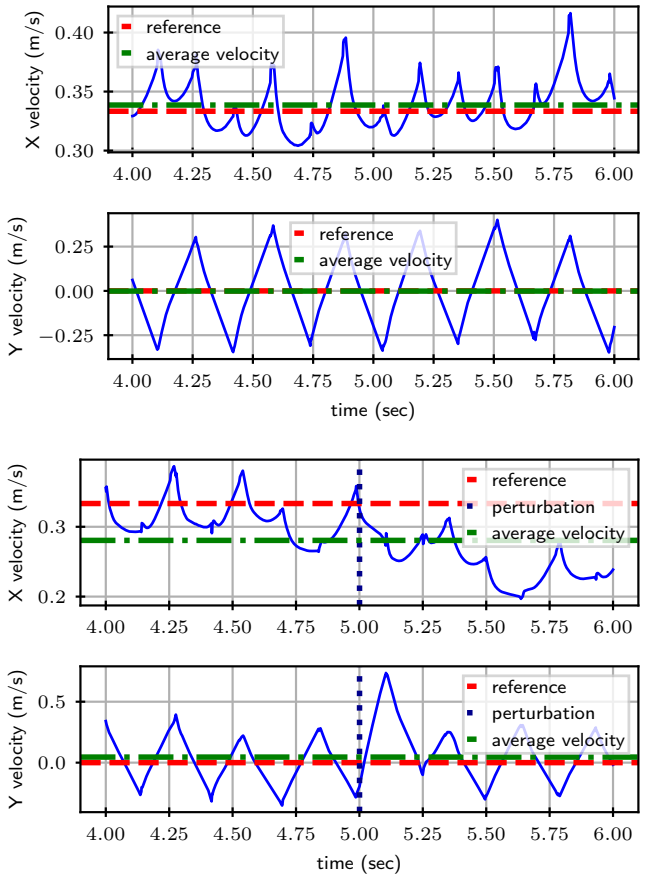

Fig. 6: Comparison of the base velocity (blue) and the average velocity (green) along the x and y axes over time when walking at $V_x^* = 0.33\,m.s^{-1}$ (shown in red). Top: Unperturbed walking. Bottom: Walking perturbed at $t = 5\,\mathrm{s}$ by an applied force of $F = 6.3\,\mathrm{N}$ along the y-axis at the robot's base for 0.1 s.

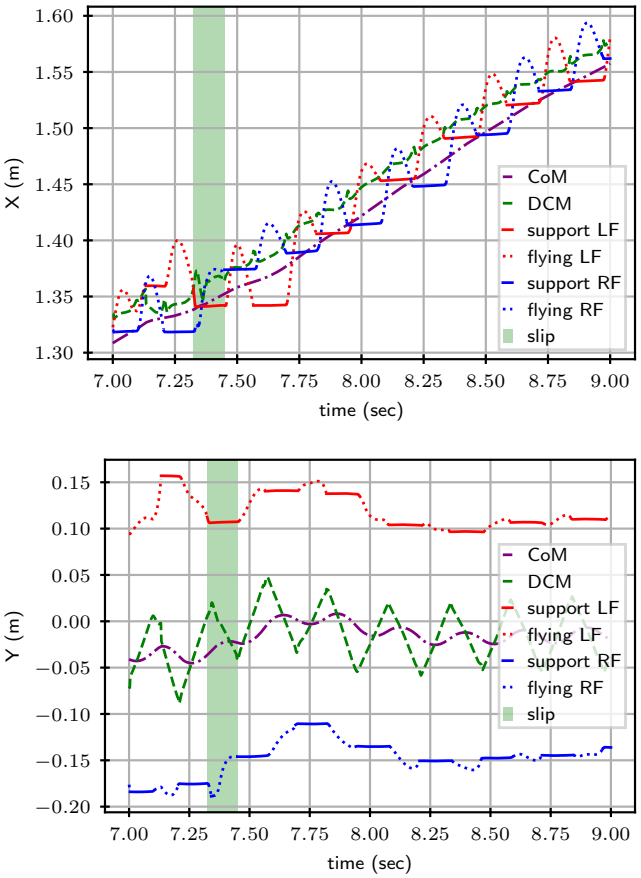

Fig. 7: CoM (purple), the DCM (green), and the feet positions (red for left, blue for right) along the y-axis over time when walking at $V_x^* = 0.33\,m.s^{-1}$. The left foot is slipping in the green area.

to a larger discrepancy between the LIPM and the simulator model. This difference can be modeled as a perturbation on the DCM. As explained in section II-B, foot placements are more sensitive to perturbations than robot balance. Therefore, while striving to maintain balance, the feet positions deviate from the optimal position, resulting in a slower achievement of the desired velocity.

V. Discussion and Conclusion

In this study, we demonstrated that successful control of a highly unstable bipedal robot can be achieved using a footstep sequencer and a WB MPC operating at 100 Hz. Our approach eliminates the need for explicit footstep trajectories, as they naturally emerge when solving the WB MPC. Our results show that the robot exhibits significant capabilities in executing velocity transitions, recovering from perturbations, and managing foot slippage. Sensitivity analysis provides additional insights, aligning with previous research [14], [15] while offering a deeper theoretical understanding.

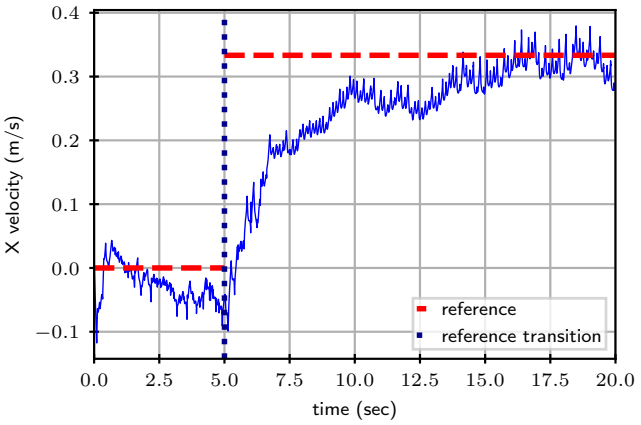

Fig. 8: Base velocity (blue) and the average velocity (green) along x-axis over time when walking at $V_x^* = 0.0\,m.s^{-1}$ before $t = 5s$ and $V_x^* = 0.33\,m.s^{-1}$ after $t = 5s$ (shown in red).

However, we observed that the desired speed was not always attained and that velocity transitions were slow, which can limit the robot's performance in scenarios requiring rapid speed changes. This outcome was anticipated due to the assumptions imposed by the LIPM.

In the short term, we plan to deploy and validate this controller on the physical Bolt robot. For future work, we plan to leverage the sensitivity analysis computed in this study as a foundational work to estimate the sensitivity of the whole control structure (DCM MPC + WB MPC). We would additionally need to compute the sensitivity WB MPC with respect to the contact timings calculated by the step sequencer. This study could allow to provide more depth in the theoretical interpretation of the performance of our control structure as well as offering robustness guarantees.

ACKNOWLEDGMENT

This work was supported by the cooperation agreement ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015). We extend our gratitude to Tommaso Belvedere for his discussions on sensitivity parameters in MPC.

REFERENCES

[1] Unitree, *Unitree's first universal humanoid robot*, Jun. 2024. [Online]. Available: https://www.unitree.com/h1/.

[2] F. Intelligence, *General-purpose humanoid robot fourier gr-1*, Jun. 2024. [Online]. Available: https://fourierintelligence.com/gr1/.

[3] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *IEEE TRO*, vol. 40, pp. 43–63, 2024.

[4] A. Assirelli, F. Risbourg, G. Lunardi, T. Flayols, and N. Mansard, "Whole-Body MPC without Foot References for the Locomotion of an Impedance-Controlled Robot," 2022, Technical report. [Online]. Available: https://laas.hal.science/hal-03778738.

[5] E. Dantec, R. Budhiraja, A. Roig, *et al.*, "Whole Body Model Predictive Control with a Memory of Motion: Experiments on a Torque-Controlled Talos," in *ICRA*, May 2021, pp. 8202–8208.

[6] T. S. Lembono, C. Mastalli, P. Fernbach, N. Mansard, and S. Calinon, "Learning how to walk: Warm-starting optimal control solver with memory of motion," in *ICRA*, 2020, pp. 1357–1363.

[7] R. P. Singh, Z. Xie, P. Gergondet, and F. Kanehiro, "Learning bipedal walking for humanoids with current feedback," *IEEE Access*, vol. 11, pp. 82 013–82 023, 2023.

[8] Z. Zhuang, S. Yao, and H. Zhao, *Humanoid Parkour Learning*, Jun. 2024. [Online]. Available: http://arxiv.org/abs/2406.10759.

[9] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, 2024.

[10] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control," *Science Robotics*, vol. 9, no. 86, 2024.

[11] A. Romero, Y. Song, and D. Scaramuzza, *Actor-critic model predictive control*, 2024. arXiv: 2306.09852 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2306.09852.

[12] J. Englsberger, C. Ott, and A. Albu-Schaffer, "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion," *IEEE TRO*, vol. 31, no. 2, pp. 355–368, Apr. 2015.

[13] Z. Zhang, L. Zhang, S. Xin, N. Xiao, and X. Wen, "Robust Walking for Humanoid Robot Based on Divergent Component of Motion," *Micromachines*, vol. 13, no. 7, p. 1095, 2022.

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

[15] M. H. Yeganegi, M. Khadiv, A. D. Prete, S. A. A. Moosavian, and L. Righetti, "Robust walking based on mpc with viability guarantees," *IEEE TRO*, vol. 38, no. 4, pp. 2389–2404, Aug. 2022.

[16] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IROS*, 2012.

[17] J. Koenemann, A. Del Prete, Y. Tassa, *et al.*, "Whole-body model-predictive control applied to the hrp-2 humanoid," in *IROS*, 2015.

[18] C. Mastalli, R. Budhiraja, W. Merkt, *et al.*, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *ICRA*, 2020.

[19] E. Dantec, M. Taix, and N. Mansard, "First order approximation of model predictive control solutions for high frequency feedback," *IEEE RA-L*, vol. 7, no. 2, pp. 4448–4455, 2022.

[20] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, 1983, vol. Volume 165.

[21] F. Grimminger, A. Meduri, M. Khadiv, *et al.*, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE RA-L*, vol. 5, no. 2, pp. 3650–3657, 2020.

[22] E. Coumans and Y. Bai, *Pybullet, a python module for physics simulation for games, robotics and machine learning*, http://pybullet.org, 2021.

[23] E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti, "Variable horizon mpc with swing foot dynamics for bipedal walking control," *IEEE RA-L*, vol. 6, no. 2, pp. 2349–2356, 2021.