# Improved Control Scheme for the Solo Quadruped and Experimental Comparison of Model Predictive Controllers

Pierre-Alexandre Léziart, Thomas Corbères, Thomas Flayols, Steve Tonneau, Nicolas Mansard, Philippe Souères

# Improved Control Scheme for the Solo Quadruped and Experimental Comparison of Model Predictive Controllers

Pierre-Alexandre Léziart[1], Thomas Corbères[2], Thomas Flayols[1,3]
Steve Tonneau[2], Nicolas Mansard[1,3] and Philippe Souères[1]

***Abstract*—This paper presents significant improvements to the nominal control architecture of the open-access Solo-12 quadruped that were done to implement and compare different centroidal Model Predictive Controllers (MPC). This work was motivated by our previous study in which various MPC schemes of increasing complexity were tested in simulation. They range from a simplified linearized model with contact points fixed by a heuristic, to a nonlinear one that also optimizes the contact points locations. We describe the developments that were necessary to implement such control schemes on the real robot while doubling its maximum velocity. Notably, to synthesize a stable whole-body controller, the inverse dynamics resolution was replaced by a mixed inverse kinematics and quadratic programming scheme. These developments enabled the successful deployments of the various centroidal MPCs on Solo-12. Experimental results show that all MPCs lead to quite similar performances with the proposed whole-body controller and, as a consequence, do not confirm the result of previous simulation study that concluded on the preeminence of the nonlinear centroidal MPC optimizing both the center of mass trajectory and the foot placements.**

***Index Terms*—Legged Robots, Motion Control, Optimization and Optimal Control**

## I. Introduction

KEEPING balance while performing agile locomotion is a long-standing research problem for legged robots [1], [2]. As movements get more demanding, both the choices of dynamical model and ground contact locations become critical. On the one hand, dynamical effects that might be negligible in quasi-static scenarios have a significant influence at higher speeds [3]. On the other hand, challenging terrains require a particular consideration for contacts, as they are for legged robots the only way to apply forces on the environment to stabilize themselves [4], [5]. Over the years various methods have been developed to tackle these challenges. Contact locations can be carefully planned with optimal control approaches that exploit information about the environment [6], [7]. Since for such methods a well-chosen heuristic can play a crucial role in the quality of the results, some planners explore several of them concurrently so that they benefit from each other [8], [9] and, as such, are less dependent on well-tuned functions that can require considerable expertise. To address this issue, studies have been conducted to refine heuristics with training using genetic algorithms [10], or by manually providing adequate footsteps at a few key frames to act as a teacher the policy can learn from [11]. By considering incoming contact locations and a model of the dynamics over a prediction horizon, MPCs can take privileged decisions to handle disturbances and perturbations to the system, compared to instantaneous controllers. Whole-body dynamics allows a complete exploitation of our knowledge about the robot model [12], yet its high dimensionality and non-linearity remains computationally demanding on a prediction horizon. The classical alternative is to combine a MPC using reduced dynamics with a low-level controller that leverages a full model of the robot [13], [14]. For quadrupeds, centroidal dynamics stands as a sound reduction that neglects limb dynamics by approximating the angular momentum to the rotation of the trunk where most of its mass is located [15], [16].

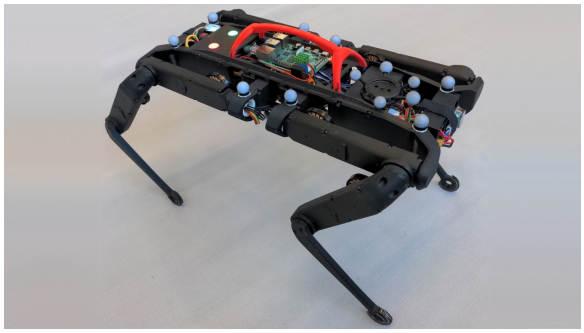


Fig. 1: Solo-12 quadruped powered with batteries. The whole control architecture can run on an on-board Raspberry Pi 4.

This paper presents key improvements done on the control architecture of the Solo-12 quadruped [17] to implement and test on the real robot several MPC variants that were previously studied in simulation only [18]. It is based on extensive experimental efforts that build upon the previous formulation and made possible their real-world deployment. The objective is to evaluate whether experimental results confirm the preeminence of the nonlinear centroidal MPC

Manuscript received: February, 24, 2022; Revised May, 28, 2022; Accepted June, 27, 2022.

This letter was recommended for publication by Associate Editor F. Pierri and Editor C. Gosselin upon evaluation of the reviewers' comments. This work was supported by the MEMMO European Union project within the H2020 Program under Grant Agreement No. 780684, by the TERRINet H2020-INFRAIA European project, by the ANITI 3IA Institute (ANR-19-P3IA-000) and by the COCOPIL Région Occitanie project.

[1]Léziart, Flayols, Mansard and Souères are with Gepetto team, LAAS-CNRS, Université de Toulouse, France paleziart@laas.fr

[2]Corbères and Tonneau are with School of Informatics, University of Edinburgh, United Kingdom

[3]Flayols and Mansard are with Artificial and Natural Intelligence Toulouse Institute, France

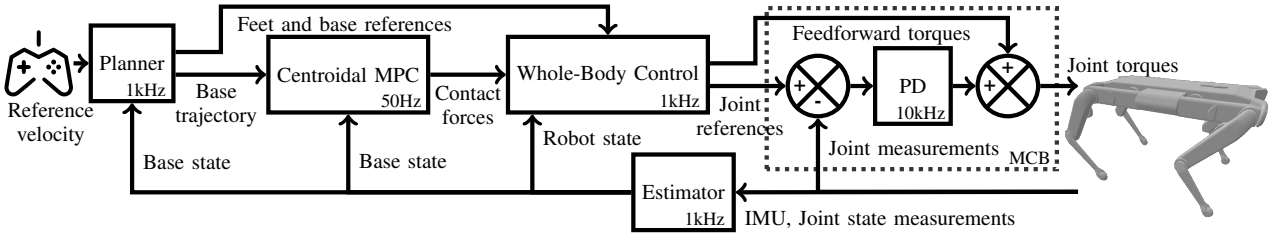



Fig. 2: Nominal reactive walking control architecture. We will explain how the new formulation partly removes the need for the footsteps planner. The PD controller is directly performed by the motor control board (MCB) of the robot.

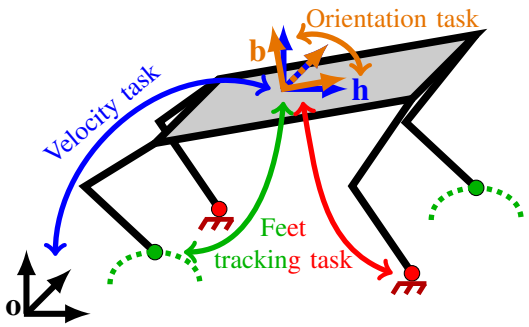



Fig. 3: Tasks and frames of the WBC.

scheme optimizing footholds over a modular scheme that relies on heuristics for foot placement and on a simplified model, as observed in simulation [18]. Depending on how well the heuristic-free MPC can perform on the real robot compared to variants that rely on hand-defined heuristics, it could be a way to relieve the need for expert knowledge at the cost of modularity by centralizing decisions in a single module.

The paper is organized as follows. The control architecture is presented in section II. Section III showcases the changes that have been made to improve the nominal behavior. Then, section IV highlights the differences between the compared MPC variants. Finally, experimental results are described in section V before concluding.

## II. Architecture Overview

The nominal control scheme of the quadruped is shown in Fig. 2. It is a refined version of our previous work deployed on the Solo-12 quadruped [17] shown in Fig. 1, inspired by the pipeline proposed in [19]. The goal of this control architecture is to track a reference base velocity specified either by a user or a higher level controller. In the nominal scheme, based on this reference velocity and the estimated state of the base, a planner outputs the base trajectory and the desired locations of upcoming footsteps using heuristics. Polynomial interpolation is then used to guide swinging feet to their target positions on the ground. The MPC relies on a simplified centroidal model of the quadruped to find the contact forces that should be applied by the feet in stance phase for the base to follow as closely as possible the reference velocity over a prediction horizon. The whole-body control (WBC) translates desired contact forces and swinging feet trajectories into joint trajectories and feedforward torques. Finally, a proportional-derivative (PD) controller provides feedback torques based on the difference between the desired and current joint positions and velocities. Estimates of the base height, orientation, linear and angular velocities come from a cascade of two complementary filters that combine forward kinematics and inertial measurement unit (IMU) data [20]. While the general control architecture is kept the same, efforts have been made to improve the nominal behavior in terms of stability and robustness to enable to pursue the work started in [18] where several MPC variants were compared in simulation only. Formerly, base oscillations during motion hampered real-world deployments by impacting the consistency of desired contact forces and footsteps locations over a gait period. Reducing the amplitude of those oscillations was especially helpful for the stability of results. The approach we followed to this end is described in the next section.

## III. Reducing Base Oscillations

### A. Tasks for the whole-body control

Inverse kinematics (IK) with a full model of the quadruped is used in the WBC to compute command accelerations $\ddot{q}_{IK}$. The IK scheme is defined by 3 tasks:

- Follow the reference horizontal base linear velocity
- Keep the base orientation horizontal and follow the reference yaw angular velocity
- Follow the reference feet motion in the air and keep the feet in contact immobile

These tasks involve three frames, highlighted in Fig. 3. The position of the robot in world frame $o$ results from the integration of its reference velocity over time. The horizontal frame $h$ has its origin located at the center of the base, with only a rotation in yaw w.r.t the world frame $o$ to point in the forward direction of the robot. The base frame $b$ is completely aligned with the trunk, with a tilt in roll and pitch w.r.t the horizontal frame. By stacking all tasks in the previous description order, a $18 \times 18$ Jacobian $J$ can be defined such that $\dot{x} = J\dot{q}$ then inverted outside singularities:

$$\begin{bmatrix} \dot{q}_{lin} \\ \dot{q}_{ang} \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = J^{-1}\dot{x} = \begin{bmatrix} {}^b\mathcal{R}_h & 0 & 0 & 0 & 0 & 0 \\ 0 & {}^b\mathcal{R}_h & 0 & 0 & 0 & 0 \\ -J_1^{-1} & J_1^{-1}[{}^b\mathcal{T}_1]_\times & J_1^{-1} & 0 & 0 & 0 \\ -J_2^{-1} & J_2^{-1}[{}^b\mathcal{T}_2]_\times & 0 & J_2^{-1} & 0 & 0 \\ -J_3^{-1} & J_3^{-1}[{}^b\mathcal{T}_3]_\times & 0 & 0 & J_3^{-1} & 0 \\ -J_4^{-1} & J_4^{-1}[{}^b\mathcal{T}_4]_\times & 0 & 0 & 0 & J_4^{-1} \end{bmatrix} \dot{x} \quad (1)$$

with ${}^b\mathcal{R}_h = \mathcal{R}(roll, pitch, 0)^T$ and ${}^b\mathcal{T}_i$ the position of the $i$-th foot in base frame. $\dot{q}$ is the time derivative of the configuration vector $q$ (6D base + 12 joints) while $\dot{x}$ is the time derivative of the state vector in task space. With $x^{des}$, $\dot{x}^{des}$, $\ddot{x}^{des}$ the

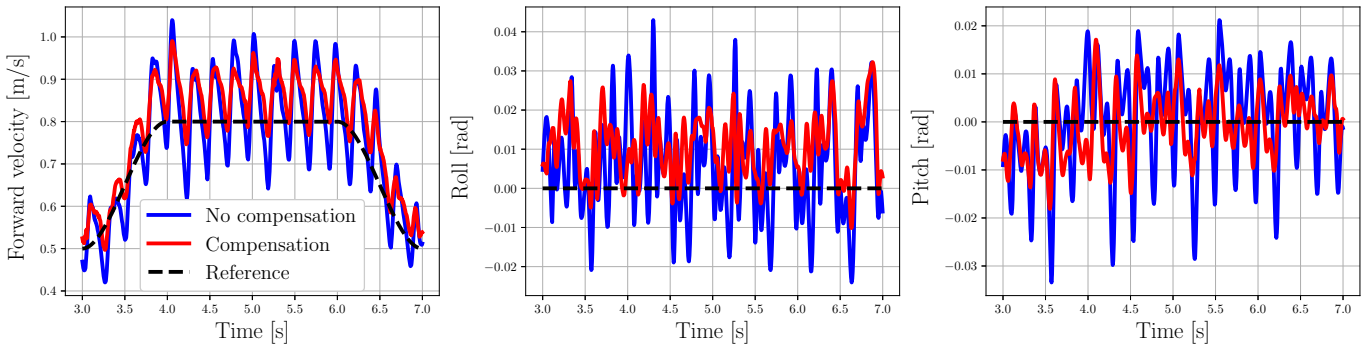



Fig. 4: The compensation of the contact forces reduces the oscillation in forward velocity, roll and pitch.

stacked desired positions, velocities and accelerations of all tasks and $K_p$, $K_d$ their position and velocity feedback gains, the command positions, velocities and accelerations can be computed as follows:

$$q^{cmd} = q_{k-1}^{cmd} + J^{-1}\left(x^{des} - x\right) \quad (2)$$

$$\dot{q}^{cmd} = J^{-1}\,\dot{x}^{des} \quad (3)$$

$$\ddot{q}_{IK} = J^{-1}(\ddot{x}^{cmd} - \dot{J}\dot{q}) \quad (4)$$

$$\ddot{x}^{cmd} = K_p(x^{des} - x) + K_d(\dot{x}^{des} - \dot{x}) + \ddot{x}^{des} \quad (5)$$

In practice a damped pseudo-inverse is used to avoid near-zero singular values that would lead to numerical instabilities when legs are near singularities [21]. It is important to note that this set of tasks is distinct from the one we deployed in [17] in the sense that position estimates of the robot base are no longer involved in the scheme. Tasks are now either expressed in base frame or in world frame in which the robot moves at the reference velocity. With current sensors (IMU + joint encoders), the absolute position in the world is not an observable quantity. Thus, by not using unobservable position estimates that come from a fusion of velocity estimates and forward geometry, we avoid injecting a slow unavoidable drift and noise into the control.

Thanks to this new formulation, the maximum velocity the robot could reach during experiments increased from less than 0.5 m/s up to 1.0 m/s. However, at high velocity, significant oscillations of the base both in orientation and linear velocity occurred and prevented from getting consistent results with the MPC variant that optimizes foot placements. A compensation term will be added to this end as described in the next section.

### B. Compensation term

As described in [17], for the computation of feedforward torques, a quadratic programming solver, which relies on relaxation variables $\delta_{\ddot{q}}$ and $\delta_f$, is used to find contact forces $f = f_{MPC} + \delta_f$ and accelerations $\ddot{q} = \ddot{q}_{IK} + \delta_{\ddot{q}}$, that are as close as possible to the force references provided by the MPC and the command accelerations computed by the IK, while taking into account the underactuated part of the dynamics.

$$\min_{\delta_{\ddot{q}}, \delta_f} \delta_{\ddot{q}}^T Q_1 \delta_{\ddot{q}} + \delta_f^T Q_2 \delta_f \quad (6)$$

$$\text{s.t.} \quad f_{MPC} + \delta_f \in \mathbf{K} \quad (7)$$

$$S(M(\begin{bmatrix}\ddot{q}_{IK,u}\\ \ddot{q}_{IK_a}\end{bmatrix} + \begin{bmatrix}\delta_{\ddot{q}}\\ 0\end{bmatrix}) + \mathrm{g} + \mathcal{C}) = SJ_c^T(f_{MPC} + \delta_f) \quad (8)$$

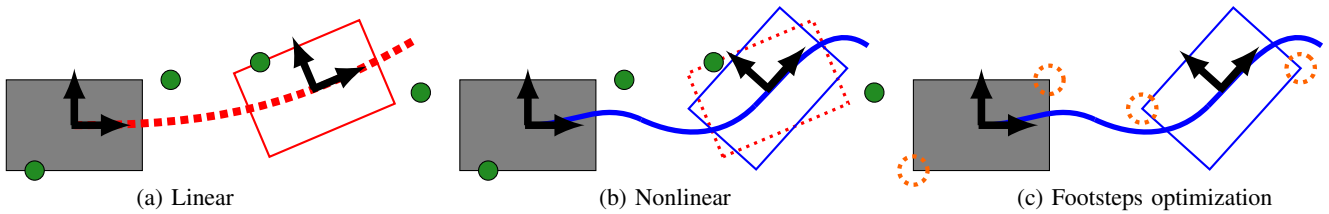
with subscripts $_u$ and $_a$ refering to the underactuated and actuated parts respectively, $M = \begin{bmatrix} Y & M_u \\ M_u^T & M_a \end{bmatrix}$ the generalized mass matrix, g the gravitational force, $\mathcal{C}$ the nonlinear forces, $S$ the matrix selecting the underactuated dynamics, $J_c$ the augmented contact Jacobian and $\mathbf{K}$ the friction cone linearized to the first order. As the MPC works with a centroidal model of the robot, it does not take into account the inertia effects that result from leg movements nor the nonlinear effects. Hence the contact forces computed by the MPC will not compensate or benefit from the forces related to these effects to stabilize the base and follow the reference velocity. As a result, while the left side of (8) includes the inertia of the base, the inertia of the joints, the nonlinear effects and the gravitational force, the MPC forces on the right side only take into account the inertia of the base and the gravitational force. If the inertia of the joints and the nonlinear effects are non-negligible, as it seems to be the case when the upper-leg joints are in motion at high speed, the QP will not work around an equilibrium point because the left and right sides of (8) may be widely different (requiring substantial $\delta_{\ddot{q}}$ or $\delta_f$ to respect the constraint). To limit this effect, a compensating term $f_{comp}$ is added to the contact forces of the MPC to diminish the offset between both side of the equation, so that the QP starts working closer to the equilibrium and $(\delta_{\ddot{q}},\ \delta_f)$ are lower. Instead of considering $f_{MPC} + \delta_f$, we consider $f_{MPC} + f_{comp} + \delta_f$ with:

$$f_{comp} = (J_c^T)^{\dagger}\,(\mathcal{C} + M_u \ddot{q}_{IK,a}) \quad (9)$$

where $M_u\ddot{q}_{IK,a}$ accounts for the effect of joints inertia on the base dynamics. As seen in Fig. 4, adding this compensation term resulted in a reduction of the oscillation of the linear velocity by a factor of roughly 2.

## IV. MPC VARIANTS

As locomotion decisions must be taken by considering the future evolution of the system [2], a wide range of quadruped controllers leverages a MPC to generate the motion in real time by predicting the behavior of the robot over a prediction horizon. Then, a WBC converts those decisions into actuator commands to follow the movement. MPCs usually exploit a reduced model of the dynamics to limit the computational complexity. The choice of reduced model is often ad-hoc or guided by intuition. Since quadruped robots tend to have lightweight limbs, most of their mass is localized in their



(a) Linear (b) Nonlinear (c) Footsteps optimization

Fig. 5: Summary of the differences between MPC variants. The front-left and hind-right feet are first in contact, with a switch to the front-right and hind-left feet later in the prediction horizon. Both linear (a) and nonlinear (b) variants use footsteps locations defined by heuristics (green dots) while the one optimizing footsteps locations (c) works around the predicted projections of shoulders on the ground (orange dotted circles). (a) relies on the reference trajectory of the CoM (red dotted line) while the two others use instead the predicted one (blue line) that results from the application of (10a).

trunk and, as such, centroidal dynamics [22] can provide an appropriate approximation of their whole-body dynamics. It describes the dynamics of the center of mass of the robot due to its interactions with the environment and corresponds to the under-actuated dynamics [23]:

$$m\ddot{p} = \sum_{i=1}^{n_c} f_i + mg \tag{10a}$$

$$\mathcal{I}\dot{\omega} + \omega \times (\mathcal{I}\omega) = \sum_{i=1}^{n_c} (r_i - p) \times f_i \tag{10b}$$

with $p$ the position of the CoM, $\omega$ the angular velocity of the body, $m$ the total mass of the robot, $\mathcal{I}$ its inertia matrix, and $g$ the gravity vector. $n_c$ is the number of 3D forces $f_i$ applied at the contact points $r_i$.

### A. Reduced dynamics models

A common choice is to use a centroidal model of the quadruped, with various levels of reduction. In our previous work [17], [18], pitch and roll velocities were supposed small so that $\mathcal{I}\dot{\omega} + \omega \times (\mathcal{I}\omega) \approx \mathcal{I}\dot{\omega}$. The trajectory of the center of mass $p$ was assumed to perfectly follow its reference $p^\star$. Foot placements $r_i$ were not part of the optimization problem but obtained beforehand based on Raibert heuristics [1], with terms similar to the ones used in [19] and noted $r_i^\star$:

$$r_i^\star = r_{sh} + \frac{T_{swing}}{2}\dot{q}_{lin} + k\,(\dot{q}_{lin} - \dot{q}_{lin}^\star) + \frac{1}{2}\sqrt{\frac{q_z}{g}}\dot{q}_{lin} \times \dot{q}_{ang}^\star \tag{11}$$

with $r_{sh}$ the projection of shoulders on the ground, $T_{swing}$ the duration of swing phases, $k$ a feedback coefficient and $q_z$ the height of the trunk. With those assumptions, (10b) becomes:

$$\mathcal{I}\dot{\omega} = \sum_{i=1}^{n_c} (r_i^\star - p^\star) \times f_i \tag{12}$$

To study the relevance of these reductions and understand their influence on the controller capabilities, they were progressively lifted and evaluated in simulation [18].

A first simplyfing assumption can be lifted by considering the predicted trajectory $p$ instead of the reference one $p^\star$, which makes the problem no more linear due to the cross product between optimization variables $p$ and $f_i$:

$$\mathcal{I}\dot{\omega} = \sum_{i=1}^{n_c} (r_i^\star - p) \times f_i \tag{13}$$

Then, another assumption can be lifted by considering the location of footsteps $r_i$ as optimization variables of the optimal control problem instead of obtaining their reference $r_i^\star$ from heuristics:

$$\mathcal{I}\dot{\omega} = \sum_{i=1}^{n_c} (r_i - p) \times f_i \tag{14}$$

The 3 considered variants are described in Fig. 5 and Tab. I.

TABLE I: Differences between MPC variants

| MPC Variant | Footsteps Locations | CoM Traj. | $\mathcal{I}\dot{\omega} =$ |
|---|---|---|---|
| Linear | Heuristics | Reference | $\sum_{i=1}^{n_c}(r_i^\star - p^\star) \times f_i$ |
| Nonlinear | Heuristics | Predicted | $\sum_{i=1}^{n_c}(r_i^\star - p) \times f_i$ |
| Footsteps | Optimized | Predicted | $\sum_{i=1}^{n_c}(r_i - p) \times f_i$ |

### B. Optimal control problem

The optimal control problem can be written as follows:

$$\min_{\{x\},\{f\},\{r\}} \sum_{t=0}^{T} \ell_t(x_t, f_t | r_t) + \ell_T(x_T)$$

$$\text{s.t.} \quad \forall t \quad x_{t+1} = \mathcal{H}(x_t, f_t | r_t) \tag{15a}$$

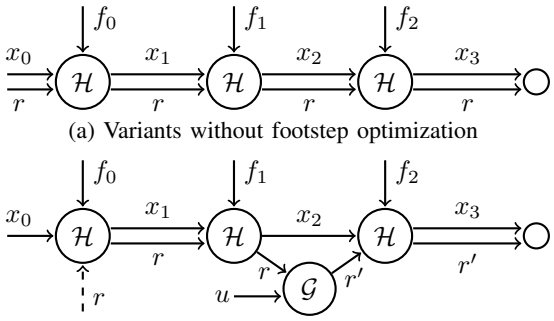
$$\forall t \quad x_t \in \mathcal{X} \tag{15b}$$

$$\forall t \quad f_t \in \mathbf{K} \tag{15c}$$

where $\ell_t$ and $\ell_T$ are respectively the running and terminal cost. $\{x\}$, $\{f\}$ and $\{r\}$ are the decision variables discretized at the optimization nodes indexed by $t$. The state vector $\{x\}$ includes the position, orientation, linear and angular velocities of the base. $\{x\}$ has to remain in the feasibility manifold $\mathcal{X}$ to ensure that a valid whole-body movement that can achieve $x$ exists. The control vector $f$ contains the 3D forces at each contact point, constrained by the friction cone $\mathbf{K}$. $\{r\}$ stores the position of footsteps which are optimization variables only for the last MPC variant. $\mathcal{H}$ enforces the system dynamics, that is (10a) and either (12), (13) or (14) depending on the variant. For the last one, the dynamics can be written as follows, with $I_6$ the identity matrix of size 6, $\Delta t$ the time step between nodes and $[r_i - p]_\times$ a skew-symmetric 3 by 3 matrix representing the cross product in (10b) as a matrix multiplication:

$$\mathcal{H}(x_t, f_t | r_t) = Ax_t + B(x_t, r_t)f_t \tag{16a}$$

$$A = \begin{bmatrix} I_6 & \Delta t I_6 \\ 0_6 & I_6 \end{bmatrix} \tag{16b}$$




(a) Variants without footstep optimization

(b) Variant with footsteps optimization

Fig. 6: Factor graph displaying the correlations between the decision variables of the MPC variants. Example with a prediction horizon of 3 time steps and a contact switch occurring at the 2nd time step.

$$B = \begin{bmatrix} \dots & \frac{\Delta t^2}{m} I_3 & \dots \\ \dots & \Delta t^2 \mathcal{I}^{-1} [r_i - p]_\times & \dots \\ \dots & \frac{\Delta t}{m} I_3 & \dots \\ \dots & \Delta t \mathcal{I}^{-1} [r_i - p]_\times & \dots \end{bmatrix} \tag{16c}$$

$r_i$ is replaced by $r_i^\star$ in (16c) for the variants (12) and (13) that do not optimize footsteps location and $p$ is replaced by $p^\star$ for the first variant (12). The $i$-th column of B is disabled when the $i$-th foot is not in contact phase.

The footsteps positions are not optimization variables for the first two variants, so the state and control vectors remain the same over the whole prediction horizon ($\{x\}$ and $\{f\}$ respectively). As a result the OCP scheme is straightforward with a series of similar nodes connected to each other to enforce the dynamics $\mathcal{H}(x_t, f_t | r_t)$ at each time step with a terminal node at the end, as depicted in Fig (6a).

The last variant is a parametric OCP that we have to reorganize to cast it under a form that our Differential Dynamic Programming (DDP) solver can handle. The footsteps are added as plain state variables whose values can only be changed at impact time, as shown in Fig (6b). This is implemented as a specific dynamic function inserted in the time line at the beginning of each contact phase:

$$r_{t+1} = \mathcal{G}(\Delta r_t | x_t, f_t) = r_t + \Delta r_t \tag{17}$$

where $\Delta r_t$ is the step length taken by the corresponding foot during the previous flying phase. The size of $\Delta r_t$ depends on the number of contacts that are modified.

### C. Running and terminal costs

Four running costs are shared by all variants:

- quadratic cost $\ell_x$ on the error between predicted and desired state vectors to track the desired state trajectory

$$\ell_x = \|x^\star - x\|^2 \tag{18}$$

- quadratic cost $\ell_f$ on the norm of ground reaction forces to minimize them if possible (regularization)

$$\ell_{f,i} = \|f_{z,i} - \frac{mg}{n_c}\|^2 \tag{19}$$

- barrier cost $\ell_{\mathbf{K}}$ to avoid slipping by enforcing friction cone constraint $\forall t, f_t \in \mathbf{K}$

$$\begin{aligned} \ell_{\mathbf{K},i} = &\|(f_{x,i} - \mu f_{z,i})^+\|^2 + \|(-f_{x,i} - \mu f_{z,i})^+\|^2 \\ &+ \|(f_{y,i} - \mu f_{z,i})^+\|^2 + \|(-f_{y,i} - \mu f_{z,i})^+\|^2 \\ &+ \|(-f_{z,i})^+\|^2 + \|(f_{z,i} - f_{z,max})^+\|^2 \end{aligned} \tag{20}$$

- barrier cost $\ell_{kin}$ to enforce kinematic limits on the distance between shoulders and their associated foot. That way, contact forces do not lead to an unfeasible motion for the whole-body control ($\forall t, x_t \in \mathcal{X}$)

$$\ell_{kin,i} = \| \left( \|sh_i - r_i\|^2 - d_{lim}^2 \right)^+ \|^2 \tag{21}$$

with $\{\cdot\}^+ = max(\{\cdot\}, 0)$, $\mu$ the friction coefficient, $sh_i$ the position of the $i$-th shoulder, $d_{lim}$ a limit distance (80% of the leg limit). Since constraints are enforced through a quadratic penalization using $\{\cdot\}^+$, there is no guarantee that they will be respected. In practice, with a small margin for $\mu$, this approximation works well and no slipping occurs. For the last variant, contrary to [18] which applied a quadratic cost on the distance between $r_i$ and $r_i^\star$, so that the optimization was done around the heuristic locations, here only the distance between the foot location and the shoulder is penalized to be completely heuristic-free. This cost $\ell_{r,i}$ can be seen as a form of regularization so that, on average, contacts are centered on the shoulders projection, as they are using Raibert's heuristic.

$$\ell_{r,i} = (sh_{x,i} - r_{x,i})^2 + (sh_{y,i} - r_{y,i})^2 \tag{22}$$

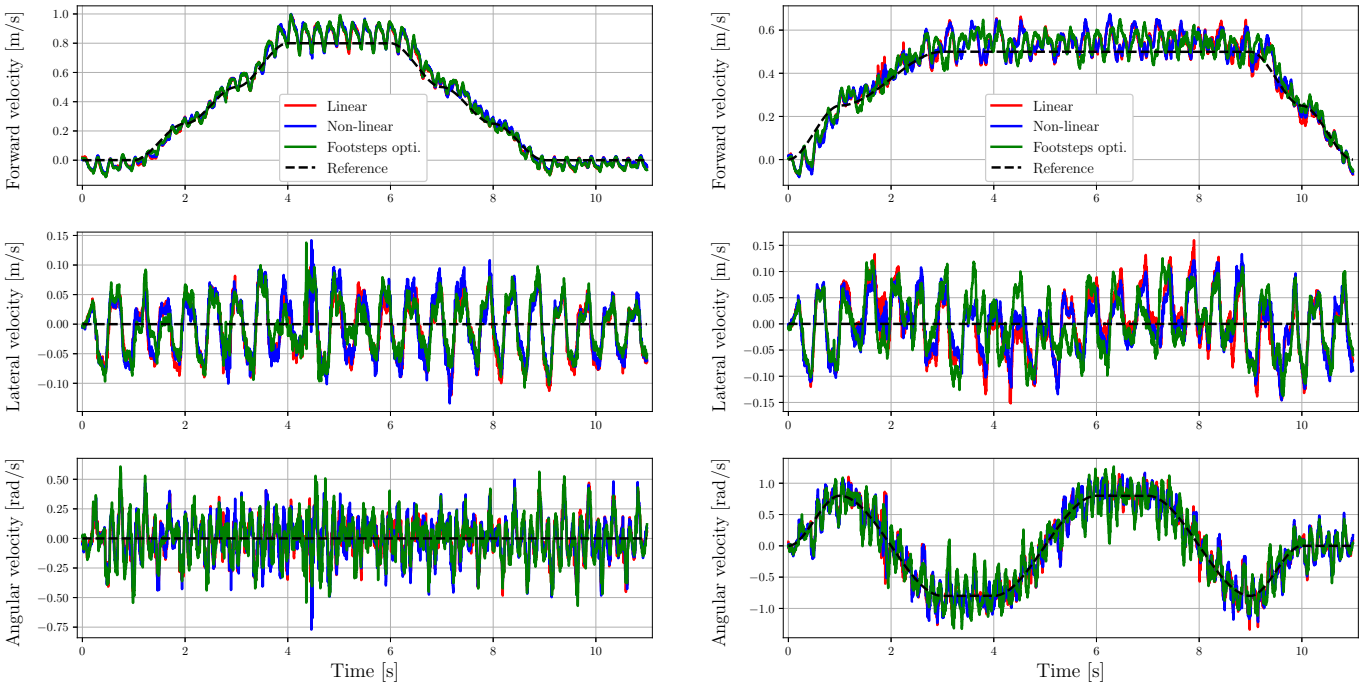
The last node of the OCP has no command so the terminal costs in the current formulation are only $\ell_x$, $\ell_{kin,i}$ and $\ell_{r,i}$. The nodes responsible for footsteps optimization have no running costs but they still act on the global cost value through the running costs $\ell_{kin,i}$ and $\ell_{r,i}$ of the other nodes.

## V. EXPERIMENTAL EVALUATION

Experiments were conducted to compare the performances of the three MPC variants based on the improved control architecture which made possible their real-world deployment.

### A. Experimental setup

Experiments were first performed indoors on a flat carpet-like material. Ground truth was retrieved thanks to a motion capture system comprising 20 infrared cameras spread around the workspace that track 13 reflective markers, installed on top of the robot base, at 200 Hz. During the experiments the robot was powered via an external power supply. Communications with the robot (sensors data retrieval and command sending) were done using an Ethernet link to the control desktop computer. Out of the prototyping phase, all control blocks were converted from Python to C++ for computational efficiency, except for the main loop which calls them, which allowed it to go from 500 Hz to 1 kHz. MPCs were implemented using Crocoddyl [24] as in [18]. For real-time purpose, they all run in a parallel process called at 50 Hz. Desired contact forces are retrieved after a delay due to the solving time ($\approx$ 2 ms for the linear and nonlinear MPCs, $\approx$ 6 ms for the footsteps MPC). The QP problem in the WBC is solved with OSQP [25]. The MPC weights chosen for position, orientation,




(a) First scenario (b) Second scenario

Fig. 7: Forward, lateral and angular velocity profiles for the two scenarios of Fig. 8.

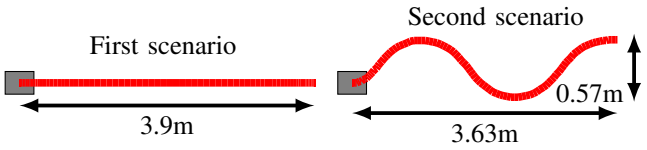



Fig. 8: Top view of the robot trajectory in the two considered scenarios obtained by integration of the reference velocity.

linear velocity and angular velocity errors are respectively $[2.0, 2.0, 10, 0.25, 0.25, 10, 0.2, 0.2, 0.2, 0, 0, 0.3]$. The weights for contact force regularization were set to $5 \times 10^{-5}$ for all components. To perform inverse kinematics we used $K_p = 10$, $K_d = 2\sqrt{K_p} = 6.3$ and a weight of 1 for all tasks. For the QP problem (6) we used $Q_1 = 0.1I_6$ and $Q_2 = 10I_{12}$ for the weights of the acceleration and contact force relaxation variables. For the on-board impedance controller, all joints shared the same proportional and derivative feedback control gains of 3 Nm/rad and 0.3 Nm/(rad/s) respectively. The performed gait was a trot with a period of $0.48s$ as it proved to be a good trade-off for evaluating the MPC performances. A faster gait would be naturally more stable due to the faster switching between diagonally opposed pairs of contacts, thus making the MPC role less crucial. A slower gait proved to be harder to stabilize because the base can tilt too much during a single swing phase, which can harldy be corrected (with two contact points we can only act along an axis). Deployment of the last MPC variant on the robot was made possible by the reduction of velocity oscillations due to the compensating contact forces described in Section III-B: they improved the consistency of the footsteps optimization which was previously diverging. As the estimated velocity of the base influences footsteps positions over the prediction horizon, the smaller the oscillations, the less these positions are modified over the span of a gait period.

### B. Results

**Indoor tests:** Performances are compared for the two scenarios shown in Fig. 8. During the first scenario the quadruped goes straight forwards. The velocity command is slowly increased during 4 seconds, stays at 0.8 m/s during 2 seconds, then goes back down to 0 m/s in 4 seconds. During the second scenario the quadruped performs several turns in a row. The velocity command goes up to 0.5 m/s forwards with $\pm 0.8$ rad/s along the vertical axis to get a S-shaped trajectory. Polynomial interpolation generates command profiles that are continuous both in velocity and acceleration. Motion capture data is reported in Fig. 7. Linear and non-linear variants lead to very close behaviours over the whole movement in both scenarios. Differences with the variant that optimizes footsteps location are noticeable but the values and amplitudes of errors and oscillations with respect to the references are roughly the same. The oscillations of the forward velocity around its reference have a maximum amplitude of around 0.15 m/s during the high-velocity phase of the first scenario. Lateral velocity tracking seems stable during both tests with an amplitude of roughly 0.1 m/s, even when turning in scenario 2, and with a shift of the average value toward the outside of the turns. For the considered angular velocities (up to $\pm 0.8$ rad/s), turning does not impact forward velocity tracking in a noticeable way. Joint torques estimated through current measurements peak at 2.1 Nm during the high-velocity phase. Actuators can deliver up to 2.5 Nm at 12 A, so hardware capabilities are not fully exploited yet [26]. There is still way to improve the control architecture and reach higher velocities.

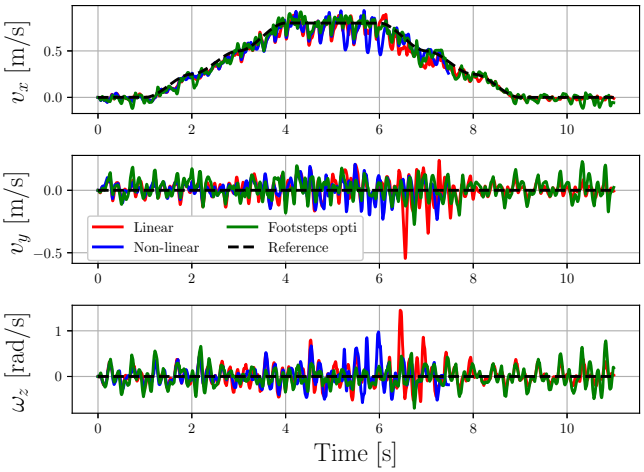



Fig. 9: Forward, lateral and yaw velocities during the outdoor tests. Due to a lack of motion capture system outdoors, represented quantities are the estimated ones.

Fig. 10: Trotting gait on wet grass. Connection with the robot was done by Wifi with the same computer than we used indoors. Previous experiments showed that packet loss did not hamper the behavior for distance under 25 m.

**Outdoor test:** In complement to indoor locomotion, the different controllers were tested on wet grass (Fig. 10). The quadruped was powered by 2 on-board batteries, one at the front, one at the back, that were not included in the model. Their weight of 100 g acted as an additional perturbation. The robot managed to follow the velocity profile up to 0.8 m/s without falling with each MPC variant. The lack of contact detection on this wet and bumpy surface resulted in numerous foot slipping when the robot tried to apply forces with a foot that had not properly landed. The most notable slipping occurs at 6.6 s for the linear MPC (Fig. 9), with peak lateral and angular velocities of -0.5 m/s and 1.3 rad/s respectively. Yet it is not a sign that this MPC tends to slip more than the others, but is rather due the slight differences in foot placements across experiments. The terrain also leads to larger oscillations around the references compared to indoors, especially during the high-velocity phase. Currently, neither the footsteps heuristics of the first two variants nor the footsteps optimization of the last one can handle a non-flat ground (slope, stairs). They could be extended either by using privileged information about the environment or by implementing some sort of online slope detection. Friction cone constraints would have to be refactored as well to match non-horizontal surfaces, hence why performances were only compared on a flat ground. Moreover, applying a repeatable disturbance to a walking quadruped is not trivial in practice, contrary to simulation. Slight differences in direction, strength or duration of the push can lead to widely different behaviours, especially depending on the gait status. If it happens at the beginning of a swing phase, the controller can directly react and adjust footsteps positions accordingly. However, if it occurs near touchdown, then it is too late to widely modify contact location. For these reasons, robustness to disturbances was not compared during our experiments. Videos of both indoor and outdoor tests can be found online[1].

**Discussion:** Our first attempt in designing the whole-body control block of this architecture was to implement an inverse dynamics (ID) that could directly handle both position and velocity for the base and the feet, as well as ground reaction forces, as in simulation [18]. However, using this approach we did not achieve a stable behavior on the real robot. This could be due to the sensitivity of ID to mismatches between model and hardware. Moreover, Solo-12 is a lightweight quadrupedal robot with almost direct-drive actuators. As studied in [27], coupling among links is more significant with the fast dynamics of such actuators. So as we did not manage to run the ID faster than 500 Hz, it might have been insufficient to avoid instability. For this reason, we decided to use instead the approach proposed in [19], that includes an IK to compute the joint accelerations to perform the position and velocity tasks (1-5), based on which a QP problem is solved to find a compromise between tracking these joint accelerations and taking into account MPC decisions and the equation of dynamics (6-8). This QP problem has to balance decisions that might be conflicting since the instantaneous decisions of the IK are done without knowing what has been decided by the MPC that works on a prediction horizon. Finding the right balance is not trivial, as mainly relying on IK ($Q_1 \gg Q_2$) would remove the predictive aspect of the architecture, whereas mainly relying on the MPC ($Q_2 \gg Q_1$) would instead hamper the feet tracking tasks. Both simulations and experimental tests show that the implementation of the three MPC variants leads to quite similar behaviours. This result does not seem to confirm the one of our previous simulation study [18]. We suppose that, with the prior architecture, the footstep optimization was really beneficial to the controller and allowed the quadruped to go faster in simulation. Now that the whole architecture has been refined and the ID replaced by an IK+QP scheme, the margin for behavioural improvement has been reduced, so the gains of using nonlineary and footsteps optimization has been somehow smoothed out, at least for the considered velocities. The quadruped falls at higher velocities due to the legs reaching singularity during motion. Our controller does not handle flight phases yet (no feet in contact) so the velocity the robot can reach is limited by the gait frequency and the dimensions of the legs. A key result is to have shown that a centroidal MPC which optimizes footsteps location online can be successfully deployed on our quadruped robot, which

[1]https://gepettoweb.laas.fr/articles/leziart2022.html

was an important objective following [18]. The question of optimizing all the variables in the same MPC or treating them in separate blocks arises. On the one hand, modularity can be preferred with footsteps decisions that are independent from the computation of contact forces by the MPC. With the first and second variants, the planner with simple heuristics could be replaced by another control system to explore new paradigms. It could go from a neural network that implicitly learns adaptive heuristics during its training [28] to more path-planning oriented approaches that leverage information about the environment to place the feet at the best locations [29]. Gait type and period could be tuned as well since the MPC does not have any notion of gait per se. On the other hand, all-in-one optimization with the third variant avoids the need to formulate heuristics and let the possibility to tackle more extensive and difficult challenges, such as an optimization of footsteps timings as well [18].

## VI. CONCLUSION

We presented key improvements to the nominal control architecture of Solo-12 [17], based on which three variants of a centroidal MPC previously tested in simulation were implemented and compared on the real robot. Efforts were made to refine our previous control architecture and reduce oscillations of the base, which were disrupting the consistency of foot placement decisions. These modifications allowed real-world deployment and increased the maximum velocity up to 1 m/s. The central result is to have shown that comparable performances can now be obtained both in simulation and on real hardware with all MPC variants. However, this does not confirm the preeminence of the nonlinear centroidal MPC scheme optimizing footholds over a modular scheme that uses a heuristic for foot placement and relies on a simplified model, that was previously observed in simulation [18]. In view of the modifications that were made to implement these controllers on the real robot, the reasons for this difference have been thoroughly discussed. We also discussed the interest of using a modular architecture versus a more global optimization scheme. Going further, we plan to extend our approach towards a whole-body MPC in order to exploit the complete dynamical model on the prediction horizon while being aware of its kinematic limits for footstep placements. We also plan to work on feet contact detection to handle contact timing mismatches, increase the robustness on rough terrain and include flight phases into the gait.

## REFERENCES


[1] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[2] S. Kajita and B. Espiau, *Legged Robot*. Springer Handbook, 2008.

[3] F. Xi, R. Sinatra, and W. Han, "Effect of leg inertia on dynamics of sliding-leg hexapods," *J. Dyn. Sys., Meas., Control*, vol. 123, 2001.

[4] T. Bretl and S. Lall, "A fast and adaptive test of static equilibrium for legged robots," in *IEEE ICRA*, 2006.

[5] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *IJRR*, vol. 30, 2011.

[6] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics*, vol. 31, 2012.

[7] A. W. Winkler, C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Planning and execution of dynamic whole-body locomotion for a hydraulic quadruped on challenging terrain," in *IEEE ICRA*, 2015.

[8] S. Aine, S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev, "Multi-heuristic A*," *IJRR*, vol. 35, 2016.

[9] A. Hornung, A. Dornbush, M. Likhachev, and M. Bennewitz, "Anytime search-based footstep planning with suboptimality bounds," in *IEEE-RAS Humanoids*, 2012.

[10] M.-T. Shing and G. B. Parker, "Genetic algorithms for the development of real-time multi-heuristic search strategies." in *International Conference on Genetic Algorithms*, 1993.

[11] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, "A control architecture for quadruped locomotion over rough terrain," in *IEEE ICRA*, 2008.

[12] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds," *IEEE RA-L*, vol. 2, 2017.

[13] A. W. Winkler, F. Farshidian, M. Neunert, D. Pardo, and J. Buchli, "Online walking motion and foothold optimization for quadruped locomotion," in *IEEE ICRA*, 2017.

[14] C. Dario Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *IEEE/RSJ IROS*, 2017.

[15] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization," *IEEE RA-L*, vol. 3, 2018.

[16] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control," in *IEEE/RSJ IROS*, 2018.

[17] P.-A. Léziart, T. Flayols, F. Grimminger, N. Mansard, and P. Souères, "Implementation of a Reactive Walking Controller for the New Open-Hardware Quadruped Solo-12," in *IEEE ICRA*, 2021.

[18] T. Corbères, T. Flayols, P.-A. Léziart, R. Budhiraja, P. Souères, G. Saurel, and N. Mansard, "Comparison of predictive controllers for locomotion and balance recovery of quadruped robots," in *IEEE ICRA*, 2021.

[19] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv:1909.06586*, 2019.

[20] J. Carpentier, M. Benallegue, N. Mansard, and J.-P. Laumond, "A kinematics-dynamics based estimator of the center of mass position for anthropomorphic system—a complementary filtering approach," in *IEEE-RAS Humanoids*, 2015.

[21] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, vol. 17, 2004.

[22] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, 2013.

[23] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning Feasibility Constraints for Multicontact Locomotion of Legged Robots," in *Robotics: Science and Systems*, 2017.

[24] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *IEEE ICRA*, 2020.

[25] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, 2020.

[26] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols, J. Fiene, A. Badri-Spröwitz, and L. Righetti, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE RA-L*, 2020.

[27] M. W. Spong, S. Hutchinson, M. Vidyasagar *et al.*, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[28] O. A. V. Magana, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini, "Fast and continuous foothold adaptation for dynamic locomotion through cnns," *IEEE RA-L*, 2019.

[29] D. Song, P. Fernbach, T. Flayols, A. Del Prete, N. Mansard, S. Tonneau, and Y. J. Kim, "Solving footstep planning as a feasibility problem using l1-norm minimization," *IEEE RA-L*, 2021.