Topology-Based MPC for Automatic Footstep Placement and Contact Surface Selection

Jaehyun Shim, Carlos Mastalli, Thomas Corbères
Steve Tonneau, Vladimir Ivan, Sethu Vijayakumar

***Abstract*—State-of-the-art approaches to footstep planning assume reduced-order dynamics when solving the combinatorial problem of selecting contact surfaces in real time. However, in exchange for computational efficiency, these approaches ignore limb dynamics and joint torque limits. In this work, we address these limitations by presenting a topology-based approach that enables model predictive control (MPC) to simultaneously plan full-body motions, torque commands, contact surfaces, and footstep placements in real time. To determine if a robot's foot is inside a polygon, we borrow the winding number concept from topology. Specifically, we use winding number and electric potential to create a contact-surface penalty function that forms a harmonic field. Using our topology-based penalty function, MPC can then select a contact surface from all candidate surfaces in the vicinity and determine footstep placements within it. We highlight the benefits of our approach by showing the impact of considering full-body dynamics, which includes joint torque limits and limb dynamics, in the selection of footstep placements and contact surfaces. Additionally, we demonstrate the feasibility of deploying our topology-based approach in an MPC scheme through a series of experimental and simulation trials.**

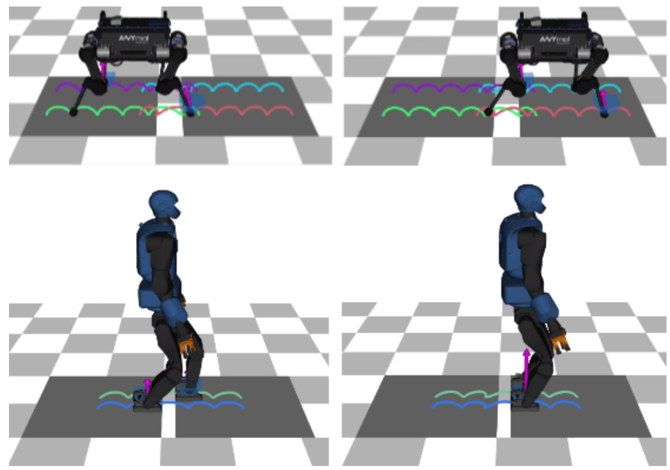

Fig. 1. Visualization of the planned motion and footsteps, taking into account the robot's full-body dynamics (including limb dynamics and joint torque limits), friction cones, and all potential contact surfaces. Different colors are used to represent the swing-foot trajectories, while candidate contact surfaces are indicated by dark gray squares. Our approach can be used to plan footstep placements and contact surfaces for both quadruped and humanoid robots. This video is available to view at `https://youtu.be/VpZX-YgHf34`.

I. Introduction

To *traverse discrete terrains* such as stepping stones, legged robots need to carefully plan their footsteps and motion [1], [2], [3]. In previous works, footsteps and motion were computed separately [4], [5], [6] to reduce the combinatorial complexity of these nonlinear problems [7], [8]. However, in doing so, assumptions need to be introduced into the gait pattern, kinematics, and dynamics model. Alternatively, to focus on the footstep planning (i.e., combinatorial problem), other approaches neglect limb dynamics, thereby allowing for formulating this problem as a mixed-integer convex problem [9], [10]. However, for robots with heavy limbs or limited actuation torque, this assumption does not hold, nor can the full reachability (i.e., kinematics) of the robot be exploited. These limitations also lead to errors in footstep tracking, which are caused by improper tracking of angular momentum [11]. The above-mentioned limitations can be addressed by considering the robot's full-body dynamics, as it also takes into account limb dynamics and actuation limits, and computing motion and footsteps together. However, this results in a large optimization problem that is challenging to solve within a control loop of a few milliseconds.

Recently, we have demonstrated an MPC that takes into account the full-body dynamics of the robot [11], [12]. A key advantage of our previous approaches is the ability of our MPC to generate agile and complex maneuvers through the use of feasibility-driven search [13] and optimal policy tracking. However, it requires a predefined sequence of footstep placements and contact surfaces. To address these limitations, in this paper, we borrow concepts from topology and classical electrostatics to enable the automatic selection of footstep placements and contact surfaces in real time. Compared to other state-of-the-art approaches, our footstep plans ensure torque limits, friction-cone constraints, and full-body kinematics and dynamics (Fig. 1). To the best of our knowledge, our work is the first to introduce the deployment of full-body dynamics MPC with optimizing footstep placement and contact surface.

Jaehyun Shim, Thomas Corbères, Steve Tonneau and Sethu Vijayakumar are with the School of Informatics, University of Edinburgh, U.K. (e-mail: jshim.robotics@gmail.com, t.corberes@sms.ed.ac.uk, stonneau@exseed.ed.ac.uk, sethu.vijayakumar@ed.ac.uk).

Carlos Mastalli is with the Institute of Sensors, Signals and Systems, School of Engineering and Physical Sciences, Heriot-Watt University, U.K. (e-mail: c.mastalli@hw.ac.uk).

Vladimir Ivan is with Touchlab Limited, U.K. (e-mail: v.ivan.mail@gmail.com).

This research was conducted as part of the Memory of Motion (Memmo) project, a collaborative project supported by the European Union within the H2020 Program (Grant Agreement No. 780684).

A. Related Work

Recent methods for footstep planning can optimize the footstep placement and gait pattern [9], [14], [15], [16]. These approaches can overcome discrete terrains by smoothing their geometry [7], [14], [15] or using discrete variables in mixed-integer optimization [9], [10], [16], [17]. However, the computational complexity of these approaches makes it infeasible to deploy them online. The reason is that they result in a combinatorial explosion of hybrid modes, which cannot be resolved by employing simplified models to cast the problem as a mixed-integer convex optimization [9], [17]. Alternatively, it can be formulated as a continuous problem with complementary constraints. However, the increased computation time due to the ill-posed nature of the complementary constraints makes it difficult to deploy this approach online [7], [8], [14]. To circumvent combinatorial complexity or ill-conditioning, our work concentrates solely on *optimizing the footstep placement*.

The narrow focus on optimizing the footstep placement enables online re-planning. For instance, as shown in [18], MPC can generate a walking motion with automatic footstep placement. However, despite the impressive achievement, this approach assumes that the robot behaves as a linear inverted pendulum, which cannot account for its kinematic feasibility, actuation limits, orientation, or the effect of non-coplanar contact conditions, and limb and vertical motions. Later, Di Carlo et al. [19] proposed a convex relaxation of the single rigid body dynamics (SRBD) that can accommodate the robot's orientation. As in [18], this boils down to a linear MPC that is solved with a general-purpose quadratic programming solver. More recently, other MPC approaches employ SRBD or centroidal dynamics (CD), using direct transcription and general-purpose nonlinear programming solvers [20], [21]. Although these approaches can address non-coplanar contacts and vertical motion, they still cannot account for the robot's kinematic and actuation limits, and the effects of the limb dynamics. Furthermore, one may think that these assumptions are not critical for robots with lightweight legs; however, a recent study shows that they still have a substantial impact on the control [22]. This justifies why, in our work, we compute motions that *ensure the robot's full-body dynamics*.

Real-time handling of both body and leg kinematics and dynamics in an MPC manner can be achieved by taking advantage of the temporal structure of the optimal control problem. Differential dynamic programming (DDP) [23] takes advantage of this structure by factorizing a sequence of smaller matrices, rather than employing sparse linear solvers [24] commonly done in nonlinear programming [25]. This reduction in computational complexity facilitates the deployment of MPC with full-body dynamics, as demonstrated by simulation results in [26]. Inspired by these results, recent research has shown the application of MPC with SRBD and full kinematics [27], [28], [29] and full-body dynamics [11], [12], [30], [31], [32]. However, these approaches do not address the issue of footstep placement and contact surface selection. While algorithms based on DDP, such as iLQR [33], BOX-FDDP [13], and ALTRO [34], have been also developed, they have not yet been applied to address the aforementioned problems. In contrast to other MPC approaches, our topology-based approach *creates a continuous cost function for terrain* within the framework of full-body dynamics MPC, rather than addressing the combinatorial complexity of selecting an optimal contact surface.

B. Contribution

The main contribution of our work is a motion predictive controller that simultaneously plans full-body motions, torque commands, feedback policies, footstep placements, and contact surfaces in real time. Specifically, we identify three technical contributions as follows:

(i) A novel topology-based MPC using the winding number and electric field that plans footstep placements and contact surfaces in real time;
(ii) Demonstration of the advantages of our topology-based MPC that takes into account full-body dynamics in selecting footstep placements and contact surfaces;
(iii) Experimental validation of our topology-based MPC on the ANYmal robot, and simulations that showcase its potential capabilities.

In the next section, we introduce the concepts of electric potential and winding number that we use to create a *contact-surface penalty function*.

II. ELECTRIC POTENTIAL AND WINDING NUMBER IN CONTACT SURFACES

In this section, we explain in detail our novel approach to footstep placement and contact surface selection. It borrows the concepts of electric potential (Section II-A and II-B) and winding number (Section II-C and II-D) from classical electrostatics and topological representation, respectively. These concepts enable us to design a contact-surface penalty function, allowing our MPC to select optimal contact surfaces and footstep placements (Section II-E).

A. Unitless Electric Potential in Closed Curves

We begin with an introduction to the electric potential that arises from a charged particle. This is also called a Coulomb potential and is defined as

$$V_E(\mathbf{r}) = \frac{Q}{4\pi\varepsilon_0}\frac{1}{|\mathbf{r}-\mathbf{p}|}, \tag{1}$$

where Q is the charge of the particle, ε_0 is the permittivity of vacuum, $\mathbf{r}$ is the point at which the potential is evaluated, and $\mathbf{p}$ is the point at which the charged particle is located. Since the physical meaning of the electric potential is not relevant to the definition of the cost function, we drop the scaling factor $\frac{Q}{4\pi\varepsilon_0}$ and make the *potential unitless*. Moreover, the potential measured across a closed curve $\boldsymbol{\gamma}(s)$ can be obtained by integrating the electric field:

$$V_E(\boldsymbol{\gamma}) = \int \frac{1}{|\boldsymbol{\gamma}(s)-\mathbf{p}|}ds. \tag{2}$$

B. Electric Potential in Contact Polygons

We describe the contact surfaces as contact polygons. Their boundaries can be represented by a sequence of linear segments. Similar to Eq. (2), we calculate the potential measured across the i-th linear segment of the polygon as follows:

$$p_i = \int_0^1 \frac{1}{|\mathbf{a}_i + (\mathbf{b}_i - \mathbf{a}_i)s - \hat{\mathbf{p}}|} ds, \tag{3}$$

where $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^2$ are the endpoints of the segment and $\hat{\mathbf{p}} \in \mathbb{R}^2$ is the position of the robot's foot in the *horizontal plane*. Then, we compute the resulting potential produced by each potential segment p_i, which is described as follows:

$$V_{\boldsymbol{\gamma}} = -\sum_i p_i, \tag{4}$$

where the negative sign denotes the computation of the potential *outside* the contact polygon. Furthermore, the integral in Eq. (3) has the following analytical solution:

$$p_i = \frac{\operatorname{atan}(\frac{c_i}{e_i}) - \operatorname{atan}(\frac{d_i}{e_i})}{e_i} \tag{5}$$

with

$$\begin{aligned} c_i &= (\mathbf{a}_i \cdot \mathbf{b}_i) - (\mathbf{a}_i \cdot \hat{\mathbf{p}}) + (\mathbf{b}_i \cdot \hat{\mathbf{p}}) - (\mathbf{b}_i \cdot \mathbf{b}_i), \\ d_i &= -(\mathbf{a}_i \cdot \mathbf{b}_i) - (\mathbf{a}_i \cdot \hat{\mathbf{p}}) + (\mathbf{b}_i \cdot \hat{\mathbf{p}}) + (\mathbf{a}_i \cdot \mathbf{a}_i), \\ e_i &= [\mathbf{a}_i \times \mathbf{b}_i]_z - [\mathbf{a}_i \times \hat{\mathbf{p}}]_z + [\mathbf{b}_i \times \hat{\mathbf{p}}]_z, \end{aligned}$$

where $[\mathbf{a} \times \mathbf{b}]_z$ returns the z coordinate of the cross product of the two vectors $\mathbf{a} = [a_x, a_y]$ and $\mathbf{b} = [b_x, b_y]$ in the XY plane (i.e., $a_x b_y - a_y b_x$).

C. Winding Number

To represent discrete terrain as a topological space, we leverage the concept of *winding number*. The winding number is a measure of how many times a curve is wound around a point in the 2D plane. There are different ways to define the winding number. For instance, from the perspective of differential geometry, we can associate this number with the polar coordinates for a point in the origin ($\hat{\mathbf{p}} = \mathbf{0}$), i.e.,

$$\operatorname{wind}(\boldsymbol{\gamma}, \hat{\mathbf{p}} = \mathbf{0}) = \frac{1}{\pi} \oint_{\boldsymbol{\gamma}} \left(\frac{x}{r^2} dy + \frac{y}{r^2} dx \right),$$

where x, y defines the parametric equation of a continuous closed curve $\boldsymbol{\gamma}$, with $r^2 = x^2 + y^2$.

D. Winding Number in Contact Polygons

As in Section II-B and using the formula derived by [35], we calculate the *winding number* of the closed curve $\boldsymbol{\gamma}$ by summing over the i-th linear segments as follows:

$$\operatorname{wind}(\boldsymbol{\gamma}, \hat{\mathbf{p}}) = \sum_i^{\boldsymbol{\gamma}} \frac{\operatorname{atan2}(c_i, d_i)}{2\pi} \tag{6}$$

with

$$\begin{aligned} c_i &= [(\mathbf{a}_i - \hat{\mathbf{p}}) \times (\mathbf{b}_i - \hat{\mathbf{p}})]_z, \\ d_i &= (\mathbf{a}_i - \hat{\mathbf{p}}) \cdot (\mathbf{b}_i - \hat{\mathbf{p}}). \end{aligned}$$

Then, we only need to check if $\operatorname{wind}(\boldsymbol{\gamma}, \hat{\mathbf{p}}) \geq \frac{1}{2}$ to know if a point $\hat{\mathbf{p}}$ is inside a closed curve.

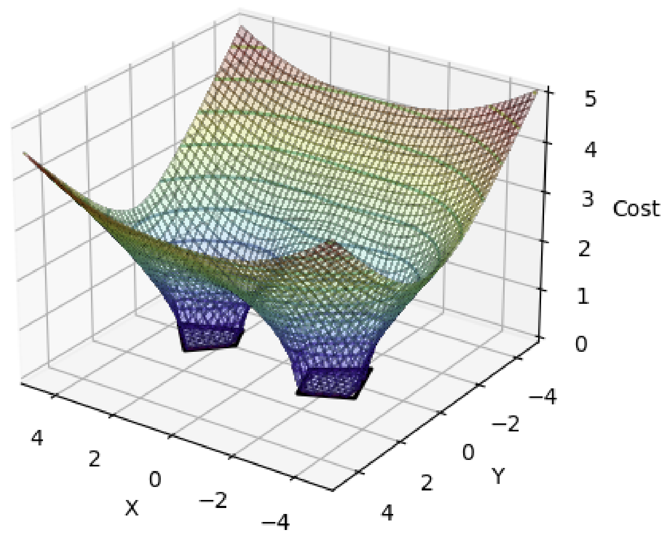

Fig. 2. Contact-surface penalty function for two contact surfaces. The cost is zero inside the two candidate contact surfaces outlined in black and positive elsewhere. To determine if the robot's feet are inside the contact surfaces, we compute the winding number.

E. Contact-Surface Penalty Function

To enable the robot to select a placement $\mathbf{p}_{\mathcal{C}_k}$ (with $\mathcal{C}_k$ as each foot) within a single contact surface, we assign a zero cost when the placement is inside the surface. To achieve this, we use an indicator function based on the winding number to compute the *contact-surface penalty* cost $\ell_{\mathbf{p}_{\mathcal{C}_k}}$ as follows:

$$\ell_{\mathbf{p}_{\mathcal{C}_k}} = \begin{cases} 0 & \text{if } \sum_j V_{\boldsymbol{\gamma}_j} = 0, \\ \sqrt{\frac{\sum_j N_j}{\sum_j V_{\boldsymbol{\gamma}_j}}} & \text{if } \sum_j \operatorname{wind}(\boldsymbol{\gamma}_j, \hat{\mathbf{p}}_{\mathcal{C}_k}) < \frac{1}{2}, \\ 0 & \text{if } \sum_j \operatorname{wind}(\boldsymbol{\gamma}_j, \hat{\mathbf{p}}_{\mathcal{C}_k}) \geq \frac{1}{2}, \end{cases} \tag{7}$$

where $V_{\boldsymbol{\gamma}_j}$ is the electric potential, N_j is the number of linear segments, and $\operatorname{wind}(\boldsymbol{\gamma}_j, \hat{\mathbf{p}}_{\mathcal{C}_k})$ is the winding number obtained for the polygon j. We compute this electric potential and winding number using Eq. (4) and Eq. (6). Finally, we can also compute the analytical derivatives of this penalty function by applying the chain rule.

As shown in Eq. (7), we define the penalty function by using a square root (second line). However, to avoid division by zero, we set the penalty function to zero when the electric potential is zero (first line). This happens when $\mathbf{p}_{\mathcal{C}_k}$ is on the boundary of the polygon. Fig. 2 shows our penalty function when we have two contact surfaces.

Since our penalty function is based on the definition of electric potential as defined in Eq. (5), it forms a harmonic field. For convex closed curves, it is also monotonic. Additionally, it remains harmonic for non-convex polygons, collections of polygons, self-intersecting curves, and overlapping polygons. This means that it contains the fewest possible number of local minima and saddle points. The topology-based representation via the winding number allows us to enforce all of these properties using our contact-surface penalty function. Inspired by [36], our penalty function exploits invariances within a homology class defined by the winding number. In other words, it captures containment, which makes this penalty an ideal candidate for numerical optimization.

III. MPC and Pipeline

In this section, we describe our MPC formulation for selecting footstep placements and contact surfaces (Section III-A). We then present the control pipeline and setup used in our experiments with the ANYmal robot (Section III-B).

A. Topology-Based MPC

Building upon our previous work [11], our MPC solves a hybrid optimal control problem at each control time step. The different modes of the hybrid dynamics define different contact conditions along the optimization horizon. These rigid contact conditions are subject to the robot's full-body dynamics (i.e., *contact dynamics*). We also model the contact-gain transitions between these modes using the *impulse dynamics*. Our Box-FDDP solver [13] then computes full-body motions, torque commands, and feedback policies within the robot's torque limits, given a predefined set of footstep placements.

Here, our contact-surface penalty function extends the capabilities of our previous MPC by enabling it to automatically plan footstep placements given a reference velocity and a set of discrete contact surfaces. In Eq. (8), the modifications introduced in this work are highlighted in blue.

$$\min_{\mathbf{x}_s,\mathbf{u}_s} \sum_{k=0}^{N-1} \left(\ell_k^{reg} + w_r\|\dot{\mathbf{r}}_k - \dot{\mathbf{r}}^{ref}\|^2 + w_f \sum_{\mathcal{C}_k} \ell_{\mathbf{p}_{\mathcal{C}_k}} \right)$$

$$\begin{aligned}
&\text{s.t. if } k \text{ is a contact-gain transition:}\\
&\quad \mathbf{q}_{k+1} = \mathbf{q}_k,\\
&\quad \begin{bmatrix}\mathbf{v}_{k+1}\\ -\boldsymbol{\Lambda}_{\mathcal{C}_k}\end{bmatrix} = \begin{bmatrix}\mathbf{M}_k & \mathbf{J}_{\mathcal{C}_k}^\top\\ \mathbf{J}_{\mathcal{C}_k} & \mathbf{0}\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{\tau}_{b_k}^{\mathcal{I}}\\ -\mathbf{a}_{\mathcal{C}_k}^{\mathcal{I}}\end{bmatrix}, && \text{(impulse dyn.)}\\
&\text{else:}\\
&\quad \mathbf{q}_{k+1} = \mathbf{q}_k \oplus \int_{t_k}^{t_k+\Delta t_k} \mathbf{v}_{k+1}\, dt,\\
&\quad \mathbf{v}_{k+1} = \mathbf{v}_k + \int_{t_k}^{t_k+\Delta t_k} \dot{\mathbf{v}}_k\, dt, && \text{(integrator)}\\
&\quad \begin{bmatrix}\dot{\mathbf{v}}_{k}\\ -\boldsymbol{\lambda}_{\mathcal{C}_k}\end{bmatrix} = \begin{bmatrix}\mathbf{M}_k & \mathbf{J}_{\mathcal{C}_k}^\top\\ \mathbf{J}_{\mathcal{C}_k} & \mathbf{0}\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{\tau}_{b_k}^{\mathcal{C}}\\ -\mathbf{a}_{\mathcal{C}_k}^{\mathcal{C}}\end{bmatrix}, && \text{(contact dyn.)}\\
&\log\left({}^{\mathcal{W}}\mathbf{p}_{\mathcal{G}_k,z}^{-1} \cdot {}^{\mathcal{W}}\mathbf{p}_{\mathcal{G}_k,z}^{ref}\right) = \mathbf{0}, && \text{(vertical foot pos.)}\\
&{}^{\mathcal{W}}\dot{\mathbf{p}}_{\mathcal{G}_k,z}^{-1} - {}^{\mathcal{W}}\dot{\mathbf{p}}_{\mathcal{G}_k,z}^{ref} = \mathbf{0}, && \text{(vertical foot vel.)}\\
&\mathbf{A}\,{}^{\mathcal{W}}\mathbf{p}_{\mathcal{G}_k,z}^{ref} = \mathbf{a}, && \text{(contact surface height)}\\
&\mathbf{C}\boldsymbol{\lambda}_{\mathcal{C}_k} \geq \mathbf{c}, && \text{(friction-cone)}\\
&\underline{\mathbf{x}} \leq \mathbf{x}_k \leq \bar{\mathbf{x}}, && \text{(state bounds)}\\
&\underline{\mathbf{u}} \leq \mathbf{u}_k \leq \bar{\mathbf{u}}, && \text{(control bounds)}
\end{aligned} \tag{8}$$

where $\mathbf{x} = (\mathbf{q}, \mathbf{v})$ is the state of the system, $\mathbf{q} \in \mathbb{SE}(3) \times \mathbb{R}^{n_j}$ (with n_j as the number of joints) is the joint configuration, $\mathbf{v} \in \mathbb{R}^{n_v}$ (with $n_v = 6 + n_j$) is the generalized velocity, $\mathbf{u} \in \mathbb{R}^{n_j}$ is the joint torque input, $\boldsymbol{\lambda}_{\mathcal{C}} \in \mathbb{R}^{n_c}$ (with n_c as the dimension of the contact forces) is the contact force, $\mathbf{p}_{\mathcal{C},\mathcal{G}} \in \mathbb{R}^{n_c}$ (with $\mathcal{C}$ and $\mathcal{G}$ as active and inactive contacts,

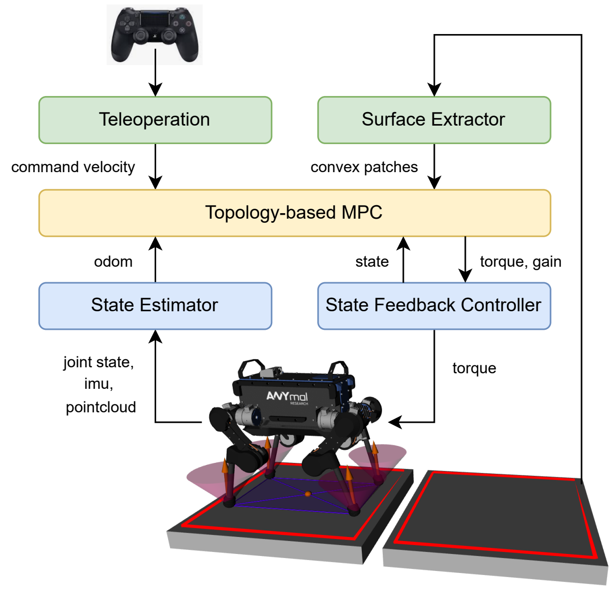

Fig. 3. Overview of our locomotion pipeline. Our topology-based MPC computes full-body motions, torque commands, and feedback policies given a reference velocity command and candidate contact surfaces.

respectively) is the position of the foot, $\mathbf{A}$ and $\mathbf{a}$ describes the contact surface, $\oplus$ and $\ominus$ denote *integration* and *difference operations* needed to optimize over manifolds [37]—notations introduced in Crocoddyl [5]. Furthermore, ℓ_k^{reg} regularizes the robot's configuration around a nominal posture $\mathbf{q}^{ref}$, the generalized velocity, joint torques, and contact forces as follows:

$$\ell_k^{reg} = \|\mathbf{q}_k \ominus \mathbf{q}^{ref}\|_{\mathbf{Q}}^2 + \|\mathbf{v}_k\|_{\mathbf{N}}^2 + \|\mathbf{u}_k\|_{\mathbf{R}}^2 + \|\boldsymbol{\lambda}_{\mathcal{C}_k}\|_{\mathbf{K}}^2,$$

where $\mathbf{Q}, \mathbf{N} \in \mathbb{R}^{n_v \times n_v}$, $\mathbf{R} \in \mathbb{R}^{n_j \times n_j}$ and $\mathbf{K} \in \mathbb{R}^{n_c \times n_c}$ are a set of (positive definite) diagonal weighting matrices.

To enable the automatic footstep placement, we first define a quadratic cost $w_r\|\dot{\mathbf{r}}_k - \dot{\mathbf{r}}^{ref}\|^2$ (with w_r as its weight) that tracks the reference base velocity $\dot{\mathbf{r}}^{ref}$ in the horizontal plane. Next, for all active contacts, we include our contact-surface penalty function $\sum_{\mathcal{C}_k} \ell_{\mathbf{p}_{\mathcal{C}_k}}$ (with w_f as its weight). We also add a constraint that specifies the contact surface height. Finally, we impose constraints that define the vertical motion of the swing foot. Note that we define the $\log(\cdot)$ operator in the vertical foot velocity constraint as the footstep placement may lie on a $\mathbb{SE}(3)$ manifold. ${}^{\mathcal{W}}\mathbf{p}_{\mathcal{G}}^{-1} \cdot {}^{\mathcal{W}}\mathbf{p}_{\mathcal{G}}^{ref}$ describes the inverse composition between the reference and current contact placements [38].

Again, we employ the robot's full-body dynamics (contact and impulse dynamics) in Eq. (8), where $\mathbf{M} \in \mathbb{R}^{n_v \times n_v}$ is the joint-space inertia matrix, $\mathbf{J}_{\mathcal{C}} \in \mathbb{R}^{nc \times nv}$ is the active contact Jacobian, $\boldsymbol{\tau}_b^{\mathcal{C},\mathcal{I}} \in \mathbb{R}^{n_v}$ is the force-bias vector, $\mathbf{a}_{\mathcal{C}} \in \mathbb{R}^{n_c}$ is the desired acceleration, and $\mathbf{C}$ and $\mathbf{c}$ describes the linearized friction cone (or wrench cone for humanoids). Note that the definition of the force-bias term ($\boldsymbol{\tau}_b^{\mathcal{C}}$ or $\boldsymbol{\tau}_b^{\mathcal{I}}$) changes between the contact and impulse dynamics. For more details on the regularization cost, dynamics, friction cone, and implementation aspects of our MPC, we encourage readers to refer to [11].

B. Locomotion Pipeline and Experimental Setup

Fig. 3 illustrates our locomotion pipeline. Reference velocities are sent using a joystick that runs at $30\,\mathrm{Hz}$. Candidate footstep surfaces in the predefined environment are extracted from mesh files at $10\,\mathrm{Hz}$. Our topology-based MPC operates at $50\,\mathrm{Hz}$ with an optimization horizon of $0.85\,\mathrm{s}$.

Our MPC, teleoperation, and surface extractor run on two external PCs. The joystick and surface extractor run on an Intel Core i9-9980 PC (8 core, 2.40 GHz) while the MPC runs on an Intel Core i9-9900 PC (8 core, 3.60 GHz). Lastly, the state feedback controller and state estimator operate at $400\,\mathrm{Hz}$ on two separate onboard PCs equipped with an Intel Core i7-7500 (2 core, 2.70 GHz).

IV. Results

In this section, we first show the impact of considering the robot's full-body dynamics when selecting footstep placements (Section IV-A). It justifies the benefits of our approach compared to other state-of-the-art methods that use simplified models. We then validate our approach in an MPC scheme on the ANYmal robot (Section IV-B). Finally, we showcase the potential highly-dynamic maneuvers that our approach can generate by exploiting limb dynamics in simulations (Section IV-C).

A. Footstep Planning with the Robot's Full-Body Dynamics

As discussed in Section I, neither CD nor SRBD can enforce torque limits. Furthermore, limb dynamics and kinematics are not included in the SRBD. Here, we present results that justify our full-body dynamics MPC that is able to consider the robot's torque limits and limb dynamics when planning footsteps placements and contact surfaces. To strictly focus on the independent variables (i.e., joint torque limits and friction coefficients), we set up the simulation as shown in Fig. 4.

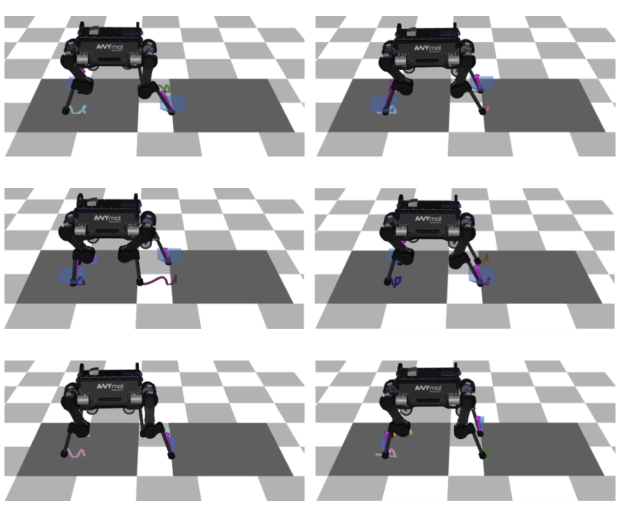

Fig. 4. Visualization of footstep placements and contact surfaces planned by our MPC when changing the joint torque limit and friction coefficient. The figures in the left and right columns of each row show the start and end of the crossing of the second foot, respectively. (a) ANYmal's default torque limits of $40\,\mathrm{N\,m}$ and high friction coefficient of 1.0. (b) Reduced torque limits of $20\,\mathrm{N\,m}$. (c) Low friction coefficient of 0.14. When the torque limits are reduced in (b), our approach moves the footstep closer to the robot's hip and selects contact surfaces that require lower torque commands. Instead, our approach stretches the robot's leg perpendicular to the ground when the friction coefficient is low. This is done to maintain the reaction force within the small friction cone.

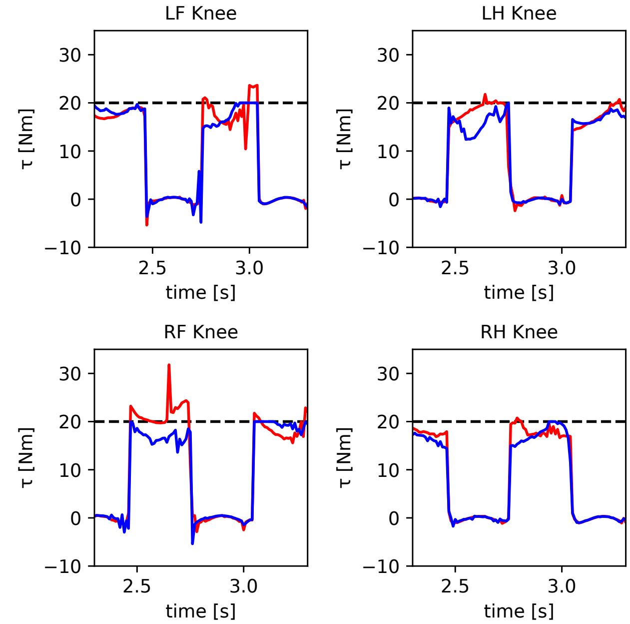

Fig. 5. Torque commands on the knee joints when the robot crosses the patches. The red line represents the torque commands computed with the ANYmal's default torque limit of $40\,\mathrm{N\,m}$, while the blue line represents the torque commands computed with the reduced torque limit of $20\,\mathrm{N\,m}$. The black dashed line indicates the reduced torque limit. With the default torque limit, the torque command for the right front knee computed by our MPC peaks at $2.7\,\mathrm{s}$. When the torque limit is lowered, our MPC changes the patch crossing moment as shown in Fig. 4b. The command torque for the left front knee reaches the limit at $2.9\,\mathrm{s}$, but it is maintained within it during that gait.

1) Joint Torque Limits: We investigated the ability of the robot to adjust to changing torque limits. The result will allow us to consider having the robot carry a payload (e.g., an arm), or more dynamic movements (e.g., crossing patches), without concern about exceeding the robot's torque limit. First, we used the ANYmal's default torque limits of $40\,\mathrm{N\,m}$ to have the robot walk with a trotting gait on the footstep surface of two pallets with a gap of $30\,\mathrm{cm}$. Next, we reduced the torque limits by half. Fig. 4a and 4b show the selection of contact surfaces and footstep placements for the default and reduced torque limits, respectively. These demonstrate that our MPC with the reduced torque limit makes the robot positions its legs closer together. As a result, the joint torque commands are reduced and kept within the torque limit, as demonstrated in Fig. 5.

2) Limb Dynamics: We varied the friction coefficient to investigate how the robot leverages limb dynamics when determining contact surfaces and footstep placements. We used the ANYmal's default torque limits and the same reference velocity used in Section IV-A1. As shown in Fig. 4c, our approach stretched the robot's leg perpendicular to the ground so it can maintain the contact forces within the friction cone (with a friction coefficient of 0.14).

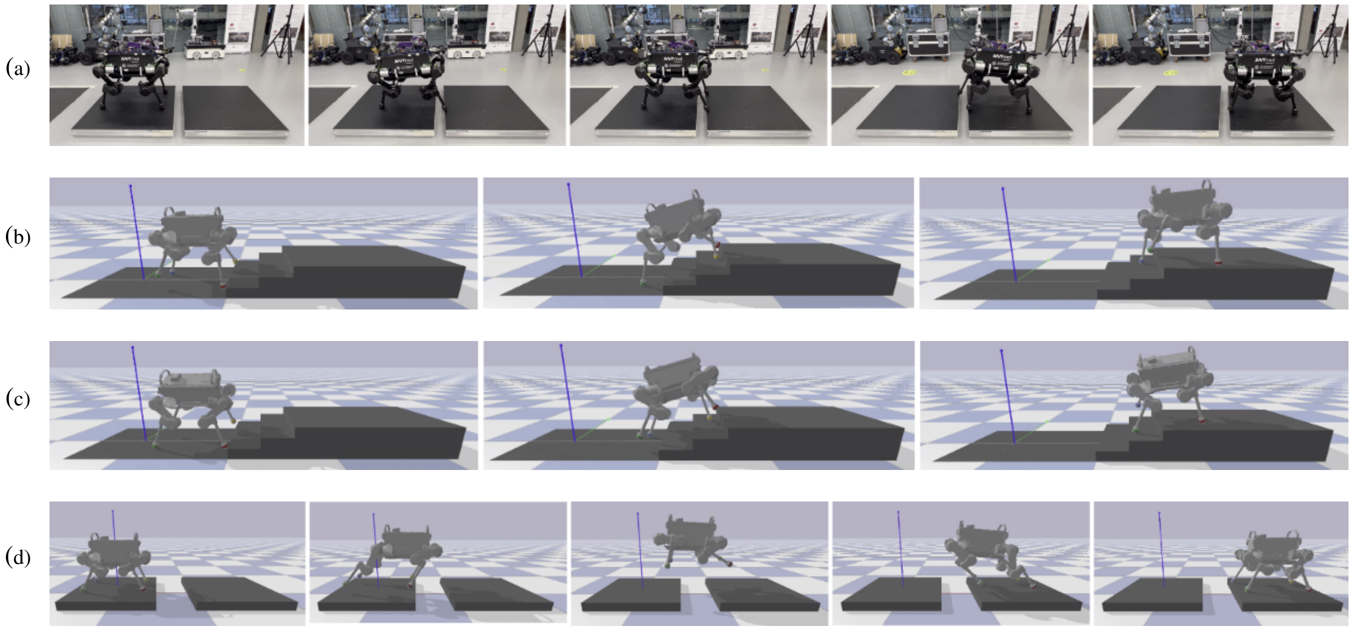

Fig. 6. Snapshots of various locomotion maneuvers computed by our topology-based MPC with automatic footstep placement and contact surface selection. We conducted all experimental and simulation trials using an MPC with an optimization horizon of $0.85\,\mathrm{s}$ and a control frequency of $50\,\mathrm{Hz}$. To seek to further explore the potential for more dynamic maneuvers that could be achieved with high-torque actuators, we increased the torque limit for simulations. (a) Experimental validation of gap-crossing maneuver with a gap of $10\,\mathrm{cm}$. (b) Climbing up a staircase of $30\,\mathrm{cm}$ depth and $10\,\mathrm{cm}$ height with a trotting gait. (c) Climbing up the same staircase with a pacing gait. (d) Jumping over a gap of $40\,\mathrm{cm}$.

B. MPC Validation on the ANYmal Robot

We validated our MPC on the ANYmal robot. The robot did not perceive its surroundings but received the candidate contact surface information extracted from the predefined environment as shown in Fig. 3. The robot moved with a walking gait. It selected the next contact surface and planned its footstep placement within it while considering the full-body dynamics and kinematics as shown in Fig. 6a. Our topology-based MPC enabled the ANYmal robot to successfully traverse a discrete terrain with a gap of $10\,\mathrm{cm}$. This demonstrates that our approach can run fast enough for MPC schemes on real robots, with an optimization horizon of $0.85\,\mathrm{s}$ and a control frequency of $50\,\mathrm{Hz}$.

C. Achieving Dynamic Locomotion

We further evaluated our MPC scheme in exploiting limb dynamics to achieve highly dynamic quadrupedal locomotion through simulations. Specifically, we demonstrated stair-climbing and gap-crossing scenarios. To showcase the potential capabilities of our approach in generating highly dynamic maneuvers when equipped with high-torque actuators in the future, we disabled the robot's torque limits.

1) Stair Climbing: We extracted the contact surfaces of a staircase with $30\,\mathrm{cm}$ depth and $10\,\mathrm{cm}$ height. Our MPC approach selected the contact surfaces and footstep placements to climb stairs in real time. The approach demonstrated the ability to handle both trotting and pacing dynamics, as shown in Fig. 6b and 6c.

2) Dynamic Jumping: We tested even more dynamic jumping gait on terrain with a $40\,\mathrm{cm}$ gap. Fig. 6d shows that our MPC can achieve highly dynamic motions and adjust the contact surfaces and footstep placements to land reliably on the next contact surface.

V. Conclusion

In this work, we introduced a novel topology-based approach that enables full-body dynamics MPC to automatically select footstep placements and contact surfaces for locomotion over discrete terrains. Specifically, we proposed a contact-surface penalty function using electric potential and winding number that optimizes both footstep placement and contact surface selection. With our method, we first justified the importance of considering full-body dynamics, which includes joint torque limits and limb dynamics, in footstep planning. We evaluated the planned footsteps and demonstrated that our full-body dynamics MPC can effectively adapt to variations in joint torque limits and friction coefficients. Second, to demonstrate its practical implementation of our method in an MPC scheme, we conducted hardware experiments on discrete terrain using the ANYmal quadruped robot. Finally, we showcased the potential capabilities of our approach in various dynamic locomotion maneuvers with robots equipped with high-torque actuators through stair-climbing and gap-crossing simulations.

For future work, we will implement perceptive locomotion with our MPC in changing environments to demonstrate the practical usefulness of our work in solving real-world problems. Furthermore, we will compare our full-body dynamics MPC approach with reduced-order model dynamics MPC applications through empirical comparisons to highlight the significance of our proposed approach.

REFERENCES

[1] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, P. Frank, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for Atlas," *Autonomous Robots*, vol. 40, 2016.

[2] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Motion Planning for Quadrupedal Locomotion: Coupled Planning, Terrain Mapping and Whole-Body Control," *IEEE Transactions on Robotics (T-RO)*, 2020.

[3] F. Jenelten, R. Grandia, F. Farshidian, and M. Hutter, "TAMOLS: Terrain-Aware Motion Optimization for Legged Systems," *IEEE Transactions on Robotics (T-RO)*, 2022.

[4] 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 International Conference on Robotics and Automation (ICRA)*, 2015.

[5] 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 International Conference on Robotics and Automation (ICRA)*, 2020.

[6] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, "An Efficient Acyclic Contact Planner for Multiped Robots," *IEEE Transactions on Robotics (T-RO)*, vol. 34, 2018.

[7] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research (IJRR)*, vol. 33, 2014.

[8] C. Mastalli, I. Havoutis, M. Focchi, D. G. Caldwell, and C. Semini, "Hierarchical planning of dynamic movements without scheduled contact sequences," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[9] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernandez-Lopez, and C. Semini, "Simultaneous Contact, Gait, and Motion Planning for Robust Multilegged Locomotion via Mixed-Integer Convex Optimization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, 2018.

[10] S. Tonneau, D. Song, P. Fernbach, N. Mansard, M. Taïx, and A. D. Prete, "SL1M: Sparse L1-norm Minimization for contact planning on uneven terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[11] C. Mastalli, W. Merkt, G. Xin, J. Shim, M. Mistry, I. Havoutis, and S. Vijayakumar, "Agile Maneuvers in Legged Robots: a Predictive Control Approach," 2022, arXiv:2203.07554.

[12] C. Mastalli, S. Chhatoi, T. Corbères, S. Tonneau, and S. Vijayakumar, "Inverse-Dynamics MPC via Nullspace Resolution," 2022, arXiv:2209.05375.

[13] C. Mastalli, J. Marti-Saumell, W. Merkt, J. Sola, N. Mansard, and S. Vijayakumar, "A Feasibility-Driven Approach to Control-Limited DDP," *Autonomous Robots*, 2022.

[14] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2014.

[15] A. W. Winkler, D. C. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems through Phase-based End-Effector Parameterization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, 2018.

[16] J. Wang, I. Chatzinikolaidis, C. Mastalli, W. Wolfslag, G. Xin, S. Tonneau, and S. Vijayakumar, "Automatic Gait Pattern Selection for Legged Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[17] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2014.

[18] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Footstep Placement," *Advanced Robotics*, vol. 24, 2010.

[19] 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 International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[20] G. Bledt and S. Kim, "Implementing Regularized Predictive Control for Simultaneous Real-Time Footstep and Ground Reaction Force Optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[21] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online Non-linear Centroidal MPC for Humanoid Robot Locomotion with Step Adjustment," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[22] T. Corberes, T. Flayols, P.-A. Leziart, R. Budhiraja, P. Soueres, G. Saurel, and N. Mansard, "Comparison of predictive controllers for locomotion and balance recovery of quadruped robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[23] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, 1966.

[24] "Harwell Subroutine Library, AEA Technology, Harwell, Oxfordshire, England. A catalogue of subroutines," http://www.hsl.rl.ac.uk/.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[26] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[27] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2017.

[28] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback MPC for Torque-Controlled Legged Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[29] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-Layered Safety for Legged Robots via Control Barrier Functions and Model Predictive Control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[30] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[31] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, 2018.

[32] S. Katayama and T. Ohtsuka, "Whole-body model predictive control with rigid contacts via online switching time optimization," 2022, arXiv:2203.00997.

[33] W. Li and E. Todorov, "Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2004.

[34] T. A. Howell, B. Jackson, and Z. Manchester, "ALTRO: A Fast Solver for Constrained Trajectory Optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[35] J. O'Rourke, *Computational Geometry in C*. Cambridge University Press, 1998.

[36] V. Ivan, D. Zarubin, M. Toussaint, T. Komura, and S. Vijayakumar, "Topology-based representations for motion planning and generalization in dynamic environments with interactions," *The International Journal of Robotics Research*, vol. 32, 2013.

[37] D. Gabay, "Minimizing a differentiable function over a differential manifold," *Journal of Optimization Theory and Applications*, vol. 37, 1982.

[38] J.-L. Blanco, *A tutorial on SE(3) transformation parameterizations and on-manifold optimization*. University of Malaga, Technical Report, 2010.