Animal Motions on Legged Robots Using Nonlinear Model Predictive Control

Dongho Kang*, Flavio De Vincenti*, Naomi C. Adam†, and Stelian Coros*

***Abstract*— This work presents a motion capture-driven locomotion controller for quadrupedal robots that replicates the non-periodic footsteps and subtle body movement of animal motions. We adopt a nonlinear model predictive control (NMPC) formulation that generates optimal base trajectories and stepping locations. By optimizing both footholds and base trajectories, our controller effectively tracks retargeted animal motions with natural body movements and highly irregular strides. We demonstrate our approach with prerecorded animal motion capture data. In simulation and hardware experiments, our motion controller enables quadrupedal robots to robustly reproduce fundamental characteristics of a target animal motion regardless of the significant morphological disparity.**

I. Introduction

Legged animals continue to inspire roboticists with their elegant motions. Not only do they possess impressive athletic motor skills, but even their slower-paced walking gaits are distinctively smooth and graceful. Reproducing such movements on legged robots comes with great challenges due to the limited understanding of the underlying patterns and principles.

A common trend in recent research addressing this challenge is to leverage animal motion capture data [1, 2, 3, 4]. This approach requires a control pipeline to bridge the gap in morphology between animals and robots and to robustly track target motions that inherit the agile and non-periodic nature of animal movements.

This paper presents a motion capture-driven locomotion control method capable of recreating animal motions on quadrupedal robots. To this end, we extend our previous work [4] by adopting a novel nonlinear model predictive control (NMPC) formulation [5] that jointly optimizes for dynamically feasible base trajectories and stepping locations using the *variable-height* inverted pendulum model (VHIPM) [6]. The foothold optimization significantly improves the robustness of the system to external disturbances and enables a robot to track retargeted animal motions with small variations in body movements and frequent flight phases. We compute the required derivatives using sensitivity analysis (SA) [7, 8, 9] and employ a sparse Gauss-Newton (SGN) method [10] to efficiently solve an unconstrained, optimal control problem.

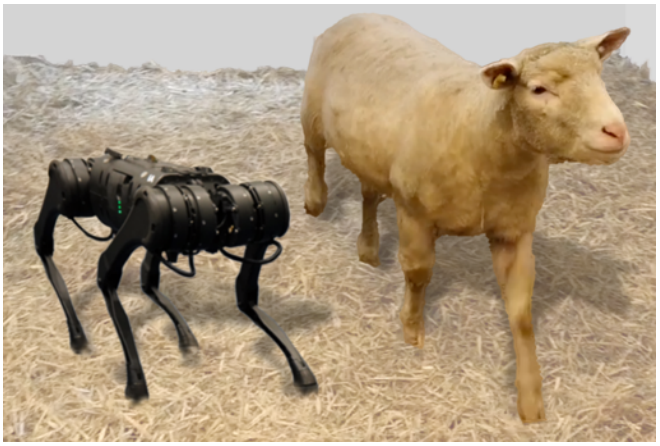

Fig. 1: The quadrupedal robot *Unitree A1* (**left**) and an alpine sheep (**right**). We recorded walking motions of the sheep using a marker-based motion capture system, and reproduced the retargeted motion profiles on the robot.

We validate our approach in simulation and through hardware tests with the two different-sized quadrupedal robots, *Unitree Aliengo* [11] and *Unitree A1* [12]. In our experiments, we reproduce a variety of motion clips recorded from a dog and sheep on the robots. As our video footage [1] shows, the resulting robot motions preserve irregular footstep timings and subtle body movements seen in animal motions. Since our controller jointly optimizes stepping locations and a base trajectory, it tracks retargeted animal motions more robustly than our previous method [4], which was based on a heuristics-based foothold planning strategy and a linear model predictive control (LMPC) formulation [13].

In summary, the main contribution of this paper is twofold. First, we present a novel NMPC formulation for quadrupedal locomotion capable of robustly handling arbitrary gait patterns. Second, we integrate the NMPC into our motion control pipeline that replicates the non-periodic footsteps and subtle body movements of animal motions.

II. Related Work

This work aims to reproduce animal motions on four-legged robots using a model predictive control (MPC) method. Thus, we present previous literature on animal-like behaviors on quadrupedal robots in Section II-A and MPC strategies for legged locomotion in Section II-B, respectively.

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 866480). The ethics have been approved by the veterinary authorities of the canton Zürich (application No. ZH095/2020)

*The authors are with the Computational Robotics Lab in the Institute for Intelligent Interactive Systems (IIIS), ETH Zurich, Switzerland. {kangd, dflavio, scoros}@ethz.ch

†The author is with the Laboratory for Movement Biomechanics in the Institute for Biomechanics (IfB), ETH Zurich, Switzerland. naomi.adam@hest.ethz.ch

The first two authors contributed equally to this work.

[1]The video is available in https://youtu.be/TVV_GcNZ0Ts

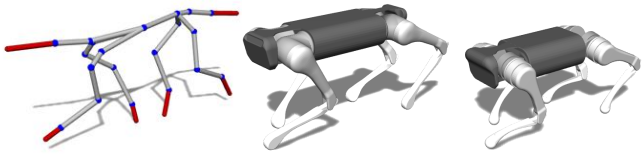

Fig. 2: The *trot* gait of a dog (**left**) reproduced on *Unitree Aliengo* (**middle**) and *Unitree A1* (**right**).

A. Animal Motion on Quadrupedal Robots

Many methods to recreate animal motions on four-legged machines rely on marker-based motion capture data recorded from animals [1, 4, 3, 2]. This approach preserves the patterns of animal movements. However, since the morphology of robotic systems is significantly different from their natural counterpart, it requires a remapping procedure to ensure that the resulting target motions are feasible on a robot.

Peng et al. [1] map a set of keypoints on an animal's body with corresponding target keypoints on a robot. They convert the position trajectories into joint trajectories using inverse kinematics. Eventually, a behavior-specific control policy is trained by a reinforcement learning algorithm in a simulated environment and afterward deployed to hardware. Li et al. [3] adopt a similar approach with an additional post-processing step that fits motion primitive parameters for a robot's base and foot trajectories in a physically simulated environment. By doing so, they transform an animal motion into a dynamically feasible target robot motion which a model-based controller can effectively track without behavior-specific tuning.

Yin et al. [2] use a learned gait planning policy to imitate the footfall patterns of an animal. The policy learns to adapt the footfall timings of an input animal footstep sequence so that a quadratic program-based whole-body controller tracks accordingly-generated reference trajectories for a robot.

In our previous work [4], we also transfer an animal footfall timings to robots. Additionally, we extract the body height and speed profiles of an animal motion and scale them according to the dimensions of a target robot platform. Then, we distill this information into reference base and foot trajectories; finally, we employ an LMPC controller based on the linearized single rigid body dynamics (SRBD) model [14, 13] to track them.

In this work, we employ the same motion transfer method from our previous work [4]. In contrast to keypoint-based motion retargeting methods [1, 3], we transfer only high-level information for a target animal motion. This approach is not constrained by a specific morphology; thus, it enables easy adaption to different robot platforms with mocap data sets collected from various legged animals – see Fig. 2. Meanwhile, we further improve the robustness of our model-based motion tracking controller by introducing a novel NMPC formulation for generating optimal, dynamically feasible base trajectories and footholds. The latter turns out to be crucial to prevent a robot from reaching unstable configurations when the target animal motions feature abrupt speed changes or fast-paced behaviors.

B. Model Predictive Control for Legged Locomotion

MPC for legged locomotion has been extensively studied by the robotics community. Most implementations solve a finite-horizon optimal control problem with simplified dynamics models. Carlo et al. [14] demonstrate an LMPC method on the four-legged robot *MIT Cheetah 3* [15] by planning ground reaction forces with a simplified SRBD model. Kim et al. [13] extend Carlo et al.'s framework by complementing it with a whole-body impulse control implementation. Ding et al. [16] propose an LMPC controller with a variation-based linearization scheme for the SRBD model that simplifies the rotational dynamics and frees them from gimbal lock and quaternion unwinding issues. The above LMPC approaches are shown to be effective for performing dynamic maneuvers. However, they rely on Raibert's foot placement rules [17] for planning footholds. In our experiments described in Section V, such heuristic strategies prove insufficient to reject external disturbances for aggressive target motions, stressing the necessity of a dynamics-aware foothold optimization method.

Optimizing stepping locations alongside the system trajectories within a single MPC framework introduces nonlinearities and complexity: consequently, it requires efficient solvers and special measures to prevent convergence to undesirable solutions. Bledt et al. [18] address these issues by adopting a foothold regularization procedure based on heuristic rules. The authors successively extend their approach by generating more polished, regularizing heuristics with a data-driven method [19, 20]. In contrast, our method does not require heuristics, and it yields motion controllers that can adapt stepping locations based on the morphological differences between a robot's kinematic design and the anatomy of the imitated animals.

In recent years, MPC methods based on differential dynamic programming (DDP) have seen numerous successful applications on quadrupedal robots. For instance, Neunert et al. [21] use a Gauss–Newton multiple shooting (GNMS) approach to optimize for both stepping locations and timings using the full system dynamics of *ANYmal* [22]. Grandia et al. [23] propose computing a feedback policy with a DDP-based algorithm and using it to bridge the gap between the low MPC update frequency and the fast rates of an inverse dynamics-based controller. While powerful, the above approaches solve high-dimensional systems, and thus they strictly rely on high-performance implementations. In contrast, we advocate adopting simpler models that enable the desired animal motions on a robot. More specifically, we achieve this by jointly optimizing stepping locations and a base trajectory subject to the inverted pendulum dynamics. Using a direct single shooting method [24], we efficiently solve an unconstrained, parameterized optimal control problem in a receding horizon fashion.

Finally, we highlight the MPC method by Xin et al. [25] that employs the linear inverted pendulum model (LIPM) [26, 27] to formulate a QP problem that optimizes relative stepping locations for gait patterns involving exclusively

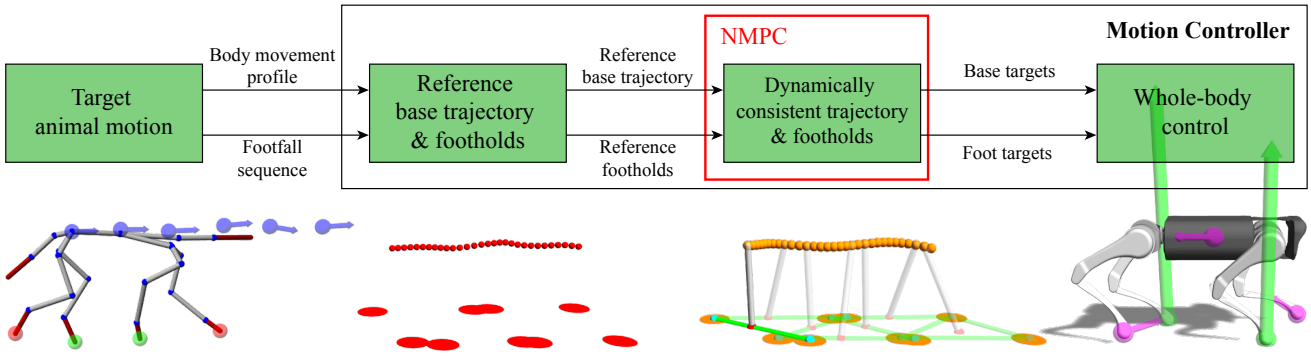

Fig. 3: Control pipeline overview. Each stage (**green blocks**) generates motion primitives or control signals using information (**black arrows**) from the previous stage. As a first step, we extract a body movement profile (**blue spheres/arrows**) and a sequence of contact (**green spheres**) and swing (**red spheres**) foot timings from a target animal motion. Next, we generate a reference base trajectory (**red dots**) and footholds (**red circles**). Then, an MPC planner optimizes the reference signals obtained based on a simplified dynamics model (**orange dots/circles**). Finally, a whole-body controller calculates joint torque and acceleration commands as well as ground reaction forces (**green arrows**) given the planned base and foot trajectories (**purple arrows**.)

single-support phases. We similarly elect to optimize relative footsteps, but we do so through a nonlinear programming (NLP) problem based on the more expressive VHIPM [6]. Moreover, by parameterizing the center of pressure positions, we are not restricted in the choice of gaits.

III. Pipeline Overview

Our control pipeline converts target animal mocap data into joint torque commands for a quadrupedal robot in four stages (see Fig. 3). First, we process the motion data to extract a body movement profile and footstep timings. Specifically, we collect trajectories of forward, sideways, and turning speed signals, as well as parts of the body pose we want to reproduce, such as body height [2]. We multiply each quantity by a hand-tuned scaling factor that addresses the dimension and actuation power discrepancy between the robot and the animal. Then, we extract a sequence of contact and swing phase timings for each animal limb. When these data are unavailable from a given motion clip, we use a simple threshold scheme that detects a swing phase if

$$\|\mathbf{v}_{\text{foot}}\|_2 > \theta_{\text{velocity}} \quad \text{or} \quad z_{\text{foot}} > \theta_{\text{height}}\,, \tag{1}$$

where θ_{height} and θ_{velocity} are the specified thresholds for the height z_{foot} and velocity $\mathbf{v}_{\text{foot}}$ of a foot, respectively. Additionally, we discard a phase transition if the subsequent phase lasts less than a time duration threshold θ_{phase}. We refer the reader to Table II for a complete list of the parameter values we used in our experiments.

In the second stage, we generate reference base trajectories and footholds for the desired robot motion. We adopt the scaled base heights from the target animal body trajectory as reference signals for our MPC planner. We further compute the reference horizontal base position coordinates $\mathbf{p}^{\text{ref}}_{\text{base},xy}$ and heading angles ψ^{ref} by numerically integrating the horizontal velocity components of the animal motion. Next, we determine the reference footholds $\mathbf{r}^{\text{ref}}_{\text{foot},i} \in \mathbb{R}^2$ for the foot i so that they lie below the corresponding hip at the middle of the stance phase. More formally, given the start time t_{stance} of a stance phase, we can write the corresponding reference stepping location as:

$$\mathbf{r}^{\text{ref}}_{\text{foot},i} = \mathbf{p}^{\text{ref}}_{\text{base}}(t_{\text{mid-stance}}) + \mathbf{R}_z\big(\psi(t_{\text{mid-stance}})\big)\,\mathbf{p}_{\text{hip},i}\,, \tag{2}$$

$$t_{\text{mid-stance}} = t_{\text{stance}} + 0.5\,\Delta t_{\text{stance}}\,, \tag{3}$$

where $\mathbf{R}_z$ is the rotation matrix around the z-axis, $\mathbf{p}_{\text{hip},i}$ is the position of the i^{th} hip with respect to the base local frame, and Δt_{stance} is the duration of the stance phase. We note that this approach helps to regularize the foothold optimization towards kinematically feasible solutions.

Third, we refine the base trajectories and stepping locations obtained from the previous stage while accounting for the dynamic effects of a robot's movement. For this purpose, we employ a novel MPC formulation with a simplified dynamics model – see Section IV. Given a pair of consecutive, optimized footholds, we generate target trajectories for the corresponding foot by interpolating cubic Bezier curves.

Finally, we track the optimized robot base and foot trajectories using a QP-based whole-body controller (WBC). We refer the reader to our previous papers [4, 7] for more details about our WBC implementation.

IV. Method

This section presents our nonlinear MPC method for parameterized, discrete-time dynamical systems. We introduce sensitivity analysis as a tool to efficiently compute the derivatives of a system dynamics required by our SGN solver. Our experiments adopt the VHIPM as a simplified template [28] for a legged system, and we augment it by including stepping locations as input parameters. However, our formulation can generalize to multiple dynamics models for quadrupedal robots [5].

A. Derivatives of General Dynamics

In a time-discretized setting, we let $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{u}_k \in \mathbb{R}^m$ be the state and control input vectors of a dynamical system at time step k. Also, let $\mathbf{p} \in \mathbb{R}^p$ be a time-invariant vector

[2]With a minor modification, we note that body pitch angle can be included in the movement profile for target animal motions where a variation of body pitch angle is salient.

of system parameters. Then, for $k \in \{0, 1, \ldots, N_T - 1\}$, we can express the dynamics in the following form:

$$\mathbf{x}_{k+1} = \mathbf{g}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}), \tag{4}$$

where $\mathbf{g}_k \colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$ is a differentiable function capturing the system evolution at time step k. In this formulation, we can use both $\mathbf{u}_k$ and $\mathbf{p}$ as control variables to drive the system towards desired states: the former only affects the dynamics at time step k, whereas the latter does so for multiple time steps. More specifically, in our NMPC formulation described in Section IV-C, $\mathbf{u}_k$ is the control input at time step k while $\mathbf{p}$ is a set of footholds over a time horizon.

We stack state and input vectors as

$$\mathbf{X} \coloneqq \begin{bmatrix} \mathbf{x}_1^\top & \mathbf{x}_2^\top & \ldots & \mathbf{x}_{N_T}^\top \end{bmatrix}^\top,$$
$$\mathbf{U} \coloneqq \begin{bmatrix} \mathbf{u}_0^\top & \mathbf{u}_1^\top & \ldots & \mathbf{u}_{N_T-1}^\top & \mathbf{p}^\top \end{bmatrix}^\top,$$

where N_T denotes the time horizon. Given the measured state $\mathbf{x}_0$, the vector $\mathbf{X}$ can be uniquely determined as a function of $\mathbf{U}$ through forward integration of (4). Thus, we can write $\mathbf{X} = \mathbf{X}(\mathbf{U})$.

Let $\mathcal{J}(\mathbf{X}, \mathbf{U})$ be a cost function that depends on both the stacked state and input vectors. We are concerned with finding the control inputs and parameters $\mathbf{U}^*$ minimizing the cost expressed as a function of $\mathbf{U}$, i.e., $\mathcal{J}(\mathbf{U}) = \mathcal{J}(\mathbf{X}(\mathbf{U}), \mathbf{U})$. We perform such optimization using a second order method, which affords a quadratic convergence rate for initial guesses sufficiently close to local minima. For this purpose, we employ *sensitivity analysis* [5, 7, 8, 9] to compute the first and second derivatives of the cost function:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\boldsymbol{U}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{X}}\mathbf{S} + \frac{\partial \mathcal{J}}{\partial \boldsymbol{U}}, \tag{5a}$$

$$\frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}\boldsymbol{U}^2} = \left(\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{U}}\frac{\partial \mathcal{J}}{\partial \boldsymbol{X}}\right)\mathbf{S} + \frac{\partial \mathcal{J}}{\partial \boldsymbol{X}}\frac{\mathrm{d}\mathbf{S}}{\mathrm{d}\boldsymbol{U}} + \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{U}}\frac{\partial \mathcal{J}}{\partial \boldsymbol{U}} \tag{5b}$$

$$\approx \mathbf{S}^\top \frac{\partial^2 \mathcal{J}}{\partial \boldsymbol{X}^2}\mathbf{S} + \mathbf{S}^\top \frac{\partial^2 \mathcal{J}}{\partial \boldsymbol{U}\partial \boldsymbol{X}} + \frac{\partial^2 \mathcal{J}}{\partial \boldsymbol{X}\partial \boldsymbol{U}}\mathbf{S} + \frac{\partial^2 \mathcal{J}}{\partial \boldsymbol{U}^2}, \tag{5c}$$

where $\mathbf{S} \coloneqq \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\boldsymbol{U}}$ is the *sensitivity matrix*. We employ the generalized Gauss–Newton approximation (5c) in place of the Hessian (5b) to reduce the computational burden and guarantee the semi-positive definiteness of the second derivative [10]. To calculate $\mathbf{S}$, we define the following vector of dynamics residuals:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0^\top & \mathbf{G}_1^\top & \ldots & \mathbf{G}_{N_T-1}^\top \end{bmatrix}^\top, \tag{6}$$

where

$$\mathbf{G}_k \coloneqq \mathbf{x}_{k+1} - \mathbf{g}_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}). \tag{7}$$

By construction, it holds $\mathbf{G}(\mathbf{X}, \mathbf{U}) = \mathbf{0}, \forall \mathbf{U}$. Then, under the assumptions of Dini's implicit function theorem, we can eventually write:

$$\mathbf{S} = -\left(\frac{\partial \mathbf{G}}{\partial \boldsymbol{X}}\right)^{-1}\frac{\partial \mathbf{G}}{\partial \boldsymbol{U}}. \tag{8}$$

We observe that (8) only requires partial derivatives, which are easy to compute analytically given that (4) has an analytic expression. Also, by making the vector $\mathbf{X}$, for which we can write an explicit expression through (4), artificially implicit, we can employ efficient batch approaches, such as sparse Gauss–Newton [10].

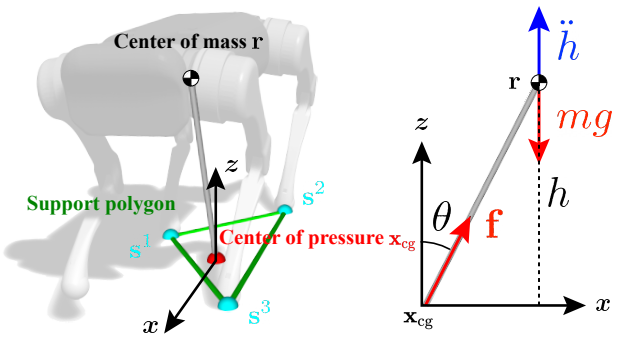

Fig. 4: Quadrupedal robot represented as a variable-height inverted pendulum. The translational dynamics of a legged system making contact with the ground can be approximated using the variable-height inverted pendulum model (**left**). A diagram of the forces acting on a two-dimensional projection of the system is on the **right**. For brevity, we depict a 2-dimensional case where the inverted pendulum lies on the xz-plane in this figure.

B. Simplified Robot Dynamics

In this section, we describe the VHIPM [6], which we use as a simplified, low-dimensional model for the NMPC formulation we adopt in this work. It represents a legged robot as a point mass m concentrated at the system's center of mass $\mathbf{r}$. The point mass is attached to a massless telescoping rod in contact with flat ground. The contact point of the rod is at the center of pressure (CoP) of the robot $\mathbf{x}_{cg}$, i.e., the location at which the resultant ground reaction force vector $\mathbf{f}$ would act if it were considered to have a single point of application [29]; a graphical description of the VHIPM is provided in Fig. 4.

The CoP always exists inside the support polygon of stance foot positions $\mathbf{s}^i \in \mathbb{R}^3$ [3]. Therefore, we express it as a convex combination

$$\mathbf{x}_{cg} = \sum_{\mathbf{s}^i \in \sigma} w^i \mathbf{s}^i, \tag{9}$$

where σ is the set of the stance foot positions, and $w^i \in \mathbb{R}_{\geq 0}$ is a non-negative scalar weight corresponding to $\mathbf{s}^i$ that satisfies

$$\sum_i w^i = 1. \tag{10}$$

Since the point mass has no rotational degrees of freedom, the moment of the force $\mathbf{f}$ about $\mathbf{r}$ is zero:

$$(\mathbf{r} - \mathbf{x}_{cg}) \times \mathbf{f} = \mathbf{0};$$

this implies that $(\mathbf{r} - \mathbf{x}_{cg})$ and $\mathbf{f}$ lie on parallel lines. Thus, it holds

$$\mathbf{f} = \frac{\mathbf{r} - \mathbf{x}_{cg}}{\|\mathbf{r} - \mathbf{x}_{cg}\|_2}\|\mathbf{f}\|_2. \tag{11}$$

[3]The CoP is always defined for legged machines in contact with the environment. In contrast, the zero moment point (ZMP) of a robot, where the horizontal components of the moments due to ground reaction forces applied to the system are zero, only exists for dynamically balanced gaits. However, when the ZMP is defined, it always coincides with the CoP [30].

From triangle similarities, it follows that:

$$\|\mathbf{f}\|_2 = \frac{\|\mathbf{r} - \mathbf{x}_{\text{cg}}\|_2}{r_z} f_z \,, \tag{12}$$

where r_z and f_z denote the components of $\mathbf{r}$ and $\mathbf{f}$ along the z-axis, respectively. Combining both (11) and (12), we can write

$$\mathbf{f} = (\mathbf{r} - \mathbf{x}_{\text{cg}}) \frac{f_z}{r_z} \,. \tag{13}$$

Finally, we can plug (13) into Newton's second law for the point mass, resulting in:

$$\begin{aligned} m\ddot{\mathbf{r}} &= \mathbf{f} + m\mathbf{g} \\ &= (\mathbf{r} - \mathbf{x}_{\text{cg}}) \frac{f_z}{r_z} + m\mathbf{g} \,, \end{aligned} \tag{14}$$

where $\mathbf{g} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^\top$ is the gravitational acceleration.

As shown in (14), we can control the point mass acceleration by modulating the CoP position $\mathbf{x}_{\text{cg}}$ and the component of $\mathbf{f}$ along the z-axis. We perform the following change of variables:

$$f_z = m(\ddot{h} + \|\mathbf{g}\|_2) \,, \tag{15}$$

that allows us to directly issue net vertical acceleration commands by means of the control input $\ddot{h}$. By inserting (9) and (15) into (14), we finally obtain the equations of motion of the system:

$$\boldsymbol{f}_{\text{IPM}}(\mathbf{r},\, \mathbf{u},\, \sigma) =: \ddot{\mathbf{r}} = (\mathbf{r} - \sum_{\mathbf{s}^i \in \sigma} w^i\, \mathbf{s}^i) \frac{\ddot{h} + \|\mathbf{g}\|_2}{r_z} + \mathbf{g} \tag{16}$$

with control input vector $\mathbf{u} := \begin{bmatrix} \ddot{h} & w^1 & w^2 & \dots & w^{\|\sigma\|} \end{bmatrix}^\top$, and system parameters $\mathbf{s}^i \in \sigma$.

We note that when no feet are in contact with the ground, (14) reduces to the equation of a projectile motion, i.e., $\ddot{\mathbf{r}} = \mathbf{g}$, and the system cannot be controlled. However, the capability of MPC to reason about time horizons rather than time instants enables our controller to gracefully overcome such uncontrolled states and, thus, handle dynamic gaits with extended flight phases.

C. Nonlinear Model Predictive Control

This section describes a single shooting MPC strategy [24] using the inverted pendulum model presented in Section IV-B. The optimization variables comprise vertical accelerations of a legged robot center of mass, stepping locations, and through corresponding weights, CoP positions. Instead of representing footholds as time-varying control inputs and enforcing transition constraints to ensure that stance feet do not move [19], we regard each stepping location as a single system parameter shared between multiple time steps. This approach significantly reduces the total number of optimization variables and allows for faster solution times.

Henceforth we denote the physical quantities sampled at time $t + k\,\Delta t$, $k \in \{0,\, 1,\, \dots,\, N_{\text{T}}\}$ using the subscript k, where $\Delta t \in \mathbb{R}_{>0}$ is the time step duration.

Based on the desired gait pattern, we fix the timings at which touchdown events occur over the optimization time horizon, and we associate each with a corresponding stepping location $\mathbf{s}^i$. Finally, we stack all the footstep positions in a vector $\mathbf{s} := \begin{bmatrix} \mathbf{s}^1 & \mathbf{s}^2 & \dots & \mathbf{s}^{N_{\text{f}}} \end{bmatrix}^\top$ sorted into footfall order, where N_{f} is the total number of footsteps.

To discretize the continuous VHIPM dynamics (16), we employ a *semi-implicit Euler method*; given $\mathbf{r}_k$ and $\mathbf{r}_{k-1}$ we approximate the point mass velocity at time steps k and $k+1$, respectively, as $\tilde{\mathbf{v}}_k \approx (\mathbf{r}_k - \mathbf{r}_{k-1})/\Delta t$ and $\tilde{\mathbf{v}}_{k+1} \approx \tilde{\mathbf{v}}_k + \ddot{\mathbf{r}}_k \Delta t$, where $\ddot{\mathbf{r}}_k$ can be computed using (16). Finally, we can write:

$$\begin{aligned} \mathbf{r}_{k+1} &\approx \mathbf{r}_k + \tilde{\mathbf{v}}_{k+1} \Delta t \\ &= \mathbf{r}_k + \tilde{\mathbf{v}}_k \Delta t + \ddot{\mathbf{r}}_k \Delta t^2 \\ &= 2\mathbf{r}_k - \mathbf{r}_{k-1} + \ddot{\mathbf{r}}_k \Delta t^2 \\ &= 2\mathbf{r}_k - \mathbf{r}_{k-1} + \boldsymbol{f}_{\text{IPM}}(\mathbf{r}_k,\, \mathbf{u}_k,\, \sigma_k) \Delta t^2 \\ &=: \mathbf{g}_{\text{IPM},k}\left(\mathbf{r}_{k-1},\, \mathbf{r}_k,\, \mathbf{u}_k,\, \mathbf{s}\right) \,, \end{aligned} \tag{17}$$

where $\sigma_k \subseteq \{\mathbf{s}^1,\, \mathbf{s}^2,\, \dots,\, \mathbf{s}^{N_{\text{f}}}\}$ denotes the set of the stance foot positions at time step k.

Let us redefine the stacked state and input vectors for the discretized IPM as $\mathbf{X} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \dots & \mathbf{r}_{N_{\text{T}}} \end{bmatrix}^\top$ and $\mathbf{U} = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_2 & \dots & \mathbf{u}_{N_T - 1} & \mathbf{s} \end{bmatrix}^\top$. Given a reference trajectory $\mathbf{r}_k^*$ for $k \in \{0,\, 1,\, \dots,\, N_{\text{T}}\}$, and a set of reference stepping locations $\mathbf{s}^{i*}$, $i \in \{1,\, 2,\, \dots,\, N_{\text{f}}\}$, we define the following cost function:

$$\mathcal{J}(\mathbf{X},\, \mathbf{U}) := \sum_{k=0}^{N_{\text{T}}-1} \|(\mathbf{r}_{k+1} - \mathbf{r}_k) - (\mathbf{r}_{k+1}^* - \mathbf{r}_k^*)\|_2^2 \tag{18a}$$

$$+ \sum_{k=0}^{N_{\text{T}}-1} \|h_{k+1} - h_{k+1}^*\|_2^2 \tag{18b}$$

$$+ \sum_{i=1}^{N_{\text{f}}} \sum_{j=i+1}^{\min(N_{\text{f}}, i+3)} K_1 \|(\mathbf{s}^i - \mathbf{s}^j) - (\mathbf{s}^{i*} - \mathbf{s}^{j*})\|_2^2 \tag{18c}$$

$$+ \sum_{k=0}^{N_{\text{T}}-1} \left(\frac{K_2}{2} \|1 - \sum_i w_k^i\|_2^2 + \sum_i \mathcal{S}_{\geq 0}(w_k^i) \right), \tag{18d}$$

where (18a) penalizes base velocity tracking errors, (18b) penalizes base height tracking errors, (18c) regularizes the displacements between adjacent stepping locations, and (18d) is a soft constraint on the stance foot weights w_k^i enforcing both (10) and nonnegativity. We define $\mathcal{S}_{\geq r} \colon \mathbb{R} \to \mathbb{R}_{\geq 0}$, $\forall r \in \mathbb{R}$ as a $\mathcal{C}^2$-continuous function following [31, eq. (8)] with unitary stiffness and $\epsilon = 0.1$.

We note that the term (18c) penalizes relative positions between stepping locations to make the corresponding support polygons loosely resemble the reference support polygons [25]. Also, we do not add cost terms for pairs of footstep positions separated by more than three touchdown events since we assume they are not significantly correlated.

Given the objective function (18), we formulate our MPC as the following unconstrained optimization problem:

$$\mathbf{U}^\star = \arg\min_{\mathbf{U}} \mathcal{J}(\mathbf{X}(\mathbf{U}),\, \mathbf{U}) \,, \tag{19}$$

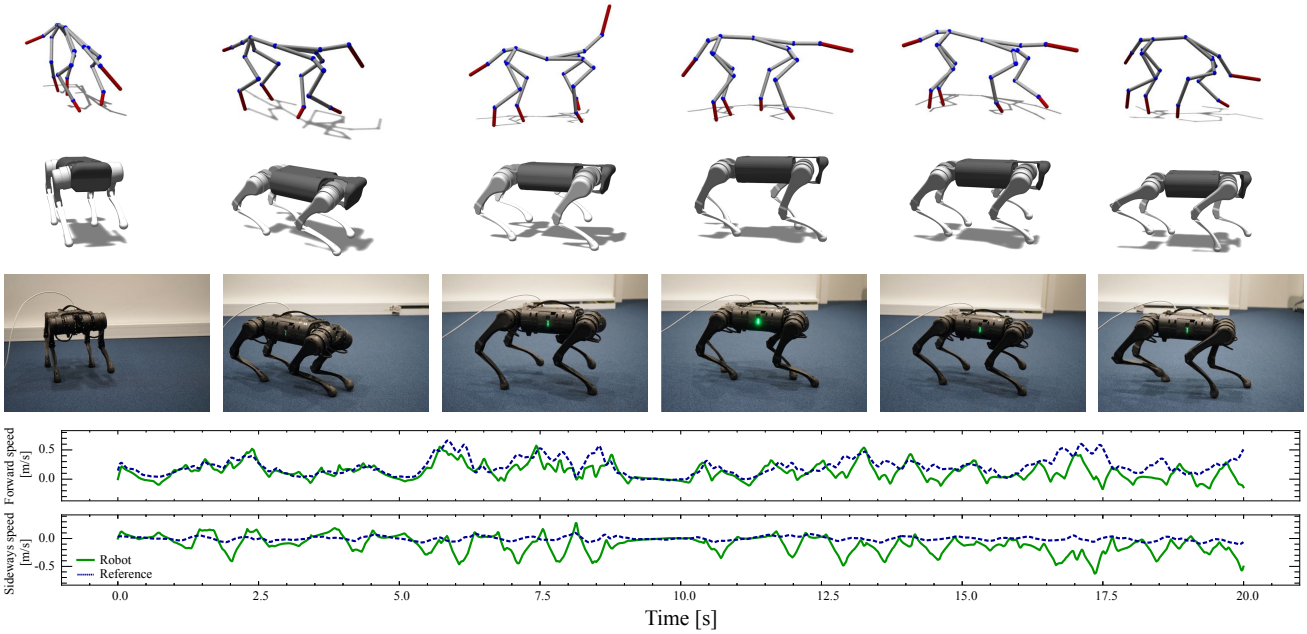

Fig. 5: A dog's motion capture data sequence (**first row**) reproduced on a physically simulated (**second row**) and a real (**third row**) *Unitree A1* robot. The plots show the simulated robot base's forward and sideways speed trajectories along with references extracted from the corresponding animal motion (**last row**). The root-mean-square errors for the two plots are $\sim 0.16\,\mathrm{m\,s^{-1}}$ and $\sim 0.17\,\mathrm{m\,s^{-1}}$, respectively.

where the stacked state vector $\mathbf{X}(\mathbf{U})$ is obtained through forward integration of (17) given the initial conditions $\mathbf{r}_0$ and $\mathbf{r}_{-1} := \mathbf{r}_0 - \mathbf{v}_0\,\Delta t$.

To solve the optimal control problem (19), we employ an SGN solver [10], and we compute the required derivatives using SA[4] – see Section IV-A. Once we compute a descent direction, we perform a backtracking line search, and we interrupt the optimization if the cost function (18) does not improve more than a predefined relative tolerance.

V. Results

We verify the effectiveness of our control pipeline in a series of simulation and hardware experiments with the quadrupedal robots *Unitree Aliengo* [11] and *Unitree A1* [12]. We use publicly available motion data recorded from a dog [32], which includes unconstrained motions with varying walking speeds. Additionally, we use data collected in-house from a sheep trained to walk on a treadmill. The body dimensions of the animals and the robots are provided in Table I. For the simulation experiments, we adopt a physically simulated environment based on the Open Dynamics Engine (ODE) [33] with a step size of $1/480\,\mathrm{s}$. In all our tests, we employ the same set of parameter values listed in Table II. With these settings, we manage to run the MPC at a rate of $\sim 250\,\mathrm{Hz}$ on an i7-9700K CPU. Our two-step integration scheme makes it possible to not include velocities in the state vectors but only positions, thus enabling three-dimensional states. This dimensionality reduction comes at the cost of slightly less sparse dynamics Jacobians that feature a non-zero block diagonal band below the main diagonal.

[4]We notice that the discretized dynamics (17) at time k are a function of both $\mathbf{r}_k$ and $\mathbf{r}_{k-1}$ due to our choice of time-stepping scheme, in contrast to the formulation (4) presented in Section IV-A. However, it is easy to verify that (8) is valid also for dynamics equations spanning multiple time steps.

TABLE I: Body dimensions of animals and robots. **Body height**: height of the sacrum for animals and height of the center of mass of the base for robots. Measured in the nominal standing pose. **Body length**: measurement from the sacrum to the cervical vertebra for animals and length of the base for robots. **Limb length**: measured in the fully-stretched configurations.

	Animals		Robots	
	Dog	Sheep	Aliengo	A1
Body height	0.46 m	0.8 m	0.4 m	0.3 m
Body length	0.42 m	0.7 m	0.345 m	0.25 m
Forward limb length	0.4 m	0.51 m	0.5 m	0.39 m
Hind limb length	0.45 m	0.62 m	0.5 m	0.39 m

Fig. 5 shows snapshots of simulation and real-world experiments conducted on a *Unitree A1* robot that reproduced a slow-paced *walk* gait of a dog. We include more footage of our experiments in the supplementary video. They show that our method can recreate unstructured stepping patterns and natural body movements of animals with different morphologies and dimensions. As detailed in Section II-A, we can easily adapt the control pipeline to various robot models thanks to our retargeting scheme.

Fig. 6 highlights the effect of the foothold optimization on the leg configurations. In the simulation experiment, we measured the step width of *Unitree Aliengo* while it replicates a dog's gait and compared the one from our pipeline to the one from our previous motion control pipeline [4] that relies on Raibert's heuristics: from here on out, we refer to it using the authors' names: *Kang et al., 2021*. We observe that Raibert's heuristics tend to yield too aggressive displacements for the feet when the target motions are highly dynamic: this often leads to undesirable crossing or excessive

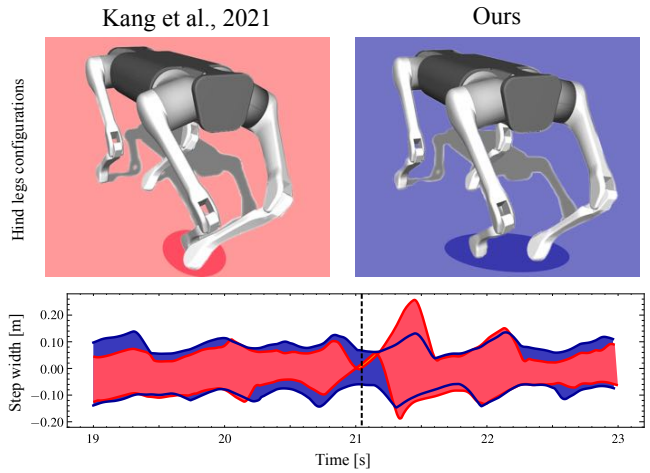

Fig. 6: Hind leg configurations (**top**, at time 21.05 s) and a step width plot (**bottom**) of *Unitree Aliengo* while it replicates a dog's *walk* gait in simulation using the previous control pipeline (**red**) and our new pipeline (**blue**). The lines in the plot are the trajectories of the hind feet in the robot base local frame along the sideways direction; the colored area between them corresponds to their distance along the same axis.

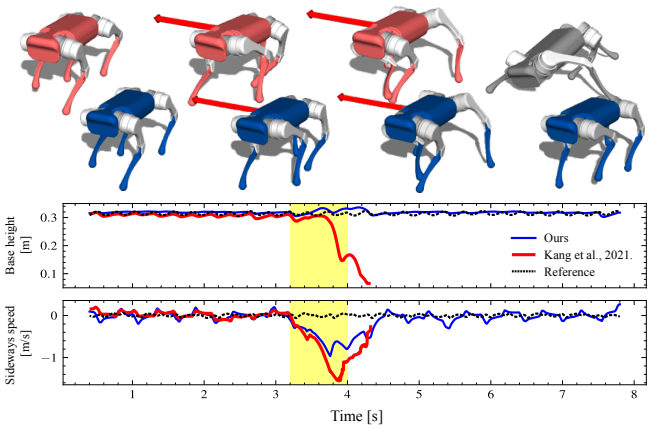

Fig. 7: Snapshots of a simulated *Unitree A1* robot during a perturbation test (**top**), a plot of its base height trajectories (**middle**) and a plot of its sideways speed (**bottom**). We apply a constant force (**red arrow**) to the robot from 3.2 s to 4 s while it imitates the *trot* gait of a sheep (shaded area in **yellow**). While the previous control pipeline (**red**) fails to withstand the disturbance, our new control pipeline (**blue**) succeeds by electing an appropriate sidestep size.

stretching of the limbs. In contrast, by optimizing stepping locations, our method can sustain appropriate step widths while ensuring dynamic stability. This detail is crucial for robustly reproducing animal movements, which are generally lively and agile.

Finally, we conducted a disturbance rejection test in simulation on the *Unitree A1* robot. Specifically, we applied a constant lateral force of 27 N to the robot for 0.8 s while it reproduced a sheep's *trot* gait. Once again, we compare our control pipeline to the prior one. As shown in Fig. 7, while the latter overreacts to the external disturbance and outputs unfeasible target accelerations, ours manages to adjust the step width adequately and reject the unexpected push.

VI. Conclusion and Future Work

In this paper, we present a motion capture-driven locomotion controller that recreates animal motions on quadrupedal robots. Our motion controller reproduces the unstructured footstep timings and natural body movements of given animal motions through a simple motion retargeting strategy. It is not constrained by a specific morphology; therefore, we can apply it to different robot platforms with mocap data sets collected from a variety of legged animals. At the core of our pipeline, our new NMPC method outputs optimal stepping locations and base trajectories, thus enabling a robot to track target animal motions more robustly. We demonstrate that the simultaneous optimization of footholds and body trajectories effectively prevents undesirable leg-crossing or -stretching configurations, which often arise when tracking abrupt speed changes or faster-paced behaviors using previous state-of-the-art control methods [4]. Most importantly, it significantly improves the ability of the motion controller to reject unexpected disturbances.

In our experiments, we use the VHIPM as a simplified template for a robot. Such a solution significantly reduces the size of the optimal control problem and allows for very efficient computation times, but it ignores the orientation dynamics of the system. Consequently, we solely address the orientation dynamics through a whole-body controller on a time instant-level rather than time horizon-level. Nevertheless, our MPC formulation is applicable to different dynamics models for quadrupedal locomotion as we described in the related paper [5]. As a future investigation, we are interested in combining our method with more complex dynamics models such as the SRBD and augmenting them with the same foothold optimization described in this work. We plan to explore the performance achievable in capturing small variations and nonperiodic patterns of animal motions by accounting for the orientation dynamics and physical constraints such as friction cone constraints and joint torque limits.

Furthermore, we intend to make our motion controller responsive to a user's high-level commands by integrating a gait planning strategy that imitates animal gait patterns. We will complement our framework with a data-driven gait planning method that can generate animal-like gaits on quadrupedal robots given a user's inputs [4].

TABLE II: Parameter values for the animal motion data post-processing and MPC planner.

	Dog	Sheep
Foot height threshold θ_{height}	0.05 m	0.15 m
Foot velocity threshold θ_{velocity}	0.8 m/s	1.5 m/s
Phase duration threshold θ_{phase}	0.1 s	0.1 s
Body forward speed scaling factor	0.5	0.25
Body sideways speed scaling factor	0.2 (*A1*) / 0.25 (*Aliengo*)	
Body turning speed scaling factor	0.2 (*A1*) / 0.25 (*Aliengo*)	

MPC Time step size	1/30 s
Planning time horizon N_{T}	30 (1 s)
Foothold regularizer weight K_1	0.1
Soft constraint weight K_2	100

Finally, we aim to investigate more fundamental research questions in future work. Researchers hypothesize that animal motion patterns emerge to maximize robustness, agility and energy efficiency [34]. We are curious whether we can achieve more benefits than just improved aesthetics by transferring these motion patterns to legged robots. Thus, we are interested in analyzing efficiency- and robustness-related aspects despite the different morphologies, dimensions, and actuation principles of animals and robots.

REFERENCES

[1] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems*, 07 2020.

[2] F. Yin, A. Tang, L. Xu, Y. Cao, Y. Zheng, Z. Zhang, and X. Chen, "Run like a dog: Learning based whole-body control framework for quadruped gait style transfer," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 8508–8514.

[3] T. Li, J. Won, S. Ha, and A. Rai, "Fastmimic: Model-based motion imitation for agile, diverse and generalizable quadrupedal locomotion," 2021. [Online]. Available: https://arxiv.org/abs/2109.13362

[4] D. Kang, S. Zimmermann, and S. Coros, "Animal gaits on quadrupedal robots using motion matching and model-based control," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8500–8507, 2021.

[5] D. Kang, F. D. Vincenti, and S. Coros, "Nonlinear model predictive control for quadrupedal locomotion using second-order sensitivity analysis," 2022. [Online]. Available: https://arxiv.org/abs/2207.10465

[6] J. Liu, H. Chen, P. M. Wensing, and W. Zhang, "Instantaneous capture input for balancing the variable height inverted pendulum," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7421–7428, 2021.

[7] F. De Vincenti, D. Kang, and S. Coros, "Control-aware design optimization for bio-inspired quadruped robots," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021)*. IEEE, 2021.

[8] S. Zimmermann, R. Poranne, and S. Coros, "Optimal control via second order sensitivity analysis," 2019. [Online]. Available: https://arxiv.org/abs/1905.08534

[9] J. M. Bern, P. Banzet, R. Poranne, and S. Coros, "Trajectory optimization for cable-driven soft robot locomotion." in *Robotics: Science and Systems*, vol. 1, no. 3, 2019.

[10] J. Zehnder, S. Coros, and B. Thomaszewski, "Sgn: Sparse gauss-newton for accelerated sensitivity analysis," *ACM Trans. Graph.*, vol. 41, no. 1, sep 2021.

[11] Unitree Robotics, "Aliengo," https://www.unitree.com/products/aliengo.

[12] ——, https://www.unitree.com/products/a1.

[13] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *ArXiv*, vol. abs/1909.06586, 2019.

[14] J. D. Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018.

[15] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.

[16] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, pp. 1154–1171, 2021.

[17] M. Raibert, *Legged Robots That Balance*. Cambridge, MA: MIT Press, 1986.

[18] G. Bledt, P. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4102–4109, 2017.

[19] G. Bledt and S. Kim, "Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6316–6323, 2019.

[20] G. Bledt, "Regularized predictive control framework for robust dynamic legged locomotion," Ph.D. dissertation, Massachusetts Institute of Technology, 2020.

[21] 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*, vol. 3, pp. 1458–1465, 2018.

[22] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[23] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4730–4737, 2019.

[24] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*. Springer, 2006, pp. 65–93.

[25] S. Xin, R. Orsolino, and N. Tsagarakis, "Online relative footstep optimization for legged robots dynamic walking using discrete-time model predictive control," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 513–520, 2019.

[26] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 1405–1411 vol.2.

[27] T. Koolen, T. D. Boer, J. R. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The International Journal of Robotics Research*, vol. 31, pp. 1094 – 1113, 2012.

[28] R. Full and D. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land." *The Journal of experimental biology*, vol. 202 Pt 23, pp. 3325–32, 1999.

[29] P. Cavanagh, "A technique for averaging center of pressure paths from a force platform." *Journal of biomechanics*, vol. 11 10-12, pp. 487–91, 1978.

[30] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life," *Int. J. Humanoid Robotics*, vol. 1, pp. 157–173, 2004.

[31] J. M. Bern, K.-H. Chang, and S. Coros, "Interactive design of animated plushies," *ACM Transactions on Graphics (TOG)*, vol. 36, pp. 1 – 11, 2017.

[32] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–11, Aug. 2018.

[33] R. Smith, "Open dynamics engine," https://www.ode.org/, 2001-2004.

[34] D. F. Hoyt and C. R. Taylor, "Gait and the energetics of locomotion in horses," *Nature*, vol. 292, no. 5820, pp. 239–240, 1981.